

UNIVERSITÉ DE LIÈGE
FACULTÉ DES SCIENCES APPLIQUÉES

MACHINE LEARNING APPROACHES
TO
POWER SYSTEM SECURITY ASSESSMENT

par

Louis WEHENKEL

Ingénieur civil électricien (électronique)
Docteur en Sciences Appliquées
Agrégé de l'enseignement supérieur en Sciences Appliquées
Chercheur qualifié du F.N.R.S.

1995

Thèse défendue, avec succès, le 24 mai 1994, pour l'obtention du grade d'Agrégé de l'enseignement supérieur en Sciences Appliquées de l'Université de Liège.

La commission de lecture était composée de :

D. RIBBENS, Président, Université de Liège
M. PAVELLA, Promoteur, Université de Liège
P. BORNARD, Electricité de France, Clamart, France
G. CANTRAINÉ, Université de Liège
B. J. CORY, Imperial College of Science and Technology, Londres, U.K.
T. DY LIACCO, Cleveland, U.S.A.
W. LEGROS, Université de Liège
R. PONCELET, Université Libre de Bruxelles
M. ROUBENS, Université de Liège
J. WILLEMS, Université de Gand

Le jury était composé des professeurs de la Faculté des Sciences Appliquées de l'Université de Liège, ainsi que des membres de la commission de lecture; il était présidé par le Doyen, G. L'HOMME.

Acknowledgements

I am pleased to express my deep gratitude to Professor Pavella who initiated me to research. Her permanent advices and discerning suggestions allowed me to progress in my task.

I am very much indebted to Dr. Cory who gracefully accepted to read a first draft of this thesis and gave many insightful indications.

I am also particularly thankful to all the other members of the Reviewing Committee who kindly accepted to assess my thesis.

I would like to acknowledge the contributions of colleagues of CEGELEC, Electricité de France and Hydro-Québec, who allowed progress by asking judicious questions and providing valuable information about practice.

I thank my colleagues, here at the University of Liège, who motivated my work through their rich discussions and fruitful collaborations.

Summary

Machine learning approaches to power system security assessment

In power system planning and operation, security assessment is one of the major, multifaceted problems. Increasing economic and environmental pressure as well as higher speeds and stronger action of modern control algorithms and devices make the conflicting aspects of reliability and economy even more challenging.

Until recently, the security studies carried out in a given context were essentially limited by the available simulation hardware and software. As computing powers grow, however, the bottleneck becomes more and more due to the ability of engineers to extract relevant information from bulky simulations. In particular, present day computer networks and their foreseeable growth in the near future, together with forthcoming fast and reliable simulation tools will allow the generation of huge amounts of detailed studies, by exploiting inherent parallelisms. In order to take due advantage of these possibilities, it is necessary to develop tools able to assist engineers to appraise and interpret the obtained results.

The present work describes research moving along the line of developing such information synthesis tools, adapted to the specific needs of power system security assessment. In the proposed approach, random sampling techniques are considered to screen all relevant situations in a given context, while existing numerical simulation tools are exploited to derive detailed security information. The heart of the framework is provided by statistical techniques able to extract and synthesize relevant information and to reformulate it in a suitable way for decision making.

Our work on this subject matter started about 8 years ago. The primary objective was

to explore whether and to what extent machine learning techniques were able to tackle power system transient stability.

The scope of our research has gradually broadened. We thus have been developing and improving the inductive inference method to make it meet specialities of the physical problem; at the same time we have been diversifying the practical application to other types of security assessment, in particular voltage security in both preventive and emergency fashions.

The work presented in this thesis is the culmination of the above research. Three themes are scrutinized : machine learning methods, power system security problems and the application of the former to the latter.

Machine learning methods that we have been developing, improving and adjusting throughout our research belong to the more general category of computer based learning methods. Our purpose in this work is twofold. On the one hand, to critically compare the various families of methods, in order to justify a posteriori our initial choice of the inductive inference method. On the other hand, to identify interesting specific strengths of various other techniques and select “good” candidates, i.e. likely to advantageously complement and enhance our method.

The second theme is power system security. The purpose is to present a comprehensive account of the phenomena and to point out general as well as specific characteristics from both the physical side and the practical contexts within which security can be assessed.

The success of the method resulting from the application of computer based learning techniques to power system security assessment heavily relies on the in-depth understanding of these two matters. It is the aim of the third theme to show that this original method has by now matured enough and that it is indeed able to bridge the gap between practical needs not met as yet, despite being urgent, and tools which are beginning to be available.

The thesis is structured as follows.

The introduction discusses different aspects of security assessment, introduces present day simulation tools, and outlines the information synthesis paradigm and the available statistical techniques.

Part 1 provides a unified description of information synthesis techniques, from three different perspectives. Firstly, a detailed account of machine learning is given; the emphasis is put on decision tree induction methods, the cornerstone of the proposed tools. This is followed by a synthetic overview of complementary methods of classical statistical pattern recognition as well as artificial neural networks. Finally, various types of machine learning problems are considered, and suitable techniques for solving

them are identified.

Part 2 focuses on security problems, both from the physical and the operational points of view. Transient stability, voltage security and to a lesser extent steady state security problems are compared and feasibility of preventive and emergency control modes in the context of on-line operation are discussed. The two last chapters of this part describe an in depth investigation of the data base generation techniques appropriate for different types of physical problems.

Part 3 provides a synthetic account of the practical experience we gained from several application studies, carried out at the University of Liège. A rather diverse range of tests are considered, combining different physical problems and power systems, in particular three real-life problems, investigated in the context of collaborations with Electricité de France and Hydro-Québec.

If a man will begin with certainties, he will end with doubts; but if he will be content with doubts, he shall end in certainties.

Francis Bacon (1561-1626)

Notre esprit a une irrésistible tendance à considérer comme plus claire l'idée qui lui sert le plus souvent.

Henri Louis Bergson (1859-1941)

A book should have either intelligibility or correctness; to combine the two is impossible.

Bertrand Russel (1872-1970)

Contents

Summary	iii
Notation	xix
1 Introduction	1
1.1 MACHINE LEARNING FOR SECURITY ASSESSMENT	1
1.2 AN OVERVIEW OF SECURITY PROBLEMS	2
1.2.1 Classification of operating states	3
1.2.2 Physical classification of security problems	5
1.2.3 Practical application domains	6
1.3 ANALYTICAL TOOLS	9
1.3.1 Transient stability	9
1.3.2 Voltage stability and security	11
1.3.3 Static security	12
1.4 AN OVERVIEW OF LEARNING METHODS	13
1.4.1 Generic problem of supervised learning	13
1.4.2 Classes of supervised learning methods	15

1.4.3	Clustering and unsupervised learning	20
1.5	A FLAVOR OF THE PROPOSED FRAMEWORK	21
1.5.1	Which methods should we combine in a tool-box ?	21
1.5.2	A hypothetical illustration of the framework	22
1.6	READING GUIDELINES	25

Part I COMPUTER BASED LEARNING METHODS

2	General definitions and notation	29
2.1	REPRESENTATION OF OBJECTS BY ATTRIBUTES	30
2.2	CLASSIFICATION PROBLEMS	31
2.2.1	Classes	31
2.2.2	Types of classification problems	32
2.2.3	Decision or classification rule	33
2.2.4	Learning and test examples	34
2.2.5	Learning a classification rule	35
2.3	REGRESSION PROBLEMS	35
2.3.1	Regression variables	36
2.3.2	Regression models	36
2.4	CLUSTERING PROBLEMS	37
2.4.1	Distances between objects in an attribute space	37
2.4.2	Attribute similarity	38
2.5	PROBABILITIES	39
2.5.1	General probabilities	39
2.5.2	Random variables	40
2.5.3	Classification	40

2.5.4	Entropies	41
2.5.5	Reliabilities	42
2.5.6	Standard sample based estimates	44
2.5.7	Various estimates of error rates	44
3	Machine learning	47
3.1	INTRODUCTION	47
3.2	GENERAL PRINCIPLES OF TREE INDUCTION	49
3.2.1	Trees	49
3.2.2	Tree hypothesis space	53
3.2.3	Top down induction of trees	55
3.2.4	Conclusions	62
3.3	MAIN VARIANTS	63
3.3.1	Variable combinations	63
3.3.2	Batch vs incremental learning procedure	64
3.3.3	Missing attribute values	65
3.3.4	Generalized “tree” structures	66
3.4	THE ULg METHOD	68
3.4.1	Description of a real illustrative problem	68
3.4.2	Quality evaluation	71
3.4.3	Optimal splitting	73
3.4.4	Stop splitting and pruning	82
3.5	OTHER CLASSES OF MACHINE LEARNING METHODS	91
3.5.1	Rule induction	92
3.5.2	Instance based learning (IBL)	94
3.5.3	Genetic algorithms	97

3.6	CONCLUDING REMARKS	100
4	Statistical methods	103
4.1	INTRODUCTION	104
4.2	PARAMETRIC METHODS	104
4.2.1	Linear discriminant functions	105
4.2.2	Quadratic and generalized linear discriminants	109
4.2.3	Conclusion	111
4.3	NONPARAMETRIC METHODS	112
4.3.1	The nearest neighbor class of methods	112
4.3.2	Projection pursuit	116
4.3.3	Other techniques	118
4.4	CLUSTERING METHODS	122
4.4.1	Algorithms of dynamic clusters	122
4.4.2	Hierarchical agglomerative clustering	124
4.4.3	Mixture distribution fitting	126
4.5	DATA PREPROCESSING	128
4.5.1	Pre-whitening	128
4.5.2	Feature selection	128
4.5.3	Feature extraction	130
4.6	CONCLUDING REMARKS	132
5	Artificial neural networks	133
5.1	INTRODUCTION	133
5.2	MULTI-LAYER PERCEPTRONS	134
5.2.1	Single layer perceptron	135
5.2.2	Multiple layer feed-forward networks	141

5.2.3	Other objective functions	146
5.2.4	Efficient network optimization algorithms	147
5.2.5	Network architecture and data pre-processing	153
5.2.6	Interpretations of neural network models	154
5.3	KOHONEN FEATURE MAPS	156
5.3.1	Unsupervised learning	156
5.3.2	Possible uses	159
5.3.3	Supervised learning	162
5.4	CONCLUDING REMARKS	163
6	Hybrid approaches	167
6.1	INTRODUCTION	167
6.2	MACHINE LEARNING AND NEURAL NETWORKS	168
6.2.1	Introduction	168
6.2.2	A hybrid decision tree - artificial neural network approach for power system security assessment	169
6.3	MACHINE LEARNING AND DISTANCE COMPUTATIONS	171
6.3.1	Margin regression	171
6.3.2	Nearest neighbor	173
6.4	DISCUSSION	174
7	Comparing supervised learning methods	175
7.1	CRITERIA	176
7.1.1	Computational criteria	176
7.1.2	Functional criteria	176
7.1.3	Evaluation methodologies	177
7.2	SURVEY OF METHODS	180

7.2.1	Three classes of methods	181
7.2.2	Synthetic comparison	182
7.3	RESEARCH PROJECTS	184
7.3.1	Description of the Statlog project	184
7.3.2	Other studies	185

Part II POWER SYSTEM SECURITY PROBLEMS

8 Physical problems 189

8.1	APPLICATIONS OF LEARNING TECHNIQUES	189
8.2	PHYSICAL PHENOMENA	191
8.2.1	Transient (angle) stability	191
8.2.2	Voltage security	196
8.3	PROBLEM FORMULATION	201
8.3.1	Prefault power system configurations	202
8.3.2	Classes of contingencies	204
8.3.3	Learning problems	205

9 Practical contexts 209

9.1	INTRODUCTION	209
9.2	OFF-LINE STUDIES	210
9.2.1	Planning	210
9.2.2	Operational planning	212
9.2.3	Training	213
9.3	ON-LINE APPLICATIONS	214
9.3.1	Normal operation	214

9.3.2 Under emergencies 215

9.4 COMPUTING ENVIRONMENTS 215

9.5 CONCLUDING REMARKS 217

10 Typical applications 219

10.1 ON-LINE PREVENTIVE SECURITY ASSESSMENT 219

10.1.1 Example problem statement 220

10.1.2 Data base generation 221

10.1.3 Security criteria learning 223

10.1.4 Comments 226

10.2 EMERGENCY STATE DETECTION 227

10.2.1 Example problem statement 228

10.2.2 Data base generation 230

10.2.3 Security criteria learning 231

10.2.4 Comments 232

11 Meaningful data bases 235

11.1 LOCAL NATURE OF SECURITY PROBLEMS 236

11.2 RANDOM SAMPLING OF STATES 237

11.2.1 Primary parameters 237

11.2.2 Free parameters 238

11.2.3 Topologies 239

11.2.4 Constraining the set of generated states 239

11.2.5 How many states should be generated 240

11.3 ALL SAMPLING TECHNIQUES ARE BIASED 241

11.4 HOW TO VALIDATE . . . TRULY 242

11.5 RELATIONSHIP WITH MONTE CARLO SIMULATIONS 243

11.6	CONCLUDING REMARKS	244
12	Modelling aspects and numerical tools	247
12.1	SIMULATION MODELS AND METHODS	247
12.1.1	Voltage security	247
12.1.2	Transient stability	249
12.1.3	Coping with model uncertainties	249
12.2	PHYSICAL ASPECTS OF LEARNING PROBLEMS	251
12.2.1	Problem decompositions	251
12.2.2	Security classes vs margins	252
12.2.3	Types of attributes	254
 Part III APPLICATIONS		
13	Transient stability	259
13.1	INTRODUCTION	259
13.2	ACADEMIC STUDIES	260
13.2.1	Study systems and data bases	261
13.2.2	General trends	265
13.2.3	Discussion	268
13.3	EDF SYSTEM	269
13.3.1	Study system and data base description	270
13.3.2	General parameters	276
13.3.3	Effect of attributes	280
13.3.4	Quality improvement	284
13.3.5	Multicontingency study	287

13.3.6	Other learning approaches	295
13.3.7	Summary	299
13.4	HYDRO-QUEBEC	300
13.4.1	Transient stability power flow limits	301
13.4.2	Study system and data base description	303
13.4.3	Global decision trees	310
13.4.4	Problem decompositions	313
13.4.5	Quality improvement	315
13.4.6	Other approaches	315
13.4.7	Discussion and perspectives	317
14	Voltage security	319
14.1	INTRODUCTION	319
14.2	ACADEMIC STUDY	320
14.3	PRELIMINARY INVESTIGATIONS	321
14.3.1	Preventive mode	321
14.3.2	Emergency mode	323
14.4	PRESENT DAY RESEARCHES FOR EMERGENCY MODE VOLT- AGE SECURITY	325
14.4.1	Data base generation	325
14.4.2	Overview of obtained results	332
14.4.3	Further investigations on contingency number 1	336
14.4.4	Hybrid approaches	341
14.5	MULTICONTINGENCY STUDY	342
14.5.1	Data base generation adaptations	342
14.5.2	Summary of generated data bases	343
14.5.3	Illustrations of load power margins	345

14.6 FUTURE PERSPECTIVES	347
15 Conclusions	349
Appendix - Uncertainty measures	353
A.1 MOTIVATION	353
A.2 GENERALIZED INFORMATION FUNCTIONS	353
A.2.1 Properties of H^β	354
A.2.2 Conditional entropies	355
A.3 SHANNON ENTROPY	356
A.3.1 Conditional entropies and information	356
A.3.2 Normalizations	358
A.3.3 Hypothesis testing	362
A.4 QUADRATIC ENTROPY	362
A.4.1 Conditional entropies and information	363
A.4.2 Normalizations	364
A.4.3 Hypothesis testing	364
A.5 OTHER LOSS AND DISTANCE FUNCTIONS	364
A.5.1 Kolmogorov-Smirnoff distance	365
List of Figures	367
List of Tables	371
Bibliography	376
Index	395
Glossary	399

Notation

The mathematical notation, used at several places in the context of the theoretical descriptions given in part1, is introduced in chapter 2. Other notations used more locally are introduced where they are used.

An index of references to frequently used notions as well as a glossary providing a list of acronyms, symbols and abbreviations are collected at the end of the manual.

1

Introduction

This chapter introduces the overall framework of the thesis. The basic notions and methods elaborated in the following chapters are considered here from a practical and intuitive point of view.

1.1 MACHINE LEARNING FOR SECURITY ASSESSMENT

Note. Generally, the term “machine learning” denotes a rather restricted subset of computer based learning methods (see §1.4.2 and chapter 3). Here we use it in a broader sense, to denote all types of computer based learning methods, including machine learning per se, as well as statistical pattern recognition and artificial neural network learning paradigms.

The overall methodology discussed in this work is based on the automatic synthesis of relevant security information from large sets of pre-analyzed cases generated off-line.¹ This is schematically represented in Fig. 1.1.

For a given security problem and a given power system, security cases are first generated

¹The meanings of “relevant” and “off-line” depend on the particular security assessment context and will be discussed later.

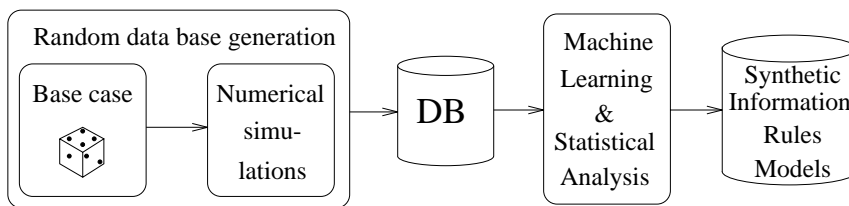


Figure 1.1 *Machine learning framework for security assessment*

via a random sampling approach, in a sufficiently broad and diverse range so as to screen all situations deemed relevant. Second, each case is pre-analyzed in terms of security by simulating numerically various possibly harmful contingencies. At this step massive parallelism may be exploited in order to speed up this off-line simulation phase, which is by far the most involved one from the computational point of view. The existing analytically oriented (“system theory”) methods for security assessment, used here as auxiliary tools, are presented below. An in depth discussion of approaches for the generation of representative data bases will be given in chapter 11.

The obtained data bases are typically composed of several thousands of cases for which security information was gathered with respect to several tens of disturbances. To exploit them properly, statistical learning techniques are used to extract the relevant information. The statistical techniques must be able to (i) identify the relevant attributes among those used to describe the system states, and (ii) build a model which explains the relationship among these attributes and the security status and/or which can be used to predict the security of new situations, different from those in the data base. Great flexibility is required for the choice of the interesting input parameters and the type of output security information.

Thus, the two main practical uses of the resulting information (rules and/or statistical models, correlation analyses, scatter plots . . .) are to help engineers obtain a better understanding of the security problems of their system [PA 85, WE 90a] and to make fast decisions in the context of real-time operation, for analysis and control [DY 68, ED 70, PO 72, PA 82, WE 90a].

Below, we will first discuss the security assessment problem(s) in general, introducing notation and problem classifications, and providing some indications on potential applications of the proposed framework in different security assessment contexts. Subsequently, important classes of learning problems and statistical methods used to synthesize the security information will be described in an intuitive way and their important characteristics in the context of security problems will be pointed out. Finally, the practical application of the approach will be illustrated on the basis of a hypothetical example.

1.2 AN OVERVIEW OF SECURITY PROBLEMS

In planning and operation of electric power systems, decision making is necessary in order to maintain a reliable and economic service in spite of a continuously changing environment. At the planning stage, tradeoffs are evaluated between cost of investment and security during future operation. Closer to the operation stage, outages for maintenance are planned and generation is allocated in order to achieve a minimum operation cost while minimizing the probability of service interruption. On-line, the operator has

to handle unforeseen events by adjusting the controls and topology of the system so as maintain its capability to cope with further disturbances, while maximizing economy.

Within these contexts, security assessment is concerned with the ability of a power system to withstand disturbances while preserving an acceptable operating condition. A disturbance is a planned or unforeseen event corresponding to changes in the parameters and/or structure of the system, such as an outage of a transmission or a generation equipment or a significant change in system loading. In this work we will focus on security problems involving *large* disturbances (or contingencies) corresponding to nonlinear system behavior. Although such disturbances are generally very unlikely to happen, their potential consequences can be extremely important, leading to complete system blackout.

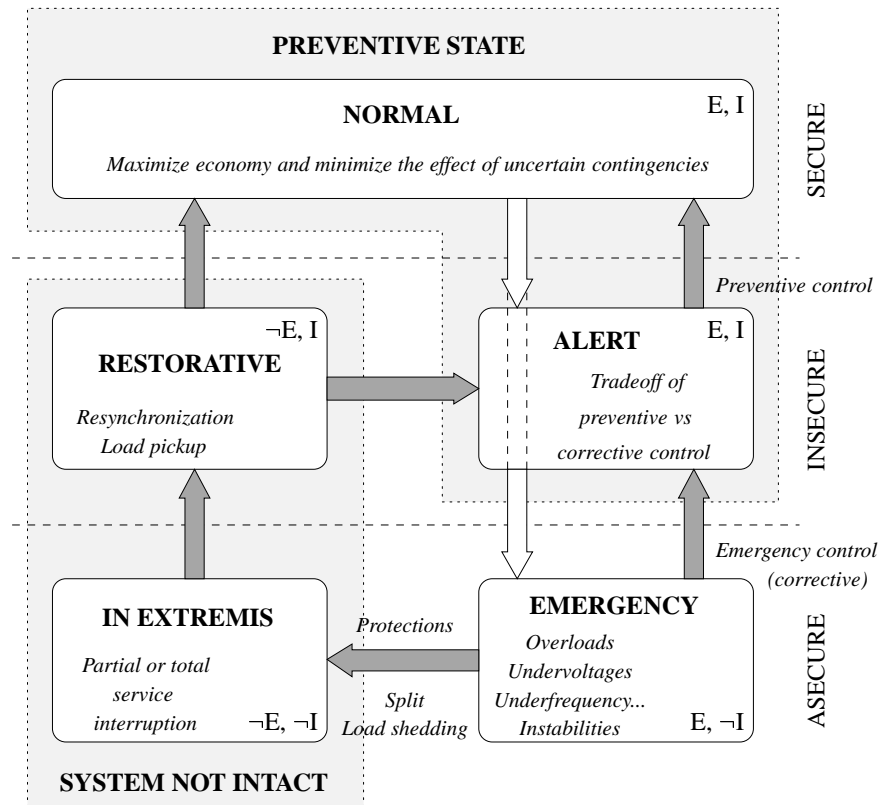
1.2.1 Classification of operating states

Figure 1.2 shows the different operating modes of a power system as identified by Dy Liacco [DY 67] in the late sixties.

Preventive security assessment is concerned with the question whether a system in its normal state is able to withstand every possible (or likely) disturbance, and if not, preventive control would consist of moving this system state into a secure operating region, by acting on the system controls or topology. Since predicting future disturbances is difficult, preventive security assessment will essentially aim at balancing the reduction of the *probability* of losing integrity with the economic cost of operation. In addition to yes/no type information about the ability of the system to withstand predefined contingencies, it is interesting to define various security *margins* and to appraise sensitivity coefficients of such margins with respect to important system parameters.

Emergency state detection aims at assessing whether the system is in the process of losing integrity, following an actual disturbance inception. This is a purely deterministic evolution, which involves very unusual situations and, while response time is critical, economic considerations become secondary. Thus the objective of emergency (or corrective) control is to take fast enough last resort actions, so as to avoid partial or complete service interruption. To achieve fast enough responses, most of the emergency control actions (e.g. generation rejection, load shedding, corrective switching) are presently designed in advance, either at the operational planning step or in the context of normal operation, during preventive mode security assessment. However, with the increased speed of computers and communication systems, a more important part of emergency control could be done in real-time, on the basis of real-time information on the pre-disturbance state and a fast enough disturbance identification (see chapter 9).

Finally, when both preventive and emergency controls have failed to bring system parameters back within their inequality constraints, automatic local protective devices




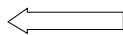
E : equality constraints  Control and / or protective actions
 I: inequality constraints  Foreseen or unforeseen disturbances

Figure 1.2 Operating states and transitions. Adapted from [F178]

will act so as to preserve power system components operating under unacceptable conditions from undergoing irrevocable damages. This leads to further disturbances, which may result in system islanding and partial or complete blackouts.

Consequently, the system enters the restorative mode, where the task of the operator is to minimize the amount of undelivered energy by resynchronizing lost generation as soon as possible and picking up the disconnected load, in order of priority. In this context expert system technology as well as simulation software may be valuable tools to assist the operator [IE 92a].

1.2.2 Physical classification of security problems

In this introduction we give a first glimpse at the different kinds of security problems further considered in chapter 8. Note that various security problems are distinguished according to the time scale of the corresponding dynamic phenomena and corresponding characteristic symptoms (low voltage, large angular deviations. . .) as well as the control means (reactive power, switching. . .) to alleviate problems. These aspects will indeed strongly influence the possible types of emergency control for a given problem, and, in particular, the best compromise between preventive and emergency control strategies.

Transient (angle) stability

The fastest security related phenomena are transient instabilities, which typically take of the order of a second to develop irrevocably. At this time scale, only a fully automatic emergency control strategy could be applicable, if any.

Transient *angle* stability assessment concerns the ability of the generators of a power system to recover synchronous operation following the electromechanical oscillations caused by a large disturbance. In this context, the dynamic performance is mainly affected by switching operations, by fast mechanical and active power controls (e.g. fast valving, high voltage direct current converters, flexible alternating current transmission systems (FACTS)), as well as by voltage controls (automatic voltage regulators of synchronous generators and static var compensators). Possible emergency control consists of varying generation by shedding some generators, or by modifying either their mechanical input power (e.g. by fast valving), or their electrical output power (e.g. via braking resistors, or - possibly in the future - via superconducting magnetic energy storage devices).

Although transient angle instability problems sometimes coexist with voltage ones, and may interact in the same time scale, they are driven by essentially different physical phenomena and characterized by different symptoms.

Voltage security

Transient *voltage* instabilities are characterized by sudden voltage collapse phenomena which may develop at the same or even faster speeds than transient angle instabilities. This is mainly due to an important proportion of fast recovery components in the load, such as industrial induction motors and DC links, for example.

The classical *mid-term* voltage instability problem corresponds to a typical time frame of one to five minutes. In this case voltage collapse is mainly driven by automatic transformer on-load tap changers which try to restore voltage nearby the loads. The available time for emergency control is still below the limit of operator response time

and most of the task should rely on automatic devices, such as under voltage tap-changer blocking schemes, reactive compensation switching (e.g. disconnecting shunt reactors) or fast generation unit start up.

It is important to mention that, although voltage collapse may result in a wide spread degradation of the voltage profile [HA 90], and also in subsequent losses of synchronism, it is initially a *local* problem, linked to a local deficiency in reactive power. The main causes of voltage instabilities following a large disturbance are insufficient local reactive power reserves and/or a reduced reactive power transmission capability. Thus, the voltage collapse phenomena are typically driven by a very important increase in reactive transmission losses following an outage, caused by high non-linearities. The risk of sudden voltage collapse further increases due to low voltage sensitivities of highly compensated loads and fast dynamic load recovery characteristics, acting together with machine (over)excitation limits. A recent survey of voltage collapse phenomena is given in [NO 91].

There is a third, even slower time frame, corresponding to the so-called *long-term* voltage instability, which involves the gradual buildup in load demand. This interacts with classical static security and is well within the scope of operator intervention.

Static security

Under the term *static* security one generally designates classical thermal overload problems of the generation transmission system components. These phenomena span over significantly longer periods of time. For example, line overloads may be tolerated during 30 to 60 minutes under favorable weather conditions. This allows time to rely on operators' decision making to correct overloads, provided that appropriate support is available from energy management system security functions, such as corrective switching and optimal power flow packages [GL 93].

1.2.3 Practical application domains

Table 1.1 shows the practical study contexts or environments which may be distinguished in security assessment applications. The first column identifies the study context; the second specifies how long in advance (with respect to real-time) studies may be carried out; the third column indicates the type of subproblems that are generally considered in a given environment; the last two columns indicate respectively if an operator is involved in the decision making procedure and if an expert in the field of power system security is available.

In the first three types of study contexts we currently rely mostly on the intervention of human experts and numerical simulation tools. But in the context of real-time monitoring and emergency control, the very reduced time scales call for more or less

Table 1.1 Security assessment environments. Adapted from [WE 93i]

Environm.	Time scales	Problems	Operator	Expert
System planning	1 - 10 years	Generation Transmission Protection	No	Yes
Operation planning	1 week - 1 year	Maintenance Unit commitment Protection settings	No	Yes
On-line operation	1 hour - 1 day	Preventive mode Security assessment	Yes	Partly
Real-time* monitoring	sec. - min. - hour	Emergency control Protective actions	No**	No
Training	months - days	Improve operator skill	Yes	No

* Here we distinguish between *real-time*, which considers dynamic situations following a disturbance inception, from merely *on-line* which considers static pre-disturbance situations.

** except for static security corrective control

fully automatic procedures, as already mentioned above. Below we will identify for each type of study the information synthesis approach, discussed in the following chapters, which could be useful.

System planning

In the context of system planning studies, multitudinous alternative generation/transmission system configurations must be screened for several load patterns. For each situation a large number of contingencies must be analyzed. An order of magnitude of 100,000 different scenarios per study would be realistic for a medium sized system.

Even though time is available in the context of planning studies, and even if security simulations may be achieved efficiently (e.g. if thermal overloads are considered) there is clearly room for improved data analysis methods in order to exploit all these simulation results, so as to identify the structural weaknesses of a system and provide information on how to improve its reliability.

Operation planning

As indicated in Table 1.1, operation planning studies concern a rather broad range of problems, such as maintenance scheduling (typically one year to one month ahead in time) and the design of operating guidelines in order to handle unusual or potentially weak situations (generation plants operating in radial configuration, primary protections

out of operation, very low/high loading). In the context of maintenance scheduling studies, the number of combinations of situations which must be considered is also generally very large, and data analysis approaches could equally well be used in order to summarize information and thereby make better use of it, for instance in order to be able to exploit the system with reduced margins.

Similarly, for studies closer to real-time, e.g. for the determination of operating security criteria, the machine learning approach seems particularly well adapted. It would allow us to systematically screen large and representative samples of situations, in order to identify critical operating parameters and to determine security limit tables needed for on-line operation. This merely consists of generalizing and automating the manual approaches presently in use at many utilities to solve this problem [RI 90].

The main advantage, however, of the automatic approach is that it will enable one to exploit easily the very rapidly growing computing power. While the manual approach becomes limited by the number of cases and the number of parameters an engineer is able to appraise simultaneously, the automatic approach would be able to scale up to hundreds of variables and thousands of operating states, provided that computing power is available in proportion.

On-line operation

On-line operation in the context of this framework, would consist of using on-line the rules or models previously derived in the context of operation planning, where one has to determine the range of operating situations for which the models should be valid. E.g. several weeks ahead routine security criteria could be designed for a forecast range of topologies, load levels and generation schedules. Closer to real-time, maybe one or two days ahead, these criteria might then be refreshed in order to handle more exceptional situations (e.g. high number of outages, very low load, protection failures, high transit wheeling ...).

In this context, it is particularly important for the models to be transparent and interpretable, so as to provide useful information compatible with the operators' view on the power system.

Real-time monitoring

For emergency control, machine learning type of approaches have been proposed for voltage security and transient stability problems [EU 92, RO 93].

Here, the purpose is to design a criterion in order to apply emergency control actions such as tap-changer blocking or generation shedding, so as to prevent the post-contingency system to evolve towards an in extremis situation. As we will discuss in chapter 11, an important aspect in this case is the use of appropriate models to reflect

the disturbed power system behavior. On the other hand, the use of readily available system measurements (e.g. EHV voltage magnitudes and/or phasor measurements) as input parameters is often an additional requirement, since state-estimation results are generally unreliable under highly disturbed conditions.

Training

During operator training, the security criteria derived in either of the preceding contexts might be used as guidelines for the operator, provided they are presented in an intelligible way. In addition, these models might be used internally in the training simulator program, in order to set up particular scenarios presenting particular insecurity modes.

1.3 ANALYTICAL TOOLS

A rather large set of numerical methods are available for security assessment, which are based on more or less accurate analytical models of the power system. Some tools, being based on general purpose power system dynamic simulation packages [ME 92, DE 92], have a very broad scope; others are based on simplified models and approaches aiming at the representation of only those features relevant for the study of a particular subproblem. The validity of the latter methods may be restricted to some particular physical phenomena and some particular (classes of) power systems. Below we give a brief overview of the most well known available tools, for each one of the security problems discussed above.

1.3.1 Transient stability

In addition to the machine learning approaches, which are within the scope of this thesis, there are two classes of tools for transient stability assessment : the time-domain or step-by-step (SBS) approach and the direct methods based on the second Lyapunov method.

Short-term time-domain simulation

The general power system dynamic model is composed of mixed algebraic and differential equations strongly non-linear, involving typically a few thousand state variables for real systems. Some have discrete time behavior while others have continuous time behavior. Reference [VE 92] gives an in depth mathematical analysis of stability problems of these kind of systems.

To assess stability for a given disturbance, the time-domain approach consists of simulating the during and post-fault behavior of the system and observing its dynamic performance. The simulation starts with the pre-fault system state as initial conditions, and the observation of the electromechanical angular and voltage swings during a few seconds allows assessment of stability. The practical criteria vary from one utility to another, but generally an unacceptable performance would imply large angular deviations (pole slips), and/or voltage and frequency variations. To obtain stability margins, repetitive simulations must be carried out for various pre-fault operating states and/or for various assumptions concerning the action of protection devices.

Nowadays, several industrial grade time-domain simulation packages are available for transient stability studies. Some of them use fixed integration step and explicit-partitioned solution algorithms, while others use variable step and simultaneous-implicit methods. The main asset of time-domain simulation tools is their flexibility w.r.t. models, which allows them to exploit with the same ease simplified and very detailed power system models. Until recently, they have been the only widely accepted method in use in the electric industry, for operation and operational planning.

The time-domain approach used to be considered as very CPU time consuming; it is interesting to observe that within the last three years the time required for a single simulation with high order models of a typical power system has shrunk from one hour to some minutes, essentially thanks to increased CPU speeds of high performance workstations.

Direct Lyapunov type methods

Direct methods aim at identifying when the system leaves its stability domain, without requiring further integration of the system trajectory. They therefore avoid the simulation of the post-fault trajectory, and require only simulation of the during fault trajectory. This reduces the simulated time period to a fraction of a second instead of several seconds used by the standard time-domain methods. In addition, these methods are expected to provide a stability margin without significant computational cost, and in some cases also sensitivity coefficients of this margin with respect to operating parameters. Most of these methods also provide information about the *mode of instability*, indicating which generators would lose synchronism. Such information may be exploited for the design of appropriate emergency control actions.

Thus, direct methods are, in principle, able to provide a rather rich stability assessment within a fraction² of the time required for a single time-domain simulation.

The major drawback of direct methods is related to difficulties in taking into account realistic models of generators, voltage and speed controls as well as non-linear and

²For the fastest direct methods the improvement is more than one order of magnitude, with respect to SBS using an equivalent model [XU 93a, GE 93a].

dynamic loads and devices such as SVCs. However, since the first multimachine direct methods, which were developed in the late sixties for the classical model³, much progress has been achieved in incorporating more sophisticated models. In particular, recently developed hybrid approaches are based on the coupling of more or less general purpose SBS simulations with energy function evaluations [MA 90, PA 89a, RA 91, XU 93b]. We believe that this kind of approach will eventually succeed in taking into account the main transient stability related modelling effects, while preserving most of the attractive features of direct methods.

1.3.2 Voltage stability and security

Tools for voltage security assessment range from simple purely static load-flow type calculations, to pseudo-dynamic and full short-term/mid-term time domain simulations. However, due to the rather recent emergence of voltage security problems, modelling practices have not yet reached maturity comparable to those used in transient stability studies.

In particular, one intrinsic difficulty of analyzing voltage collapse phenomena is the well known very strong dependence on load behavior for the modelling of which no good methodologies exist for the time being. Indeed, most of the load of a power system is composed of large numbers of rather small domestic and industrial customers connected to the distribution networks. Modelling the load at this level would however not be feasible due to computational intractability and the lack of data. On the other hand, building equivalent models is difficult due to the essentially variable nature (in time and in space) of the load.

Short-term/mid-term dynamic simulations

As we mentioned earlier, voltage collapse phenomena involve time constants ranging from a fraction of a second to a few minutes. Thus, for the sake of efficiency variable integration step methods with stiff system simulation capability are deemed necessary for time-domain simulations in the context of voltage stability studies [ST 93].

Although admittedly the time-domain simulation method is also here the reference tool, its usefulness may be limited due to the difficulty of determining appropriate models, and prohibitive computing times in the case of large scale systems.

³The classical model is the most simplified transient stability model, where the synchronous machines are represented by a constant electromotive force behind transient reactance and constant mechanical power, and all loads are taken as constant impedances.

Pseudo dynamic mid-term simulations

The fact that many voltage security problems are essentially driven by the automatic on-load tap changer (OLTC) mechanism rather than by fast interactions among load and generation dynamics, motivates the development of simplified pseudo-dynamic simulation tools in order to simulate these discrete OLTC dynamics, while filtering out the faster continuous short term transients.

In this case dynamic equations corresponding to the faster phenomena are considered to be at equilibrium during the simulation, and only the slower mechanisms such as OLTCs, machine excitation limits and secondary controls (voltage, frequency) are actually simulated [VA 93b]. With the limitation of being unable to highlight problems caused by the fast dynamics and their interaction with the slower ones, this kind of approach allows drastic reduction in computing times. It is thus liable to provide fast simulation tools for on-line operation, including load power margin computations and sensitivity analyses leading to emergency control applications [VA 93c].

Static load-flow type calculations

An important set of voltage security tools, based on purely static, load-flow type calculations, have been developed for security assessment in the context of system planning, operation planning and operation.

Typically, this kind of software allows us to compute maximal loading limits, based on successive computations [LE 90a] or direct optimization [VA 91a]. With up to date technology, this may typically be done in an efficient way, to allow systematic contingency evaluation⁴ within response time of some minutes.

In addition to these tools, approximate indices have been proposed for the fast screening and filtering of large sets of contingencies. For example, a clever application of fast *performance index* computation is proposed in [RE 93], allowing us to compute within the time required for 1 or 2 alternating current load-flow computations the post-contingency performance index for all single-outages.

1.3.3 Static security

Static security assessment has been one of the major concerns in many utilities in the last 20 years. Thus the field has acquired a certain maturity and, not astonishingly, many interesting tools have been developed, comprising simplified performance indices based on the direct current load-flow model for contingency ranking, as well as efficient

⁴Typically, the severity of a contingency is measured by the load power margin in the post-contingency state.

bounding techniques and full alternating current post-contingency optimal power flow and corrective switching programs [MI 81, CA 93a, BR 93].

These various methods may be combined to provide a satisfactory set of screening tools for the planning engineer and detailed security assessment modules for on-line operation to assist operators in taking decisions [ST 92, RE 92].

1.4 AN OVERVIEW OF LEARNING METHODS

In this section we introduce classes of potentially useful automatic learning methods for the synthesis of security assessment information, for the various physical problems and practical application contexts highlighted above. We will first give a definition of the generic *supervised* learning problem and introduce three important classes of algorithms for this problem, and finish with some comments on the use of *unsupervised* learning methods.

1.4.1 Generic problem of supervised learning

The generic problem of learning from examples can be formulated as follows :

Given a learning set of examples of associated input/output pairs, derive a general model for the underlying input/output relationship, which may be used to explain the observed pairs and/or predict output values for any new unseen input.

Input states are described or characterized by a vector of *attributes* or *features* assuming continuous or discrete values. Output is generally a scalar, with values belonging either to a finite set of mutually exclusive classes, or equal to real number in the case of regression problems.

In the context of security assessment, an example would correspond to a snapshot of a power system in a given operating situation. The input attributes would be (hopefully) relevant parameters describing its electrical state and topology and the output could be information concerning its security, in the form of either a discrete classification (e.g. secure / marginal / insecure) or a numerical value derived from security margins or indices.

In general, the solution of this overall learning problem is decomposed into several subtasks.

Representation consists of (i) choosing appropriate input attributes to represent the power system state, (ii) defining the output security information, and (iii) choosing

a class of models suitable to represent input/output relations.

Feature selection aims at reducing the dimensionality of the input space by dismissing attributes which don't carry useful information to predict the considered security information. This allows us to exploit the more or less local nature of many security problems (see chapter 11).

Model selection (or learning per se) will typically identify in the predefined class of models the one which best fits the learning states. This generally requires choice of model structure and parameters, using an appropriate search technique.

Interpretation and validation are very important in order to understand the physical meaning of the synthesized model and to determine its range of validity. It consists of testing the model on a set of unseen test examples and comparing its information with prior expertise about the security problem.

Model use consists of applying the model to predict security of new situations on the basis of the values assumed by the input parameters, and if necessary to "invert" the model in order to provide information on how to modify input parameters so as to achieve a security enhancement goal.

Solving the *representation problem* is completely left to the engineer, although there is a lot of research going on to develop automatic feature construction methods [ME 89]. In the context of power system security, a compromise has to be found between the use of very elementary standard operating parameters and more or less sophisticated compound features, known to show strong correlation with security. Ideally, the standard operating parameters would be preferable, but, depending on the problem and class of learning methods, this may lead to unsatisfactory performance, in terms of reliability. Thus choosing an appropriate set of candidate attributes is often done in an iterative fashion, during the first trials of applying a learning algorithm to a new security problem.

The distinction between *feature selection* and *model selection* is somewhat arbitrary, and some of the methods discussed below actually solve these two problems simultaneously rather than successively.

From the *interpretation and validation* point of view, as we will see, some of the methods provide rather black-box information, difficult to interpret, while some others provide explicit and very transparent models, easy to compare with prior knowledge.

Finally, as far as the *use* of the model for fast decision making is concerned, although speed variations of several orders of magnitude may exist between various techniques, all methods discussed in this work are sufficiently fast in the context of power system security analysis, taking into account the computing powers available in the security assessment environments. However, the methods producing their information in an explicit fashion are easier to exploit for control applications.

1.4.2 Classes of supervised learning methods

Below we introduce the three established families of learning algorithms, in the chronological order of their appearance. A very accessible description of these methods and a discussion of their practical uses are given in [WE 91f]. A more extensive discussion of a large number of algorithms and a very systematic comparative study are provided in [TA 94]. An introduction to a theoretical framework for studying learning algorithms is given in [AN 92].

While in the following chapters we will give a more technical unified description of those methods which we deem attractive in the context of power system security problems, here we will put the emphasis on relevant differences in the philosophies, and provide some basic bibliographic references for further discussions.

Statistical pattern recognition

Statistical pattern recognition⁵ methods are generally characterized by an explicit underlying probability model of the relation between inputs and outputs [DU 73]. The approach then consists of estimating the probability model from the learning data and using the probability model for decision making [BE 85].

Many of the modern methods have been developed in the context of signal processing applications, such as image and speech processing and letter recognition problems. Some of the discrimination methods used in pattern recognition have also been applied by statisticians for data analysis and modelling in economic and social sciences. Interestingly, almost all these methods have been applied to medical problems, such as blood cell counting and medical diagnosis.

Assuming that the joint probability distribution $p(i, o)$ of input/output pairs is known, we may, for any given input value i , compute the conditional probability distribution $p(o|i)$ or some relevant characteristics derived from this distribution. For example, for regression problems one would typically compress its information to one or two numbers such as the expected value and standard deviation, whereas for decision problems one could replace it, for a given loss-matrix, by the minimum expected cost decision.

Statistical methods come in two categories according to the assumptions made on the probability distributions and the corresponding technique used to estimate conditional probabilities $p(o|i)$.

Parametric methods. These assume a simple a priori known functional form of either

⁵The field of Pattern Recognition traditionally concerns the discrete case of classification or discrimination. Similar techniques, have been derived for regression problems, and will be discussed more in detail in chapter 4.

$p(i|o)$ or $p(o|i)$, which leads to linear or quadratic decision surfaces, together with various criteria for estimating their parameters.

Admittedly, these methods are hardly powerful enough to handle the large diversity of essentially non-linear power system security problems, although in some circumstances, when the underlying assumptions are valid, they may perform surprisingly well (e.g. see the comparative results given in chapter 14).

Non-parametric methods. These are *distribution-free* techniques, including generalized histogram methods, kernel estimators, k nearest neighbor ($K - NN$), and various series expansions of the probability density functions.

It is worth mentioning that in order to be effective, the non-parametric methods impose the use of regularity conditions on the estimated densities such as smoothness or complexity constraints, so as to prevent overfitting problems (see the discussion in chapter 4). The non-parametric methods are often rather black-box like, tending to provide only very limited insight into the problem structure, as compared to the parametric methods, but in the recent years more powerful techniques have been proposed, which combine up to a certain degree the non-parametric nature with data analysis features [FR 81, FR 84, FR 87]

In addition to the distribution estimation techniques, a number of statistical methods have also been designed for feature selection and extraction and for the estimation of classification error rates. We ask the interested reader to refer to [DU 73, HA 81, DE 82], for further information on this topic.

Machine learning

In the restricted sense, *machine learning* is the subfield of artificial intelligence which is concerned with the design of automatic procedures based on logical operators, which are able to learn a task on the basis of the observation of a learning set of solved instances of that task.

In the context of classification, the term *concept learning from examples* is used to denote the process of deriving a logical description (or *rule*) in some given representation language, of the - ideally - necessary and sufficient conditions corresponding to a class of objects. The stress is then often put on the use of powerful representation languages for the examples and the rules and an important part of the machine learning research has been devoted to the definition of appropriate search procedures, able to derive efficiently the appropriate rules.

To avoid overfitting, one of the major concerns of machine learning methods is to derive adequate compromises between rule *complexity* and data fit. An *Occam's razor*⁶ argument is used, to filter statistically unrepresentative variations observed in

⁶"Entities should not be multiplied unnecessarily" is the famous razor argument William of Occam

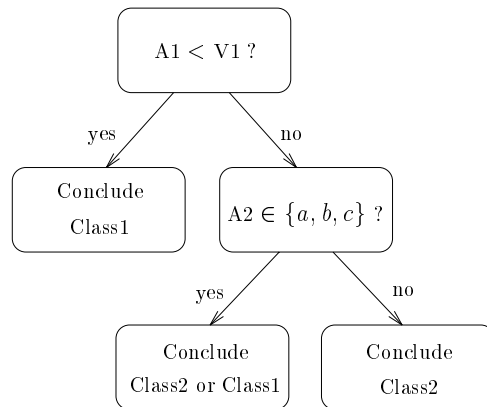


Figure 1.3 *Hypothetical decision tree*

the learning set. Quite interestingly, this is very similar to the regularization techniques used in non-parametric statistical procedures and artificial neural networks.

An important asset of machine learning methods lies in the explicit and logical representation they use for the induced classification rules, which, together with simplicity, provide a unique explanatory capability. One of the most successful classes of machine learning methods is the *top down induction of decision trees* (TDIDT) paradigm, initially popularized by Quinlan [QU 83]. This initially purely deterministic approach - developed for the classification of chess endgames - has evolved into a probabilistic approach and is now quite similar to the hierarchical (or sequential) methods developed by statisticians [MO 63, FR 77, BR 84]. As shown by the recent review given in [SA 91a], a very large number of variants of decision tree classification methods have been published since the early sixties. In chapter 3 we will discuss important aspects and provide a detailed description of our algorithm.

Figure 1.3 shows a hypothetical binary decision tree (DT). It is composed of two types of nodes: test nodes, including the top-node, correspond to dichotomous tests on some input attributes; terminal nodes correspond to a conclusion on the output value, such as class labels or conditional probability distributions. To infer the output information corresponding to a given input vector, one traverses the tree, starting at the top-node, and applying sequentially the dichotomous tests encountered to select the appropriate successor. When a terminal node is reached, the output information stored there is retrieved.

originally used against the superfluous elaborations of his Scholastic predecessors, and which was since then (around 1320) incorporated into the methodology of experimental science in the following form: given two explanations of the observed data, all other things being equal, the simpler explanation is preferable.

Thus decision trees essentially partition the input space into a finite number of hyper-boxes, to each one of which they attach a model for deriving the output information. In the very elementary case illustrated in Fig. 1.3, this model simply consists of class labels, but more complex models have been proposed, e.g. logistic models of conditional class probabilities (see chapter 6).

As suggested by the acronym, TDIDT methods approach the decision tree learning in a divide and conquer fashion, whereby a DT is progressively built up, starting with the top-node and ending up with the terminal nodes. At each step, a tip-node of the growing tree is considered and the algorithm decides whether it will be a terminal node or should be further developed. To develop a node, an appropriate attribute is first identified, together with a dichotomy on its values. The subset of its learning examples corresponding to the node is then split according to this dichotomy into two subsets corresponding to the successors of the current node. The terminal nodes are “decorated” with appropriate information on the output values derived from their learning examples, e.g. the majority class label.

To build good decision trees, an algorithm must rely on appropriate *optimal splitting* and *stop splitting* rules. Optimal splitting has to do with selecting a dichotomy at a test node so as to provide a maximum amount of information on the output value, whereas stop splitting has to identify situations where further splitting would either be useless or lead to performance degradation, due to overfitting. These aspects are discussed in more detail in §§3.4.3 and 3.4.4.

Artificial neural networks

The field of artificial neural networks (ANNs) started with the work on perceptrons in the early sixties, and has grown since the mid eighties to a very important and productive research field, involving quite diverse topics as for example the study of the biological plausibility of different network topologies and learning rules, the building of theoretical justifications, as well as practical hardware and software implementations, and - last but not least - the improvement of the practical learning algorithms.

In this introduction we will restrict our description to multi-layer perceptrons. Later on, in chapter 5, we will discuss another complementary technique, namely the Kohonen network [KO 90]. For further information, a widely recommended theoretical introduction to neural networks is given in [HE 91] while [ZU 90] gives a more exhaustive description of implementation issues of different types of networks and algorithms.

The perceptron, represented in Fig. 1.4, is basically a simple linear threshold unit together with an error correcting learning algorithm. It is able to represent a linear boundary in its input space.

Its limited representation capabilities have motivated the consideration of more com-

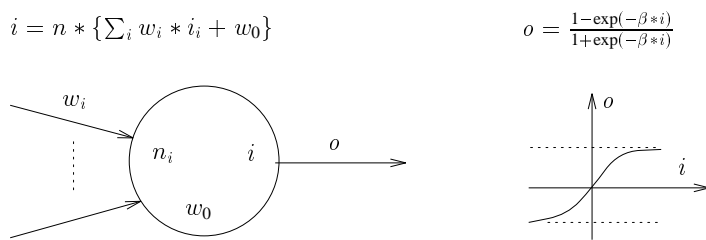


Figure 1.4 A soft linear threshold unit

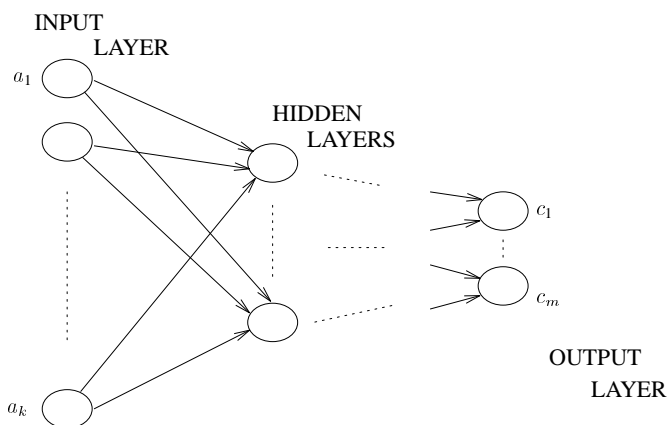


Figure 1.5 Feed forward multi-layer perceptron

plex models composed of multiple interconnected layers of perceptrons, MLPs for short. Figure 1.5 illustrates the classical feed-forward MLP. The first or *input* layer corresponds to the attribute values, and the last or *output* layer to the desired classification or regression information. Intermediate layers enable the network, provided that its topology and its weights are chosen appropriately, to approximate arbitrary “reasonable” input/output mappings.

The discovery of the back-propagation algorithm (see §5.2), allowing us to compute efficiently and in a local fashion the gradient of the output error of the network with respect to weights and thresholds, has been central to the success of MLPs [WE 74, RU 86]. This may be exploited iteratively in order to adjust the weights so as to reduce the total (or expected value) of the mean square error (MSE) for learning examples.

In recent years, much progress has been made in using more efficient optimization techniques for the learning procedures of MLPs, which have become a very popular type of neural network and have been used for many applications with quite promising results, particularly for multi-dimensional function approximation. An interesting property of the MLP is its ability to achieve feature extraction and learning in a single step : the weights connecting the input layer with the first hidden layer may be interpreted as projecting the input vector in some particular directions, realising a linear transformation of the input space, which is used in subsequent layers to approximate outputs.

However, one of the difficulties with MLPs comes from the very high number of weights and thresholds related in a non-linear fashion, which makes it almost impossible to give any insight on the relationship which has been learned. All in all, one can say that MLPs offer a flexible, easy to apply, but essentially black-box type of approach to function approximation. In the sequel we will give some illustration of possible problems with this type of approach.

1.4.3 Clustering and unsupervised learning

Unsupervised learning or clustering techniques will be discussed in chapters 4 and 5. In contrast to supervised learning, where the objective is clearly defined in terms of modelling the underlying correlations between some input variables and some particular output variables, unsupervised learning methods are not oriented towards a particular prediction task. Rather, they try to identify existing underlying relationships among a set of objects characterized by a set of variables.

One of the purposes of clustering⁷ is to identify homogeneous groups of similar objects, in order to represent a large set of objects by a small number of representative *prototypes*. Graphical, two-dimensional scatter plots may be used as a tool in order to analyze the data and identify clusters. It is interesting to note that the same techniques may also be used in order to identify similarities (and thus redundancies) among the different attributes used to characterize objects. In the context of power system security both applications may be useful as complementary data analysis and preprocessing tools.

Unsupervised learning algorithms have been proposed under the three umbrellas given above to classify classification methods. In the statistics literature the term clustering or cluster analysis is used [DU 73, HA 81, DE 82], in the machine learning community the term *conceptual clustering* is used to denote methods working with symbolic representations [MI 84, MI 86], while in the neural net terminology the term self-organizing networks or maps is often used [KO 90, ZU 90].

⁷In the latin languages the term "classification" is used to denote clustering, whereas "discrimination" is used to denote supervised learning.

1.5 A FLAVOR OF THE PROPOSED FRAMEWORK

1.5.1 Which methods should we combine in a tool-box ?

In the preceding sections we gave a first overview of typical security assessment problems and presented the motivation of providing *data management* and *data analysis* tools in order to exploit the fast growing computing powers. We concluded with a brief survey of the very numerous existing methods or techniques, able to extract relevant information from large statistical samples of security simulations.

Our own research work has mainly concentrated on the use of decision tree induction methods in this context of security assessment. But we believe that the complexity of security problems and their conflicting requirements - reliability, speed, interpretability - would prevent any single approach from providing a satisfactory overall solution. Consequently, our long term goal is to identify interesting *complementary* classes of methods and combine them in a *tool-box* approach in order to allow improved security assessment practices.

However, although keeping in mind that there is no universal panacea, we will argue that the data analysis and explanatory capabilities of decision tree based methods are paramount, and let them quite naturally assume a key role in such a framework, in order to enable cooperation between the engineer and the computer.

The importance of ongoing research in statistics, machine learning and neural networks, and in particular the cross-fertilization between these fields will certainly lead to the discovery of more powerful new techniques and an improved understanding of existing ones. Thus we should think about the development of a set of a representative power system security benchmark data bases, for the learning techniques, allowing researchers to carefully test new methods, as they appear, and compare them in terms of *performance* and *functionality* with the existing more mature techniques.

Although progress will certainly continue in the future, we will show that it is possible, with present day technology, to provide smart environments for security assessment, by building a tool-box where our learning methods will cooperate, together with numerical simulation tools and graphical man machine interfaces, with the engineer to derive better planning and operation strategies.

In the subsequent chapters we will discuss the kind of methods which may be useful from a technical point of view, and provide experimental evidence to support our conjectures. But before that, we will conclude our introduction with a hypothetical illustration of such a tool-box approach.

1.5.2 A hypothetical illustration of the framework

Note. The actual applications to real or academic security problems of the methods illustrated below will be discussed in later chapters, where appropriate references will also be given.

A security problem

Let us consider a hypothetical power system and suppose this system is voltage security limited in a weak - in terms of reactive power - area. We shall imagine that this security problem has been identified in previous screening studies, and that a set of possibly constraining disturbances have already been determined, such as some specific tie line or generation trippings.

For this system, a practical problem could be to characterize security regions with respect to combinations of these disturbances, in order to provide an operator with preventive security assessment criteria. A requirement would be that these criteria should provide information on the effective preventive control means, in case of insecurity.

Another objective could be the design of emergency state indicators applicable in case of a disturbance inception. Ideally, these indicators would provide a highly reliable and anticipative detection of the risk of voltage collapse and provide information on appropriate emergency control means, such as OLTC blocking and load shedding.

How could we generate a data base ?

In order to provide a representative sample of voltage security scenarios for the above problems, we would ask for the advice of planning and operation planning engineers and operators of that system, so as to gather information about known system weaknesses and operating practices.

From this information, data base building software would then be designed in order to generate randomized samples representative of normal operating conditions, including also a sufficient number of unusual situations, deemed relevant for security characterization. In particular, with respect to real-life operating statistics, this sample would typically be biased towards the insecure regions of the state space.

According to that sampling procedure, an initial data base would be generated, typically comprising two or three thousand states. For each state, the security would be analyzed with respect to each one of the studied disturbances. For example, a post-contingency load power margin could be computed with an appropriate simulation tool. As we will see later, this may be done within some hours response time, for real large-scale power system models on existing computer networks, by using efficient simulation software and exploiting trivial parallelism. In addition to this information, appropriate

preventive or emergency control information could be pre-determined for the insecure states.

Further, a certain number of attributes would be computed, which would be proposed as input variables to formulate security criteria. In the preventive mode security assessment problem, these attributes would typically be *contingency-independent* pre-fault operating parameters, such as voltages, reactive power generation and compensation reserves, power flows, topology indicators. For the emergency state detection problem, we would rather use raw system measurements (e.g. voltage magnitudes, power flows, transformer ratios, breaker status) of the intermediate *just after disturbance* state. In contrast to the preventive mode attributes, the emergency state attributes would depend on the disturbance and on the short-term load models, in addition to the pre-fault operating state.

Clustering methods for data pre-processing

In a security problem, many different attributes may actually turn out to provide almost equivalent information, due to the very strong physical correlations among geographically close components. Thus, for a class of similar variables, the problem arises of how to define a small set of representative attributes.

To fix ideas, let us consider the case of voltage magnitudes. One possible approach could be to compute correlation coefficients among any pair of bus voltages, on the basis of the data base statistical sample, and use these coefficients as similarity measures, together with clustering techniques so as to identify a small number of voltage coherent regions. For each group of busbars, a representative equivalent (e.g. mean) voltage would be used instead of individual voltages. This would be beneficial in terms of reducing the computational burden of the subsequent building of security criteria, and simplify the analysis of the results. Notice that similar applications of clustering techniques have been proposed in the power system literature, e.g. for the selection of pilot nodes for secondary voltage control [LA 89] and for the identification of coherent groups of machines in dynamic simulations [ZA 82].

Another appealing possibility, leading to a similar result, is to use the feature map of Kohonen, in order to analyze the relationships among these attributes. The comparison of the resulting feature map with the geographic location of busbars in the power system could provide interesting information (see §5.3).

In addition to the above “feature extraction” application, clustering techniques have been proposed, in a more conventional way, to identify groups of similar power system operating states. One possible purpose is to partition the overall data base into subsets for which the security assessment problem could be easier to solve. Another interesting application would be to “condense” the full data base to a smaller subset of representative prototypes. This would then reduce the number of required security

simulations, and shorten significantly the associated computing times.

What can decision trees do ?

Given a data base composed of examples, for which security margins have been determined for several contingencies and for which a number of candidate attributes have been computed, decision trees construction would proceed in the following way.

Data base partition. Split the data base into disjoint learning and test subsets.

Data base pre-classification. Define security classes. E.g. a state is considered preventively voltage secure if the security margin is sufficiently high for every disturbance.

Decision tree growing. Build a decision tree on the basis of the learning set. This includes (i) the automatic identification of the subset of attributes among the candidate attributes, relevant for the prediction of the security class, and (ii) the definition of appropriate threshold values for these attributes.

Decision tree testing. Compare the security classification predicted by the DT and the real classification for each test example and evaluate the proportion of non-detection of insecure states and of false alarms.

Iterate. If there are too many non-detections of insecure states, increase the threshold value used to define the secure class in terms of the security margin. If there are too many false alarms, propose better attributes or increase the number of learning states.

The building of the decision trees provides an approximate model of the voltage security region of the studied area of the power system. In addition to a global DT covering all disturbances simultaneously, single-contingency DTs may also be constructed to provide more specific information and additional insight. Further, various DTs may be constructed for various security margin threshold values, so as to discriminate between marginally secure and very secure situations.

What can neural networks add ?

In addition to the simplified view on security, provided by the DTs in terms of a discrete model relating a small number of security classes and thresholds on attribute values, one is generally interested in providing a continuous security margin, at least in the neighborhood of the threshold values used to define security classes.

As we have mentioned, one of the strong points of the MLP is its non-linear modelling capability. On the other hand, the decision tree identifies the attributes in strong

correlation with the security class. Thus, using the latter attributes as input variables to a MLP model, we may seek to approximate the security margin.

Once the weights of the MLP have been adapted, on the basis of the security margin information of the learning states, the MLP may be used for fast prediction of the margin for any seen or unseen example. Experiments with various security problems have shown that this leads to richer and more reliable security assessment information.

Further, the back-propagation algorithm may be adapted so as to compute automatically the sensitivity of the approximate security margin with respect to input attributes.

What do distance based methods offer ?

With the previous two approaches, we have essentially compressed detailed information about individual simulation results into general, more or less global security characterizations. This allows us to provide the required physical understanding, thanks to the data analysis component of decision trees and attribute clustering techniques. In addition, the derived models may be used efficiently for on-line security analysis.

In this latter context, additional information may however be provided in a case by case fashion, by matching the real-time situation with similar situations found in the data base. To achieve this matching, generalized distances must be defined so as to evaluate the similarities among power system situations, together with appropriate fast data base search algorithms.

Once the closest neighbors have been identified they may be used in multitudinous ways. For example, their distance to the current state may be used as a measure of the degree of confidence one may attach to the diagnostic provided by the DT and MLP models. If the latter distance was too large, it would be concluded that for the current state no reliable security information may be derived from the data base. If, on the contrary, the nearest neighbors are sufficiently close to the current state, then various kinds of detailed and specific security information may be extrapolated from these states to the current situation, and shown to the operator (see chapter 4.3.1).

1.6 READING GUIDELINES

The main objective of our work is the application of machine learning methods to power system security assessment problems. These methods have been briefly presented in the preceding section; they will be more thoroughly expanded below, in chapter 3 of Part 1. The consideration of other computer based learning methods in Part 1 has mainly a subsidiary threefold objective : to give an as unified as possible overview of existing methods, to justify a posteriori our choice of machine learning methods, and to

open perspectives for the possible combination of these with the other methods, used as complementary tools.

The reader interested exclusively in our main objective may skip chapters 4 and 5 of Part 1.

Part I

COMPUTER BASED LEARNING METHODS

2

General definitions and notation

In the following four chapters we provide a theoretical description of learning methods which we consider to be relevant to power system security assessment problems. We will mainly concentrate on those methods which are actually used in the applications discussed in chapters 13 and 14, for which we will also provide a detailed description of the algorithms used. For the remaining methods, used by other researchers or for which we have identified some prospective application possibilities, we will merely describe principles.

As mentioned in the introduction, the three approaches to (computer based) learning from examples are (i) statistical pattern recognition, regression or clustering, (ii) machine learning including concept learning from examples and conceptual clustering, and (iii) artificial neural network based learning. Although many of the theoretical and practical problems studied in these three fields are similar, and have received similar solutions, the three research communities have been relatively isolated in the past. For example, osmosis between the symbolic (and deterministic) oriented machine learning and statistical pattern recognition has begun only in the late eighties. On the other hand, statisticians and machine learning researchers have only very recently started looking at the new algorithms extensively developed in the last ten years within the artificial neural network paradigm.

Thus, the domains of interest of the three fields tend to overlap quite significantly, while interesting publications are spread over a large number of conference proceedings and journals. Moreover, important variations in terminology create an additional difficulty. We will therefore use a single notation and theoretical framework to describe methods from the three categories. Our terminology may sometimes seem unusual, since it is essentially a compromise. One of our aims has been to collect the sole relevant information, for the proper understanding of the subsequent discussions, while keeping the complexity of notation to a minimum.

In the present chapter, we introduce first the general definitions and notations used

throughout the following 4 chapters. In chapter 3, we start with the discussion of machine learning methods, and in chapters 4 and 5 we proceed with the consideration of complementary methods from the statistical and the neural network viewpoint, respectively. Finally, in chapter 6 we will comment on some interesting hybrid approaches, combining various frameworks. We will mainly concentrate on classification techniques for the prediction of security *status*, and on regression techniques for the approximation of *margins*, and to a much lesser extent discuss clustering techniques, which may be useful for data pre-processing.

An important topic is performance assessment. Thus, we will conclude this theoretical introduction with a short discussion, in chapter 7, of appropriate evaluation criteria and practices and give a brief review of some recent comparative studies.

2.1 REPRESENTATION OF OBJECTS BY ATTRIBUTES

Within the context of a learning problem we denote by

$$U \triangleq \{\text{all possible objects } o\}$$

the *universe* of possible *objects*. For example, in the context of preventive security assessment this could be the set of all possible normal pre-fault operating states of a power system.

Throughout this text, we will denote by an upper-case boldface letter, e.g. \mathbf{X} , any subset of U and $\neg\mathbf{X}$ its complement w.r.t. U .

We will use the term *attribute* to denote functions of objects which are defined on U . Thus

- $a(\cdot)$ denotes an attribute,
- $a(o)$ this attribute's value for object o , and
- $a(\mathbf{X})$ the set of all possible values it assumes in \mathbf{X} .

Further, for any subset of $\mathbf{V} \subset a(U)$ of values of $a(\cdot)$, we will denote by $a^{-1}(\mathbf{V})$ the set $\{o \in U | a(o) \in \mathbf{V}\}$ of objects.

Attributes are used to provide physical information on power system states, which is supposed to be useful for predicting security. We use lower-case boldface letters to denote vectors of attributes e.g.

$$\mathbf{a}(o) \triangleq (a_1(o), \dots, a_n(o))^T,$$

where n stands for the total number of different attributes, in a given context.

Attributes are in principle very general functions. Most of the time we will use scalar (numerical or qualitative) attributes, but occasionally more complex non-scalar data structures can also be considered as attributes.

In some instances, we will make the distinction between *attributes*, which will denote any defined function of objects, and *candidate attribute* which are those attributes which are being used as input variables for learning. We may also use the term *test attributes* or *selected attributes* to distinguish the subset of candidate attributes actually used in the learned rule.

2.2 CLASSIFICATION PROBLEMS

In the literature *classification* is used with two different meanings. In the case of *unsupervised learning* one looks at a set of data points and tries to discover classes or groups of similar points. In the case of *supervised learning* one is given a set of pre-classified data points and tries to discover a rule allowing us to mimic as closely as possible the observed classification. In our terminology, when we use the term classification, we are talking about *supervised learning*, which is also referred to as *concept learning from examples* or *discrimination*. We will use the term *clustering* rather than classification, to denote *unsupervised learning*.

2.2.1 Classes

In the context of classification problems, we will denote by

$$\mathcal{C} \triangleq \{c_1, \dots, c_m\}$$

the set of possible, mutually exclusive classes¹ of objects.

The number m of classes is in principle arbitrary but generally rather small. In the context of security assessment, classes will represent different levels of security of a system; they are often defined indirectly via security margins and some thresholds. In this case, we will denote by $\tau_1 < \tau_2 < \dots < \tau_{m-1}$ the $m - 1$ corresponding threshold values.

Since the classification of an object is unique, the following partition is defined on the universe

$$\{\mathcal{C}_1, \dots, \mathcal{C}_m\} : \mathcal{C}_i \triangleq \{o \in \mathbf{U} | c(o) = c_i\}, \quad (2.1)$$

where $c(\cdot)$ denotes the corresponding classification function defined on \mathbf{U} .

¹In the machine learning literature, the term concept is also used to denote a class of objects.

2.2.2 Types of classification problems

Deterministic vs non-deterministic

A classification problem is said to be deterministic if to any object *representation* corresponds a single possible class. Thus, the attributes can in *principle* be used to determine the correct class of any object without any residual uncertainty.

In practice, there are various sources of uncertainty which will prevent most of the problems from being deterministic. For example, in large-scale power system security issues it is generally not desirable to take into account every possible effect on security, due to simplicity constraints. Another example of non-determinism which is often neglected, is due to the limited accuracy of a real-time information system which provides attribute values. In some other circumstances, it is simply not possible to obtain a good knowledge of the system state in order to predict its future evolution, e.g. due to modelling uncertainties.

A trivial but fundamental property of non-deterministic problems, is the strong dependence of the theoretical upper bound on reliability of any classification on statistical distributions of objects. In particular, for an m -class problem this upper bound on reliability may be as low as $\frac{1}{m}$.

Diagnostic vs prediction

In addition to the above distinction, the notion of classification may come with different meanings, according to the type of physical problems considered.

Diagnostic problems. Classes correspond to different types of populations, which are clearly defined a priori. For example boys and girls form two mutually exclusive classes of children. In diagnostic problems, the possible values assumed by attributes are a causal consequence of the class membership. Although in principle perfect classification is possible, actual performance is often limited by the information contained in descriptive attributes.

Prediction problems. Classes correspond to some future outcome of a system, which is characterized by attributes obtained from its present state. Here, classes are a causal consequence of attributes, although one may distinguish between the deterministic case, where the class is a *function* of the attributes and situations where there exists some degree of non-determinism, either intrinsically or due to limited information contained in attributes.

Notice that there are intermediate situations where some attributes are causally posterior to the class while others are determined prior to it.

In the context of power system security, we mainly consider prediction problems, some being in principle deterministic and some others non-deterministic due to intrinsically limited information contained in attributes.

2.2.3 Decision or classification rule

Hypothesis space

A *decision rule* d , or *hypothesis* is a function assigning a value in \mathcal{C} to any possible attribute vector in $\mathbf{a}(\mathcal{U})$:

$$d(a_1(o), \dots, a_n(o)) \text{ or simply } d(o) : \mathcal{U} \mapsto \mathcal{C}. \quad (2.2)$$

In principle there is no loss of generality in assuming an identical decision and classification space. In particular, some of the classes \mathcal{C}_i may be empty, while corresponding to non-empty decision regions, and vice versa. This would allow the treatment of reject options and also to distinguish among sub-categories of classification errors.

A decision rule induces the partition $\{\mathcal{D}_1, \dots, \mathcal{D}_m\}$ on \mathcal{U} , defined by

$$\mathcal{D}_i \triangleq d^{-1}(c_i) = \{o \in \mathcal{U} \mid d(o) = c_i\} \quad (i = 1, \dots, m). \quad (2.3)$$

The *hypothesis space* \mathcal{D} is defined as a predefined set of candidate decision rules. Examples of hypothesis spaces are *the set of binary decision trees* or *the set of multi-layer perceptrons* (see chapters 3 and 5).

Rule quality

To learn a decision rule implies a search of the hypothesis space, so as to find a decision rule maximizing the chosen performance criterion.

To evaluate decision rules, we suppose that a quality measure $Q(\cdot)$ is defined, which assigns a real number $Q(d)$ to every decision rule in \mathcal{D} :

$$Q(d) : \mathcal{D} \mapsto] - \infty \dots + \infty [. \quad (2.4)$$

The higher the quality of a decision rule, the more appropriate is this rule for solving the classification problem. Appropriate quality measures will be defined later on, but in general a quality measure will combine different elementary evaluation criteria, selected among the following ones.

Reliability. The reliability (or accuracy) of a decision rule is a measure of the similarity of the partition it induces on U and the classification. Frequently, reliability is defined as the expected probability of misclassification, or more generally as the expected misclassification cost. We will use the notation $R(d)$ for reliability.

Cost of implementation. The complexity of implementing a decision rule may be another important aspect. This may involve the computational complexity of the algorithm used to apply the rule; it may also take into account the complexity of obtaining the attribute values (e.g. measurement cost).

Comprehensibility. If a decision rule has to be validated by an expert or applied by a human operator, then comprehensibility is often a key feature. The rather vague (and subjective) notion of comprehensibility is generally replaced in practice by a well defined (but also subjective) complexity measure. We will use the notation $C(d)$ to denote the model complexity. Examples of complexity measures are the number of nodes of a decision tree and the number of independent tunable parameters (weights and thresholds) of a multi-layer perceptron.

If we look more globally at the process of obtaining a classification or regression model in order to compare competing approaches, the following two aspects, related to preparatory work, become equally important.

Cost of data base collection. In our security assessment problems, the time required to generate data bases and running security simulations might become a practical limitation.

Complexity of learning. This corresponds to the computational requirements in terms of CPU time and memory, that must be fulfilled in order to learn a rule. In some real-time applications this may be a critical aspect and as we will see, there may exist variations of several orders of magnitude among different methods.

2.2.4 Learning and test examples

An example is a classified vector of attribute values corresponding to an observed or simulated object. The learning set LS is a sample composed of N different examples

$$LS \triangleq \{(\mathbf{v}^1, c^1), (\mathbf{v}^2, c^2), \dots, (\mathbf{v}^N, c^N)\}, \quad (2.5)$$

where the vector

$$\mathbf{v}^k = (v_1^k, v_2^k, \dots, v_n^k)^T = \mathbf{a}(o_k) \quad (2.6)$$

represents the attribute values of an object o_k and $c^k = c(o_k)$ its class.

Similarly, the test set TS is another, ideally independent, sample of size M . The test set is used in order to estimate the expected quality of a decision rule, once it has been

derived on the basis of the learning set. Generally, although not necessarily, both sets are drawn from the same sampling distribution.

In the sequel we will always assume that the objects of a learning or test set have been drawn independently; test and learning set based estimates of probabilities will be introduced in 2.5.

2.2.5 Learning a classification rule

The *apparent* quality $Q(d, LS)$ of a decision rule is the evaluation of its quality on the basis of a learning set. Thus, if the only information available for the choice of a classification rule is a learning set, learning will “merely” consist of searching \mathcal{D} for a rule d^* of maximum apparent quality. This implies in general the selection of an appropriate subset of the candidate attributes to be used in the formulation of the decision rule.

Clearly, this ideal situation is often not reached in practice. For example, one may be unable to compute the apparent quality, or one may be unable to reach the optimum rule. And even if the minimum apparent quality rule may be systematically reached, this may still produce inappropriate results with respect to the classification of unseen objects, because the quality measure may be inappropriate, or the hypothesis space too small, or the learning set not representative enough.

Learning algorithms are by definition inductive, since they aim at identifying a general model on the basis of a sample containing only part of the relevant information. Thus, the performance of a given algorithm for a given practical problem can only be determined empirically.

All learning methods are biased towards some particular problems. For example, the well known overfitting problem is an example where the quality measure is biased. Indeed, as we will illustrate later in our explorations, choosing a model of maximum *apparent* reliability, often (but not necessarily) leads to suboptimal *true* reliability. This has led researchers to use quality measures combining apparent reliability and model complexity (or prior credibility [BU 90, WE 90a]) or cross-validation techniques, but these are also biased [SC 93, WO 93].

2.3 REGRESSION PROBLEMS

In the context of supervised learning, in addition to classification, we will consider *regression* which aims at deriving a model for a continuous numerical value, rather than a discrete class.

2.3.1 Regression variables

We will denote by $\mathbf{y}(\cdot) = (y_1(\cdot), \dots, y_r(\cdot))$ an r -vector valued regression function, $\mathbf{y}(o)$ its value in the context of a particular observation and $\mathbf{y}(U)$ its range. Examples of regression variables in the context of security assessment could be various load power margins for voltage security, and various energy margins for transient stability.

We will use a similar notation for the learning and test samples in the context of regression, while replacing $c(o)$ by $\mathbf{y}(o)$. The above remarks concerning the non-determinism and the diagnostic or prediction type of problems apply equally to regression problems.

In the context of regression problems it may be interesting to distinguish real valued attributes from discrete ones, since continuity and differentiability requirements may be stated with respect to the former kind of attributes, while the latter would be considered as parameters of the regression model.

2.3.2 Regression models

To talk about learning a relationship between $\mathbf{a}(\cdot)$ and a continuous regression variable needs to introduce a *regression model*. Such a model, denoted by $\mathbf{r}(\cdot)$, is a function assigning a value in $\mathbf{y}(U)$ to any possible attribute vector in $\mathbf{a}(U)$:

$$\mathbf{r}(a_1(o), \dots, a_n(o)) \text{ or simply } \mathbf{r}(o) : U \mapsto \mathbf{y}(U). \quad (2.7)$$

We will denote by \mathcal{R} the space of candidate regression models.

In this context, learning often consists of a numerical optimization process adjusting the values of a certain number of weights. As for classification problems, evaluation criteria will generally take into account the accuracy and the model complexity. The apparent quality will be evaluated on the learning set and gradient techniques are often used in order to search for an appropriate regression model, maximizing the apparent quality.

An important practical difference between classification and regression is that in regression we essentially aim at modelling smooth input/output relationships whereas in classification we seek for a partition of the universe into a finite number of regions. Therefore, to avoid overfitting problems in the context of regression, the complexity term in the quality measure often aims at smoothing (or regularizing) the resulting model by penalizing high second derivatives.

2.4 CLUSTERING PROBLEMS

In our terminology we will use the term *clustering* to denote any type of *unsupervised learning*. Geometrically, unsupervised learning often aims at identifying clusters of similar objects or attributes. In the case of vector quantization applications, the purpose is to replace a large set of samples by a much smaller one, which is ideally chosen so as to minimize the overall quantization error. In both cases, the definition of similarity measures plays a central role. Another, modelling oriented way of looking at unsupervised learning, considers that the data are generated by a mixture of (unknown) probability distributions and aims at identifying a maximally plausible combination of such distribution laws chosen from a predefined catalog [DU 73, CH 88a].

In this work we consider mainly similarity based clustering and vector quantization approaches. Below, we define the type of distances between objects or attributes, used in the context of clustering as well as in the context of other nearest neighbor type of applications.

2.4.1 Distances between objects in an attribute space

Similarity based clustering requires the definition of a similarity measure. Intuitively, given a distance measure, the similarity of two objects will be inversely proportional to their distance, and although mathematically dissimilarity measures are slightly more general than distances (the triangular inequality does not necessarily hold for dissimilarities) in the context of object clustering we will restrict our discussion to distance based dissimilarity measures, which we define below.

The vector distance between two objects in the attribute space is defined by

$$\delta(o_1, o_2) \triangleq (\delta_{a_1}(a_1(o_1), a_1(o_2)), \dots, \delta_{a_n}(a_n(o_1), a_n(o_2))), \quad (2.8)$$

where $\delta_{a_i}(a(o_1), a(o_2))$ denotes a predefined scalar distance between the values of an attribute.

The definition of the distance between two attribute values depends on the attribute type. In particular, for a numerical attribute the (weighted) difference $\delta_a(a(o_1), a(o_2)) = w_a * (a(o_1) - a(o_2))$ is generally used, whereas for a symbolic attribute a difference table $\delta_a(v_i, v_j)$ is defined explicitly for each pair of possible values, such that $\delta_a(v_i, v_j) = -\delta_a(v_j, v_i)$ and $\delta(v, v) = 0$. In §3.5.2 we will describe approaches for the definition of appropriate difference tables, on the basis of a learning set.

Given the definition of a distance between attribute values, for each attribute, the k -norm of the vector distance defines the scalar distance, or simply distance, between

two objects

$$\Delta(o_1, o_2) \triangleq \sqrt[k]{\sum_{i \leq n} |\delta_{a_i}(a_i(o_1), a_i(o_2))|^k}, \quad (2.9)$$

where $k = 1$ for the *Manhattan* (or city-block) distance, $k = 2$ for the *Euclidean* distance and $k = \infty$ for the *maximum absolute deviation* distance.

Finally, the scalar distance between two sets of objects is accordingly defined by the lower bound “inf” of the distances between objects of the two sets

$$\Delta(\mathbf{X}_1, \mathbf{X}_2) \triangleq \inf\{\Delta(o_1, o_2) | o_1 \in \mathbf{X}_1 \wedge o_2 \in \mathbf{X}_2\}. \quad (2.10)$$

2.4.2 Attribute similarity

Similarity measures may also be defined between attributes, e.g. as generalized correlation coefficients.

Anticipating on the probability notation introduced below, we will define three different such measures, and their corresponding sample estimates.

Correlation coefficient. Used to measure the similarity between two *real* valued attributes. It is defined by

$$|\rho(a_1, a_2)| \triangleq \frac{|E\{(a_1 - E\{a_1\})(a_2 - E\{a_2\})\}|}{\sqrt{E\{(a_1 - E\{a_1\})^2\} E\{(a_2 - E\{a_2\})^2\}}}, \quad (2.11)$$

and estimated by

$$|\hat{\rho}^{LS}(a_1, a_2)| \triangleq \frac{|\sum_{o \in LS} \{(a_1 - \bar{a}_1)(a_2 - \bar{a}_2)\}|}{\sqrt{\sum_{o \in LS} \{(a_1 - \bar{a}_1)^2\} \sum_{o \in LS} \{(a_2 - \bar{a}_2)^2\}}}. \quad (2.12)$$

Spearman’s rank correlation. Used to measure the correlation between ordered, non-quantitative, attributes. Denoting by $rnk(a)$ the integer valued rank of an attribute value according to its predefined value order, the rank correlation is defined in terms of the correlation coefficient, by

$$|\rho_s(a_1, a_2)| \triangleq |\rho(rnk(a_1), rnk(a_2))|, \quad (2.13)$$

and estimated by eqn. (2.12), which reduces to the following formula if the ordering of the learning set provided by the two attributes is total

$$|\hat{\rho}_s^{LS}(a_1, a_2)| \triangleq \left| 1 - \frac{6 \sum_{o \in LS} \{(rnk(a_1) - rnk(a_2))^2\}}{N^3 - N} \right|. \quad (2.14)$$

This correlation coefficient is non-parametric in the sense that it is invariant with respect to any monotonic transformation of the attribute scaling.

Normalized mutual information. Different distance based measures may be used to compare symbolic attributes, in terms of the partitions they induce on \mathcal{U} . We will use a measure derived from information theory, on the basis of a normalization of the mutual information contained in two attribute values. This similarity measure is defined by

$$\rho_I(a_1, a_2) \triangleq \frac{2I_{a_1}^{a_2}(\mathcal{U})}{H_{a_1}(\mathcal{U}) + H_{a_2}(\mathcal{U})}, \quad (2.15)$$

where $I_{a_1}^{a_2}(\mathcal{U})$ denotes the mutual information of the two attributes, and $H_{a_1}(\mathcal{U})$ and $H_{a_2}(\mathcal{U})$ their uncertainty or entropy. These quantities and their estimates are defined below in §2.5.4. They yield an estimate of ρ_I defined by

$$\hat{\rho}_I^{LS}(a_1, a_2) \triangleq \frac{2\hat{I}_{a_1}^{a_2}(\mathcal{L}\mathcal{S})}{\hat{H}_{a_1}(\mathcal{L}\mathcal{S}) + \hat{H}_{a_2}(\mathcal{L}\mathcal{S})}. \quad (2.16)$$

2.5 PROBABILITIES

In this section we introduce some notation and considerations related to a probabilistic interpretation of the learning problems. Although learning may be defined in a purely deterministic fashion, as was the case with early machine learning and neural network formulations, it is now recognized that a probabilistic framework is practically unavoidable as soon as a certain level of generality is required.

From a more “impressionist” point of view, by using the probabilistic framework, we adopt right from the beginning the idea that the quantitative evaluation of *uncertainties* is one of the first issues in the context of learning problems, which admittedly calls for an explicit probabilistic treatment. Apart from these remarks, we will not discuss any other philosophical issues related to the use of probabilities.²

Note. Within the framework of general measure theory, modern probability theory allows an elegant and unified treatment of continuous, discrete and various mixed types of probability distributions [BI 79]. Within this theory, basic notions such as probability measures, random variables and conditional probability receive a precise although general meaning, allowing a rigorous mathematical treatment. In this work we don't aim at this level of rigor, and use probabilities in a naive and intuitive fashion, mainly as a notational tool.

2.5.1 General probabilities

For any $X \subset \mathcal{U}$, we denote by $P(X)$ the prior probability of observing an object of X , and $P(X_1|X_2)$ the conditional or posterior probability of an object to belong

²We refer the interested reader to [CH 85, PE 88] for discussions of the controversial subject of whether and which “probability theories” are appropriate to manage uncertainty.

to \mathbf{X}_1 given the information that it belongs to \mathbf{X}_2 . Assuming that $P(\mathbf{X}_2) > 0$, the conditional probability is defined by

$$P(\mathbf{X}_1|\mathbf{X}_2) \triangleq \frac{P(\mathbf{X}_1 \cap \mathbf{X}_2)}{P(\mathbf{X}_2)}, \quad (2.17)$$

To denote probability measures, we will use the notation dP or $p(a)da$, where $p(a)$ is the density function corresponding to a continuous probability measure.

2.5.2 Random variables

Roughly speaking, a random variable is a real-valued function defined on \mathbf{U} , e.g. an attribute or a regression variable, which maps probabilities initially defined for subsets of \mathbf{U} , to probabilities of subsets of the real line.

The random variable may be continuous or not, according to the continuity of the probability measure induced on the real line.

The expectation $E_P\{Y\}$ (or simply $E\{Y\}$) of a random variable y is defined by

$$E_P\{Y\} \triangleq \int_{\mathbf{U}} y(o)dP. \quad (2.18)$$

Similarly, the conditional expectation given the information that $o \in \mathbf{X}$ is denoted by $E_P\{Y|\mathbf{X}\}$, and defined by

$$E_P\{Y|\mathbf{X}\} \triangleq \frac{\int_{\mathbf{X}} y(o)dP}{P(\mathbf{X})}, \quad (2.19)$$

and the mean conditional expectation of y given the information about the value assumed by a function $x(\cdot)$ defined on \mathbf{U} is :

$$E_P\{Y|x\} \triangleq \int_{x(\mathbf{U})} \int_{o \in x^{-1}(x)} y(o)dP. \quad (2.20)$$

2.5.3 Classification

To simplify, we denote by $P^i(\mathbf{X})$ the conditional probability of \mathbf{X} given that the class $c(o) = c_i$, i.e.

$$P^i(\mathbf{X}) \triangleq P(\mathbf{X}|C_i). \quad (2.21)$$

To further simplify, we will denote by $\mathbf{p}(\mathbf{X}) = (p_1(\mathbf{X}), \dots, p_m(\mathbf{X}))$ the vector of conditional class-probabilities, defined by

$$p_i(\mathbf{X}) \triangleq P(C_i|\mathbf{X}), \quad (2.22)$$

and use $\mathbf{p} \triangleq (p_1, \dots, p_m)$ to denote the vector of prior class probabilities, $p_i \triangleq p_i(\mathbf{U})$.

2.5.4 Entropies

In the appendix we give a description of generalized entropy functions and related properties. Here we merely define some frequently used notions, related to the so-called logarithmic or Shannon entropy, used in information theory and thermodynamics. Unless specified otherwise, logarithms are computed in base 2.

The entropy associated to a partition of $\{\mathbf{U}_1, \dots, \mathbf{U}_p\}$ of \mathbf{U} is defined on any subset \mathbf{X} by

$$H_{U_1, \dots, U_p}(\mathbf{X}) \triangleq - \sum_{i=1, \dots, p} P(\mathbf{U}_i | \mathbf{X}) \log P(\mathbf{U}_i | \mathbf{X}). \quad (2.23)$$

The entropy is maximal in the case of uniform probabilities

$$H_{U_1, \dots, U_p}(\mathbf{X}) \leq - \sum_{i=1, \dots, p} \frac{1}{p} \log \frac{1}{p} = \log p, \quad (2.24)$$

and it is minimal in case of complete certainty

$$H_{U_1, \dots, U_p}(\mathbf{X}) \geq - \sum_{i=1, \dots, p} \delta_{ij} \log \delta_{ij} = 0, \quad (2.25)$$

where δ_{ij} denotes the Kronecker symbol defined by $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$, and the limit value $\lim_{x \rightarrow 0^+} x \log x = 0$ is assumed.

We will use the notation $H_C(\mathbf{X})$ to denote the classification entropy of a subset, defined by

$$H_C(\mathbf{X}) \triangleq - \sum_{i=1, \dots, m} P(\mathbf{C}_i | \mathbf{X}) \log p(\mathbf{C}_i | \mathbf{X}), \quad (2.26)$$

and $H_a(\mathbf{X})$ to denote the entropy of the partition induced by a (qualitative) attribute $a(\cdot)$, defined by

$$H_a(\mathbf{X}) \triangleq - \sum_{v \in a(\mathbf{U})} P(a(o) = v | \mathbf{X}) \log P(a(o) = v | \mathbf{X}). \quad (2.27)$$

Given two partitions $\{\mathbf{U}_1^1, \dots, \mathbf{U}_{p_1}^1\}$ and $\{\mathbf{U}_1^2, \dots, \mathbf{U}_{p_2}^2\}$, their joint entropy is defined as the entropy of the intersection partition $\{\mathbf{U}_{i,j} = \mathbf{U}_i^1 \cap \mathbf{U}_j^2 | i \leq p_1, j \leq p_2\}$

$$H_{U_{1,1}, \dots, U_{p_1, p_2}}(\mathbf{X}) = - \sum_{i,j} P(\mathbf{U}_{i,j} | \mathbf{X}) \log P(\mathbf{U}_{i,j} | \mathbf{X}). \quad (2.28)$$

Notice that

$$H_{U_{1,1}, \dots, U_{p_1, p_2}}(\mathbf{X}) = H_{U_1^1, \dots, U_{p_1}^1}(\mathbf{X}) + H_{U_1^2, \dots, U_{p_2}^2}(\mathbf{X})$$

only if the two partitions are independent in \mathbf{X} , i.e. if

$$P(\mathbf{U}_{i,j}) = P(\mathbf{U}_i^1)P(\mathbf{U}_j^2) \quad : \quad \forall i \leq p_1, j \leq p_2.$$

Otherwise,

$$H_{U_{1,1}, \dots, U_{p_1, p_2}}(\mathbf{X}) < H_{U_1^1, \dots, U_{p_1}^1}(\mathbf{X}) + H_{U_1^2, \dots, U_{p_2}^2}(\mathbf{X}).$$

Thus, the mutual information of two partitions, which is defined by

$$I_{U_1^1, \dots, U_{p_1}^1}^{U_1^2, \dots, U_{p_2}^2}(\mathbf{X}) \triangleq H_{U_1^1, \dots, U_{p_1}^1}(\mathbf{X}) + H_{U_1^2, \dots, U_{p_2}^2}(\mathbf{X}) - H_{U_{1,1}, \dots, U_{p_1, p_2}}(\mathbf{X}), \quad (2.29)$$

is symmetric by definition, equal to zero in case of independence, and positive otherwise. In addition, it verifies the following inequalities

$$I_{U_1^1, \dots, U_{p_1}^1}^{U_1^2, \dots, U_{p_2}^2}(\mathbf{X}) \leq H_{U_1^1, \dots, U_{p_1}^1}(\mathbf{X}), \quad (2.30)$$

$$H_{U_1^2, \dots, U_{p_2}^2}(\mathbf{X}), \quad (2.31)$$

$$H_{U_{1,1}, \dots, U_{p_1, p_2}}(\mathbf{X}). \quad (2.32)$$

Consequently, the normalization of the mutual information, may be done by dividing by either of the following quantities

$$H_{U_1^1, \dots, U_{p_1}^1}(\mathbf{X}), \quad (2.33)$$

$$H_{U_1^2, \dots, U_{p_2}^2}(\mathbf{X}), \quad (2.34)$$

$$\min\{H_{U_1^1, \dots, U_{p_1}^1}(\mathbf{X}), H_{U_1^2, \dots, U_{p_2}^2}(\mathbf{X})\}, \quad (2.35)$$

$$\max\{H_{U_1^1, \dots, U_{p_1}^1}(\mathbf{X}), H_{U_1^2, \dots, U_{p_2}^2}(\mathbf{X})\}, \quad (2.36)$$

$$\frac{H_{U_1^1, \dots, U_{p_1}^1}(\mathbf{X}) + H_{U_1^2, \dots, U_{p_2}^2}(\mathbf{X})}{2}, \quad (2.37)$$

$$H_{U_{1,1}, \dots, U_{p_1, p_2}}(\mathbf{X}). \quad (2.38)$$

To define similarities among partitions, the first two possibilities would be excluded, since they yield non-symmetric measures. Notice that only the last three measures are equal to 1, under the strict necessary and sufficient condition of perfect association between the two partitions.

2.5.5 Reliabilities

Decision rules

Give an $m \times m$ loss matrix \mathbf{L} , whose element L_{ij} defines the loss (or risk) corresponding to the decision c_j when the true class is c_i , the mean expected loss $L(d)$ of a decision rule is defined by

$$L(d) \triangleq \sum_{i=1}^m p_i \left[\sum_{j=1}^m L_{ij} * P^i(\mathbf{D}_j) \right]. \quad (2.39)$$

In the case of uniform misclassification cost, $L_{ij} = 1 - \delta_{ij}$, $L(d)$ reduces to the expected probability of misclassification $P_e(d)$, or the complement of the *reliability*

$$R(d) = 1 - P_e(d). \quad (2.40)$$

Another evaluation of the reliability of a decision rule is based on the entropy concept, in terms of the mean information provided by a decision rule on the classification,

$$I_C^d \triangleq I_{C_1, \dots, C_m}^{D_1, \dots, D_m}, \quad (2.41)$$

or one of the above defined normalizations. For example, we will use the *relative information* of a decision rule, defined by

$$RI_C^d \triangleq \frac{I_C^d}{H_C}. \quad (2.42)$$

Regression models

To evaluate a regression model, we will generally use the least squares criterion,

$$SE(\mathbf{r}) = E_P \left\{ \sum_{i=1, \dots, r} |R_i - Y_i|^2 \right\}. \quad (2.43)$$

A generalization of this criterion could be to use generalized distance (or similarity) measures (e.g. divergence, sum of absolute values . . .) to compare the output vector \mathbf{r} with \mathbf{y} .

Residual uncertainty

For any supervised learning problem, and for a given choice of object representation in terms of attributes, there exists a theoretical upper bound on performance, which could be reached if we knew for every possible attribute vector the conditional probability distribution of the output values.

Indeed, let $\ell(\mathbf{y}, \mathbf{r})$ denote a positive loss function. Then, if for every attribute vector \mathbf{a} we can determine the exact conditional probability distribution $P(\mathbf{y}|\mathbf{a})$ of \mathbf{y} and the conditional expected loss may be computed for any function $\mathbf{r}(\mathbf{a})$. Thus we may define the optimal function $\mathbf{r}_*(\mathbf{a})$, by [CH 91]

$$\mathbf{r}_*(\mathbf{a}) \triangleq \arg \min_{\mathbf{r}} E \{ \ell(Y, \mathbf{r}) | \mathbf{a} \}. \quad (2.44)$$

Provided that the above minimum value exists, this is well defined whatever the chosen loss function, expected loss, error rate, information, least squares error. We will use the term *Bayes rule* to denote the corresponding model $\mathbf{r}_*(\mathbf{a})$, and we will use the term *residual uncertainty* to denote its overall expected loss. This residual uncertainty, which is the inverse of the reliability, is thus defined by

$$L_* = E_{P(\mathbf{a})} \{ E \{ \ell(Y, \mathbf{r}_*(\mathbf{a})) | \mathbf{a} \} \} = \int_{\mathbf{U}} \ell(\mathbf{y}, \mathbf{r}_*(\mathbf{a})) dP. \quad (2.45)$$

2.5.6 Standard sample based estimates

We assume that the learning and test sets are statistical independent samples drawn from the probability distribution defined on U . We assume also that their classification is a priori given and correct, as well as their attribute values.

Thus, assuming no prior information on probabilities of events, we may estimate them by relative frequencies obtained by counting the occurrence of the events in either sample set. Some other estimates, taking into account information provided by non-uniform prior probability distributions are described in appendix A.5.

In the sequel, we will use the notation R^{LS} or R^{TS} for the learning and test set estimates of the reliability. If prior probabilities of a partition of U are given, we may sample separately the corresponding subsets, and build up estimates as weighted combinations of estimates within each subset, by the latter prior probability. For example, if prior class probabilities are known, we can build up estimates from samples of each class.

It is important to know that as soon as a learning set has been used to derive a decision rule or a regression model, any related estimates based on the learning set may become very unreliable. In particular, *apparent* reliability estimates are generally very strongly optimistically biased.

Unless other information is to be taken into account, prior probability estimates of subsets of U are given by relative frequencies of these subsets in the learning or test sets. These estimates are substituted within reliability and entropy functions, to obtain the corresponding test or learning set estimates. We use the “hat” notation to distinguish the latter estimates from their true values in U .

Expectation operators are replaced by sample means, unless specified otherwise. We use the “bar” notation to denote the sample mean of a random variable

$$\bar{x} \triangleq \frac{\sum_{o \in \text{Sample}} x(o)}{|\text{Sample}|}, \quad (2.46)$$

where $|\cdot|$ denotes the number of objects in a set.

2.5.7 Various estimates of error rates

Below we define briefly the various types of error estimates used in the context of our simulation results presented later. We kindly invite the interested reader to refer to the literature (e.g. [TO 74, DE 82, WE 91f] and the references therein) for a deeper discussion of the pros and cons of these methods. All these estimation procedures may be applied to any kind of reliability or cost measure used, with trivial adaptations. Below we merely describe the case of estimating classification error rates.

Resubstitution estimate

This consists of assessing a classification rule on the basis of the learning sample used in order to determine the criterion. Since the learning algorithms generally try to identify a rule of maximal (or high) apparent reliability, this estimate is generally strongly biased, and does not provide in most practical situations any valuable information about the ability of the rule to classify unseen situations.

Test set estimate

This consists of using an independent sample to assess a classification rule as was advocated above. The independent test sample states are supposed to be correctly classified by a bench-mark method (generally the same method which is used to classify the learning set). Their class is merely compared with the class predicted by the classification rule. This estimate is generally unbiased and similarly to the resubstitution error estimate, its computation is straightforward.

A major advantage of the test set error estimate is that its sampling distribution may be shown to be binomial, independent of the problem features, and for large sample sizes as we use in practice this distribution is very well approximated by the Gaussian distribution. Thus, confidence intervals may be derived from the test set error rates, and its standard deviation may be estimated by the following formula

$$\hat{\sigma}_{\hat{P}_e} \approx \sqrt{\frac{\hat{P}_e(1 - \hat{P}_e)}{M}}, \quad (2.47)$$

where \hat{P}_e denotes the test set error estimate and M the size of the test set.

In particular if M is sufficiently large a 95% confidence interval may be derived for the true error rate [DE 82]

$$Pr \left\{ \hat{P}_e - 1.96\hat{\sigma}_{\hat{P}_e} < P_e < \hat{P}_e + 1.96\hat{\sigma}_{\hat{P}_e} \right\} \approx 0.95. \quad (2.48)$$

For example, for a test sample size of 2000 and an estimated error rate of 3.0%, this interval is equal to [2.25% ... 3.75%].

Cross-validation estimate

Cross-validation methods aim at providing an unbiased error estimate when no independent test set is available. V -fold cross-validation exploits the learning set used to build a decision rule in the following fashion. The learning set LS is divided into V non-overlapping randomly selected sub-samples which are approximately of size $\frac{N}{V}$. Each one of these sub-samples is classified via the classification rule determined on the basis of the $V - 1$ remaining sub-samples.

This provides V unbiased estimates of the error rate of classification rules determined on the basis of a slightly smaller learning set than the classification rule. Provided that V is not too small (e.g. $V \geq 10$) and provided that each classification rule is determined with the same technique used to derive the original criterion, the average error rate of these rules will reflect closely the true error rate of the original rule.

The main disadvantage of this method is its high computational cost since it requires the repetitive learning of V different classification rules which may become overwhelming in the case of computationally intensive learning methods. If V is equal to the number N of learning states this method reduces to the well known leave-one-out method.

Choosing between the test set estimate and the cross-validation method is mainly a question of amount of available data. A rule of thumb is that below say 500 to 1000 available samples, dividing them into a test set and a learning set would either produce a too small test set, and thus high test set error estimate variances, or a too small learning set. Thus, we should probably prefer say 10-fold cross-validation and if less than 200 samples are available we could use the leave-one-out method [WE 91f].

3

Machine learning

3.1 INTRODUCTION

Machine learning is the subfield of *artificial intelligence* (AI) which provides essentially a symbolic perspective on learning algorithms. As in most AI research, machine learning has the twofold objective of modelling and understanding the corresponding psychological process on the one hand, and developing effective algorithms implementing this process on the other. One of the main motivations of the latter objective is the knowledge acquisition bottleneck encountered in the design of expert systems.

There are several sub-areas of machine learning, concerning for example learning by analogy, concept formation, discovery by experimentation, explanation based learning and finally concept learning from examples.

Concept learning from examples is the sub-area with which we are concerned. It aims at developing methods to derive a symbolic description of a class of objects, on the basis of a subset of examples (and counter-examples) of this class [MI 83].

Interestingly, the early work in concept learning was done by psychologists seeking to model human learning of structured concepts [HU 66]. This research has generated a whole family of decision tree induction systems, with the notable work on ID3 by Quinlan [QU 83] and on ACLS by Kononenko and his colleagues [KO 84, BR 88]. These methods have evolved towards a set of effective and rather mature techniques, yielding commercial implementations such as the CART software [BR 84] or the decision tree induction subroutine in the statistical package S, and the freely distributed IND package written by Buntine [BU 92]. An early large scale application of the decision tree methodology is reported in [LO 80].

In contrast to the decision tree induction techniques, other rule learning approaches have been much less successful, in particular due to their relative inefficiency in handling

large scale problems [CL 89, WE 91f, TA 94].

A second, slightly more recent trend within the machine learning research considers so-called *instance based learning* (IBL) methods, which aim at developing approximate matching techniques, in order to retrieve relevant information from similar cases stored in large data bases of examples. These methods are conceptually identical to the nearest-neighbor techniques of statistical pattern recognition. They aim, however, at an increased flexibility and generality, in particular in the context of high level symbolic and structural example description languages [ST 86, AH 91, SA 91b, CO 91]. Finally, another direction of investigation, which we will briefly comment in the last section of this chapter, concerns the work on genetic optimization and learning algorithms.

It should be emphasized that the earlier machine learning methods essentially aimed at representing deterministic concepts. The algorithms have been largely heuristically search driven, based on empirical ideas rather than theoretical motivations. Only quite recently, a certain unification with comparable work done by statisticians has emerged.

Notably, the book on *Classification and regression trees*, published by Breiman et al. in the mid eighties [BR 84], has been an important milestone in providing a theoretical (probabilistic) framework for the study of decision tree induction methods.

At about the same period, fundamental work has been done within the machine learning community around Valiant's *probably approximately correct* (PAC) learning theory¹ [VA 84]. Simultaneously, several papers were published around the idea of *minimum description length* (MDL) encoding of information and its use as learning criteria [RI 78, SO 83, RI 83, SE 85, QU 89, GA 89]. Finally, within the last few years the theoretical unification of these various frameworks has progressed significantly [BL 87, BU 89, BU 90] and resulting Bayesian frameworks have also been applied successfully within other learning paradigms, e.g. for artificial neural networks and non-supervised learning [CH 88a, BU 91].

Finally, although initially most of the work in AI considered purely binary truth values (True/False), the recent trend is clearly on incorporating appropriate techniques for handling uncertainties [CH 85, PE 88, BO 93]. Within the machine learning methods, this has led to a shift from the logical concept representations to the use of probabilistic models of attribute/class dependencies.

Nevertheless, in comparison to the statistical and neural network techniques, the machine learning methods still present the important characteristic of intelligibility, a consequence of their initial attempt to model human learning. On the other hand, we will illustrate in the sequel that their heuristic search approach to learning is a rather flexible framework, which is easily adaptable to various types of information, e.g. numerical vs symbolic, deterministic vs uncertain.

¹The term "probably approximately correct" of course applies to the learning, not to the theory, which is admittedly "provably absolutely correct".

Due to the importance of decision tree induction, as a subfield of machine learning, and due to their intensive study in the context of power system security problems, we will devote the first, most important part of this chapter to them, ending with a brief description of the other techniques which seem practically relevant, but which did not receive so far comparable attention, within this restricted application domain.

3.2 GENERAL PRINCIPLES OF TREE INDUCTION

Decision tree induction methods have been used for nearly three decades, both in machine learning [HU 66] and in applied statistics and pattern recognition [MO 63, HE 69].

3.2.1 Trees

Below we give some definitions and notation related to different types of trees, before introducing the *Top Down Induction of Decision Trees* (TDIDT) family of tree induction algorithms.

Graph and tree structures

A (finite) *graph* is a pair $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ composed of a (finite) set of nodes \mathcal{N} and a (finite) set of arcs \mathcal{A} , which are pairs of nodes. A graph is directed if the arcs are ordered pairs.

A *tree* is a connected acyclic finite graph. A tree is directed in the following way : (i) select a first node, and call it the top-node (or root², denoted by \mathcal{R}); (ii) direct all arcs containing the top-node outwards; (iii) proceed recursively, by directing arcs leaving the successor nodes of the root, until all arcs have been directed.

A non-trivial tree is a tree with at least two nodes. A node \mathcal{N}' of a non-trivial tree is a successor of \mathcal{N} if there is an arc $(\mathcal{N}, \mathcal{N}')$ from \mathcal{N} to \mathcal{N}' . Except for the root-node \mathcal{R} , every node of a directed tree is the successor of exactly one other node, called its parent node. Consequently, there is exactly one path from the root towards any other node of the tree. Graphs, trees and directed trees are illustrated at Fig. 3.1.

Nodes which have no successor nodes are called terminal, and denoted by \mathcal{N}_t . Non-terminal nodes are also called internal nodes, and denoted by \mathcal{N}_i . In the sequel we will assume that, apart from terminal nodes, the set $SUCC(\mathcal{N}_i)$ of successors of an internal node contains at least two nodes.

²Strangely, in Computer Science trees are structured upside down, maybe in order to differentiate them from trees in Botany.

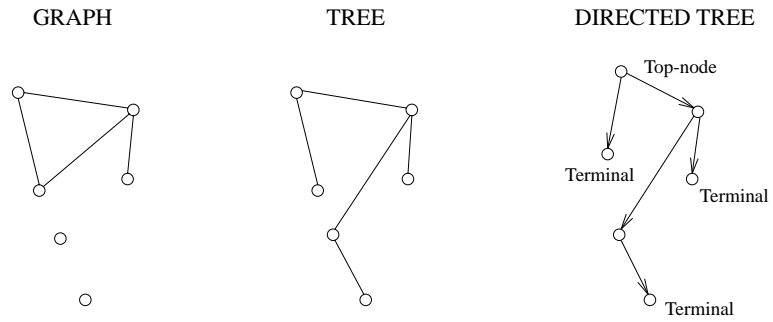


Figure 3.1 *Graphs, trees and directed trees*

We will denote by $DESC(\mathcal{N})$ the set of proper descendants of \mathcal{N} , which is recursively defined as the union of its successors and of all the descendants of these latter. The tree composed of the nodes $\{\mathcal{N}\} \cup DESC(\mathcal{N})$ and the arcs joining these nodes is called the subtree of root \mathcal{N} and denoted by $T(\mathcal{N})$.

Contracting a non-terminal node in a tree, consists of removing from the tree all the proper descendants of the node. A tree T' is said to be a pruned version of T if there is a subset \mathcal{N} of non-terminal nodes of T such that T' is obtained from T by contracting the nodes in \mathcal{N} .

Partitioning trees

A *partitioning tree* T is a directed tree each of which internal nodes has been decorated with a test $t_{\mathcal{N}_i}(\cdot) \in \{t_1, \dots, t_p\}$, defined on the space of possible attribute values of an object, $\mathbf{a}(\mathcal{U})$. Such a test has a - generally small - number of mutually exclusive and exhaustive outcomes t_i , each one of which is associated with a unique successor, i.e. corresponds to an arc leaving the test-node.

Thus, a test allows us to direct any object from a node to one of its successors on the basis of the attribute values of the object. Consequently, starting at the top-node, any object will traverse a partitioning tree along a unique path reaching a unique terminal node.

Let us define $\mathcal{U}(\mathcal{N})$, the subset of \mathcal{U} corresponding to a node \mathcal{N} of T , as the subset of objects traversing this node, while walking through the tree. Clearly, starting at the top-node and progressing towards terminal nodes, the tree defines a hierarchy of shrinking subsets :

- $\mathcal{U}(\mathcal{R}) = \mathcal{U}$, since all the paths include the top-node;
- for any internal node \mathcal{N}_i , the subsets of $SUCC(\mathcal{N}_i)$ form a partition of $\mathcal{U}(\mathcal{N}_i)$. For convenience, we will suppose in the sequel that these subsets are all non-empty;

- the subsets corresponding to the terminal nodes form a partition composed of non-empty and disjoint subsets covering U .

Similarly, for any subset X of U we will denote by $X(\mathcal{N})$ the subset $X \cap U(\mathcal{N})$.

Due to this correspondence, we will in many circumstances handle nodes of a partitioning tree as if they were subsets of U . In particular, we will talk about the probabilities of nodes and about objects belonging to nodes.

Decision, class probability and regression trees

A *decision tree* (DT) is obtained from a partitioning tree by attaching classes to its terminal nodes. The tree is seen as a function, associating to any object the class attached to the terminal node which contains the object.

Denoting by $c(\mathcal{N}_t)$ the class associated with a terminal node, \mathcal{N}_{t_i} the set of terminal nodes corresponding to class c_i , the decision regions defined by a DT are

$$\mathbf{D}_i = \bigcup_{\mathcal{N} \in \mathcal{N}_{t_i}} \mathbf{U}(\mathcal{N}). \quad (3.1)$$

In the deterministic case, these subsets should ideally coincide with the classification (i.e. $\mathbf{D}_i = \mathbf{C}_i$), and the number of corresponding terminal nodes should be as small as possible for each class.

A *class probability tree* (CT) is similar to a decision tree, but its terminal nodes are decorated with conditional class probability vectors. Ideally, these (constant) probability vectors would correspond to the conditional class probabilities $p_i(\mathbf{U}(\mathcal{N}_t))$, in the corresponding subsets of U . In addition, they should provide a maximum amount of information about classes, i.e. their residual entropy should be as close as possible to zero. This means that the tree should be designed so as to create terminal nodes where the class-probabilities would ideally be independent of the attribute values of an object.

Class probability trees may easily be transformed into decision trees. For example, given a loss matrix and a probability vector, we may use the minimum risk strategy to transform probability vectors into decisions, choosing at a terminal node the class c_{j^*} minimizing the expected loss

$$\sum_i L_{ij} p_i(\mathcal{N}_t). \quad (3.2)$$

However, in some situations it may be preferable to preserve the detailed information about conditional class probabilities, in particular when the loss matrix may change in time.

Finally, for *regression trees* (RT) the information stored at the terminal nodes should

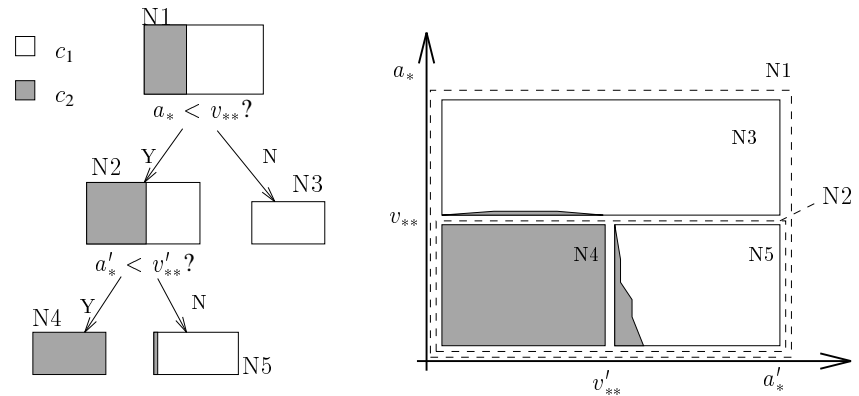


Figure 3.2 Example tree and attribute space representation

describe the conditional distribution of the regression variable, typically in the form of an estimate of its expected value and (co)variance (matrix).

Notice that class probability trees may be seen as a special kind of regression trees, where the regression variable \mathbf{y} is the class indicator variable, defined by

$$y_i(o) = \delta_{c(o),c_i}, \quad \forall i = 1, \dots, m. \tag{3.3}$$

The expected value $E\{\mathbf{y}|\mathbf{X}\}$ is then equal to the conditional class probability vector $\mathbf{p}(\mathbf{X})$.

However, real regression problems are generally characterized by smooth input/output relationships, whereas class probabilities may vary in a quite discontinuous fashion, in particular in the context of deterministic problems. Further, in the case of regression problems the value of \mathbf{y} (not only its conditional expectation) may generally vary continuously, while the class indicator variable may assume only a finite number (m) of discrete values.

In addition to the above types of trees, more sophisticated hierarchical models may be obtained by using more complicated test and terminal node decorations. For example, one may use fuzzy propagation functions at test nodes and more elaborate models to derive information from the attribute values of objects at terminal nodes. In particular, in the context of regression problems this could allow us to smooth the otherwise discontinuous information given by the trees. Such possibilities are further discussed in [WE 94b].

In the sequel we will simply use the term *tree* (T) to denote any kind of *decision*, *class-probability*, or *regression* tree. Figure 3.2 illustrates in its left part a simple two-class probability tree, and in its right part the corresponding sub-regions in the two-dimensional attribute space. The relative size of the white and grey parts of each

Table 3.1 Rules corresponding to the tree of Fig. 3.2

Rule N3 : if $[a_*(o) \geq v_{**}]$ then $[P(c(o) = c_1) = 1]$

Rule N4 : if $[a_*(o) < v_{**}]$ and $[a'_*(o) < v'_{**}]$ then $[P(c(o) = c_2) = 1]$

Rule N5 : if $[a_*(o) < v_{**}]$ and $[a'_*(o) \geq v'_{**}]$ then $[P(c(o) = c_1) = 1 - \epsilon]$

node represent the conditional class probabilities estimated at this node. The relative size of a box gives an indication of the probability of belonging to the corresponding region of the attribute space. The grey shaded area in the right part of Fig. 3.2 shows the actual region in the attribute space corresponding to class c_2 . Anticipating on a later discussion, we note that that Region N3 is not perfectly class pure although the terminal node N3 estimates $p_2 = 0$; this illustrates the possible biased character of probability estimates of trees.

Such a tree may be used to infer information about the class of an object, by directing it towards the appropriate terminal node. Starting at the top-node (N1), the attribute test $a_* < v_{**}$? corresponding to this node is applied to the object, which is directed towards the successor node corresponding to the outcome. At each test node a particular attribute value is tested and the walk through the tree stops as soon as a terminal node is reached. This will correspond to the elementary subset in the attribute-space comprising the object and the information stored there (e.g. class probabilities, expected value of the regression variable, majority class) is extrapolated to the current object.

A tree may be translated into a complete set of non-overlapping (mutually exclusive and exhaustive) rules corresponding to its terminal nodes. For example, the translation of the tree of Fig. 3.2 is given in Table 3.1.

3.2.2 Tree hypothesis space

In the context of classification or regression problems, we may define a hypothesis space of trees by defining a space of candidate test functions to be applied at the interior nodes, and a class of “models” (probability, classification, regression) to be attached to the terminal nodes.

Although most of the implementations of TDIDT methods use - on purpose - simple, rather restrictive hypothesis spaces, it is important to note that these methods may be easily generalized to more powerful hypothesis spaces. Anyway, the limitations of TDIDT approaches are probably more due to the weaknesses in search algorithms than to restrictions in representation languages. This is further discussed below and in

reference [WE 94b] which gives some suggestions for extending the current approach.

The first restriction generally put on the test functions is that they use only a single candidate attribute at one time, the reasons for this being efficiency of search (see below) and comprehensibility of the resulting tree. Thus, the definition of test functions reduces to the definition, for each candidate attribute, of a set of candidate partitions of its possible values. This is done in a generic fashion, defining types of attributes, and, for each type, a set of candidate splits.

Symbolic, purely qualitative attributes

A purely qualitative attribute represents information which is unstructured, i.e. the values of which may not be further compared among themselves.

If

$$a(\mathbf{U}) = \{v_1, \dots, v_p\}$$

is the set of possible values of the attribute, then, in principle, for any $k \in [2, \dots, p]$ all possible partitions into k non-empty subsets may be used as test functions. In practice only the two extreme cases, $k = 2$ and $k = p$, have been explored in the literature.

The binary option is preferable, since it is found to produce simpler, and more easily interpretable trees. This leads to $(2^{p-1} - 1)$ different tests of the type

$$a(o) \in \mathbf{V} ? \tag{3.4}$$

where \mathbf{V} is a non-empty subset of $a(\mathbf{U})$.

Unfortunately, the exponential growth of the number of candidate splits with p makes the traditional approach, consisting of enumerating and testing each candidate partition, questionable for values of p larger than say, 10 or 20. To handle qualitative attributes with a larger number of possible values, suboptimal heuristic search must be used in the optimal splitting procedure (see the discussion by Breiman et. al [BR 84] and Chou [CH 91]).

Symbolic, hierarchically structured attributes

More commonly, symbolic information concerns attributes such as shape or texture, the values of which are hierarchically structured. As is illustrated in Fig. 3.3, at each node of the hierarchy a small number of subclasses of possible values are defined.

Thus, candidate partitions may be defined at a given node of a tree by identifying the most specific subset in the hierarchy containing all values assumed by the attribute at this node. Only the direct subclasses of this subset will be used to define candidate partitions, which consist of adapting the “grain” of the attribute partitions to the considered subset of objects.

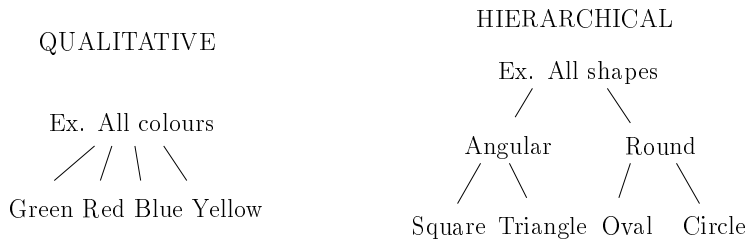


Figure 3.3 Partitioning of qualitative vs hierarchical attributes.

Let us for example consider the case of the shape attribute illustrated in Fig. 3.3. At a tree node containing all kind of objects we would use only the top-level partition distinguishing between “round” and “angular” shapes. On the other hand, at a node containing only “round” objects, we could use the distinction “circle” vs “oval” in order to split.

Ordered (integer, real valued or symbolic) attributes

Finally, a frequent kind of structure in attribute values concerns value *ordering* as it is for example the case for numerical attributes, used in most of the power system problems.

In this case, a set of threshold values v_i^{th} is defined corresponding to dichotomous tests of the form

$$a(o) < v_i^{th} ? \quad (3.5)$$

Some authors propose to use a small number of a priori fixed candidate thresholds [LI89]; this may however lead to high quantization errors and potential loss of discriminatory information. To overcome this difficulty, a better strategy consists of adapting the candidate thresholds to the distribution of values observed in the learning set (e.g. see §3.4.3).

3.2.3 Top down induction of trees

Quite a large number of variants of tree induction algorithms have been proposed in the past, not all of which fit perfectly to the generic TDIDT procedure which we will describe below. In the next section, we will give some bibliographical information on the variants which seem most relevant to us.

The basic TDIDT procedure is a greedy algorithm, building a tree in a successive refinement approach. The implicit goal of this iterative search is to produce an as simple as possible tree, providing a maximum amount of information about the classification or the regression variable of the learning examples. For instance, the objective of the initial

version of ID3 method was to build the most simple tree of minimum classification error rate in the learning set [QU 83].

In the more recent approaches, the tree building decomposes generally into two sub-tasks : tree growing which aims at deriving the tree structure and tests, and tree pruning which aims at determining the appropriate complexity of a tree.

Tree growing

During this stage the test nodes of the tree are progressively developed, by choosing appropriate test functions, so as to provide a maximum amount of information about the output variable. The objective is to produce a simple tree of maximal *apparent* reliability.

The basic idea is to develop one test-node after another, in an irrevocable top down fashion. The algorithm starts with the complete learning set at the top-node of the tree. At each step a test function is selected in order to split the current set of examples into subsets, corresponding to the current node's successors. This process stops when no further nodes need to be developed.

This is a locally rather than globally optimal hill-climbing search, which leads to a rather efficient algorithm the computational complexity of which is at most of order $N \log N$ in terms of the number of learning states and of order n in terms of the number of candidate attributes.

The basic ingredients of this algorithm are illustrated in Table 3.2.

Optimal splitting. This rule defines the criterion and search procedure in order to choose the best candidate test to split the current node. Essentially the preference criterion evaluates the capacity of a candidate split to reduce the impurity of the output variable within the subset of learning states of a node.

Stop splitting. This rule allows us to decide whether one should further develop a node, depending on the information provided in the current learning subset. For example if the local learning set is sufficiently pure in terms of the objective function values there is no point in splitting further. Another, less obvious reason for stopping a split is related to the so-called "overfitting" problem, which may occur when the learning set of a terminal node becomes too small to allow a reliable choice of a good split. This is further discussed below.

Tree pruning

The first tree induction methods reduce to the above growing procedure, essentially aiming at producing a maximum amount of information about the *learning* states. For

Table 3.2 *Hill-climbing tree growing algorithm***Given :**

- a learning objective function : a classification $c(\cdot)$ or a regression variable $y(\cdot)$;
- a set of candidate attributes defined on objects $a_i(\cdot)$;
- a learning set of examples, of known attribute values and known value of the objective function;
- an optimal splitting rule;
- a stop splitting rule.

Build : a tree with objective function statistics at its terminal nodes : class counts of $c(\cdot)$ or mean and standard deviation of $y(\cdot)$.

Procedure :

1. create a node, attach the current learning subset to this node, and compute the objective function statistics in this learning subset;
2. if the stop splitting rule applies, leave this node as a terminal node;
3. otherwise :
 - (a) apply the optimal splitting rule to find out the best test for splitting the current node, on the basis of the current learning subset;
 - (b) using the above test, decompose the current learning subset into subsets, corresponding to the p mutually exclusive outcomes;
 - (c) apply the same procedure to the newly created subsets.

example, in the context of classification it would try to split the training set into class pure subsets; in the context of regression it would try to define regions where the regression variable is constant.

Unfortunately this simple strategy is appropriate only in the context of deterministic problems with sufficiently large learning samples, which was the case in the chess endgame experiments of Quinlan [QU 83]. In the context of high residual uncertainty, or when the tree representation does not fit correctly to problem specifics, it produces overly complex, insufficiently reliable trees. In fact, for significant classification overlap in the attribute space the probability that the learning set is classified correctly by the optimal rule becomes very small as soon as the learning set size starts increasing.

This is the so-called *overfitting* phenomenon which may be explained intuitively. During the tree growing, the learning samples are split into subsets of decreasing size; if the method is unable to find splits which would allow us to reduce quickly the uncertainty

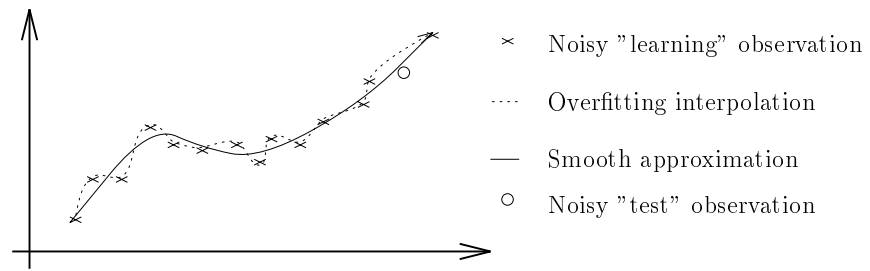


Figure 3.4 Illustration of overfitting

about the objective function, these sets may become extremely small and eventually shrink to one or two examples. Indeed, provided that there exist no learning examples of different classes with identical attribute values, even a random tree growing strategy will eventually “purify” completely the learning set. Unfortunately, if the learning subsets become too small, the statistical information which may be collected from these subsets becomes unreliable. Or, stated in another way, to be able to extrapolate the statistical information collected at the terminal nodes to unseen states, these subsets must be sufficiently representative.

The overfitting problem is well known from statistics and curve-fitting. For example, if we use a too large number of parameters in spline approximation, we may be able to fit to the small details or to the noise in the data, but overall the interpolation and extrapolation may become very poor. This is depicted graphically at Fig. 3.4 in the one-dimensional case. One can see that reducing the order of the approximation will actually allow us to reduce the overfitting.

In terms of decision trees, there exists a tradeoff between the two following sources of error: *bias* which results from insufficient splitting and *variance* which is a consequence of too much splitting. Too large trees will overfit the data, whereas too small ones will underexploit the information contained in the learning set. Thus, it is clear that some strategy is required so as to control the complexity of the tree and ensure that the learning samples at its terminal nodes remain sufficiently representative.

The first family of such “smoothing” strategies were actually proposed quite early in the tree induction history. Henrichon and Fu [HE 69], as well as Friedman [FR 77] proposed to put a lower bound $K(N)$ on the size of the terminal nodes learning set, increasing slowly with the learning set size N , i.e. such that

$$\lim_{N \rightarrow \infty} K(N) = \infty \text{ and} \quad (3.6)$$

$$\lim_{N \rightarrow \infty} \frac{K(N)}{N} = 0. \quad (3.7)$$

The main weakness of this “naive” approach is that it takes into account only the sample size related reason for stopping development of a terminal node, and will generally lead

Table 3.3 Hypothesis testing approach to pruning

-
- Given a statistic $S(\cdot, \cdot)$ measuring the correlation of two variables.
 - Let $f(S)$ be the sampling distribution of the statistic S under the hypothesis of statistical independence of the two variables.
 - Given an a priori fixed risk α of not detecting the independence hypothesis, determine the corresponding threshold $S_{cr}(\alpha)$, such that

$$\int_{S_{cr}}^{+\infty} f(S) dS = \alpha.$$

- Estimate the value of statistic $\hat{S}^{LS}(t_{\mathcal{N}}^*, y)$ applied to the objective function and the best candidate split $t_{\mathcal{N}}^*$ on the basis of the current node's learning subset.
 - If $\hat{S}^{LS}(t_{\mathcal{N}}^*, y) > S_{cr}(\alpha)$ reject the independence hypothesis, and split the node.
 - Otherwise, accept the independence hypothesis and stop splitting.
-

either to overly simple or to too complex trees.

Another possible reason for stopping to split a node is related to the discrimination capabilities of attributes. For example, in the extreme case where the attributes are “pure” noise, the “right” tree would be composed of a single top-node, whatever the size of the learning set. In most problems, of course, both sources of uncertainty may coexist up to a certain level, and a composite pruning criterion is required.

This consideration has yielded a second generation of pruning criteria, generally based on an hypothesis testing approach summarized in Table 3.3. Probably the first such method was proposed by Rounds, in terms of a non-parametric approach testing the significance of the Kolmogorov-Smirnov distance between the class conditional attribute value distributions [RO 80]. Later on, several conceptually similar but more flexible techniques have been proposed using various χ -square like statistics [KO 84, QU 86a, WE 89b].

Finally, the most recent generation of pruning approaches consider the complexity or overfitting control problem in a post-processing stage. In these methods, a tree is first grown completely and then simplified in a bottom up fashion, by removing its overspecified parts. The main reason for this new development was the difficulty with some of the above first and second generation stop splitting rules to adapt the thresholds (K, α, \dots) to problem specifics [BR 84]. However, we will see later on that there is a

Table 3.4 *Tree post-pruning algorithm*

1. Define a reliability measure $R(T)$ (e.g. amount of information, percentage of correct classification) and a complexity measure $C(T)$ of trees (e.g. number of nodes).

2. Define the global quality measure of a tree by

$$Q_{\beta}(T) \triangleq R(T) - \beta * C(T), \quad (3.8)$$

which expresses a compromise between the apparent reliability $R(T)$ and complexity $C(T)$, the latter being more strongly penalized for large values of β .

3. For β fixed, extract the optimally pruned tree $Pr^*(T, \beta)$ of T , such that $Q_{\beta}(Pr^*(T, \beta))$ is maximal, where $Q_{\beta}(T)$ is determined on the basis of the learning sample estimate of reliability. We will denote this as the β -optimal pruned tree of T .

4. Provided that the quality measure is additive in terms of decompositions of a tree into subtrees, a simple recursive bottom up algorithm will do the β -optimal pruning.

5. Moreover, for increasing β the trees form a nested sequence of pruned trees.

6. In particular, for $\beta = 0$ the pruned tree is the full tree; for $\beta \rightarrow \infty$, the pruned tree shrinks to a single node.

strong analogy between the stop-splitting and post-pruning approaches.

In the post-pruning approach a sequence of shrinking trees is derived from an initial fully grown one. One of these trees is then selected on the ground of its true reliability estimated honestly. Various methods have been suggested [BR 84, QU 87b, MI 89a, WE 93h], corresponding more or less closely to the pattern illustrated in Table 3.4.

Figure 3.5 illustrates a typical behavior of the complexity and the reliability (estimated on an independent test set) of the optimally pruned trees, as the value of β increases from 0 to ∞ . There exists an optimal value β^* of the complexity vs apparent reliability tradeoff, which leads to an optimally pruned tree of minimal estimated error rate. This overall tree selection procedure is summarized in a slightly more general version in Table 3.5, the last item of which is known as the *1 standard error rule*.

In the sequel we will use the term *pruning set* PS , to denote the set of classified objects which is used to evaluate and select pruned trees. It is indeed necessary to

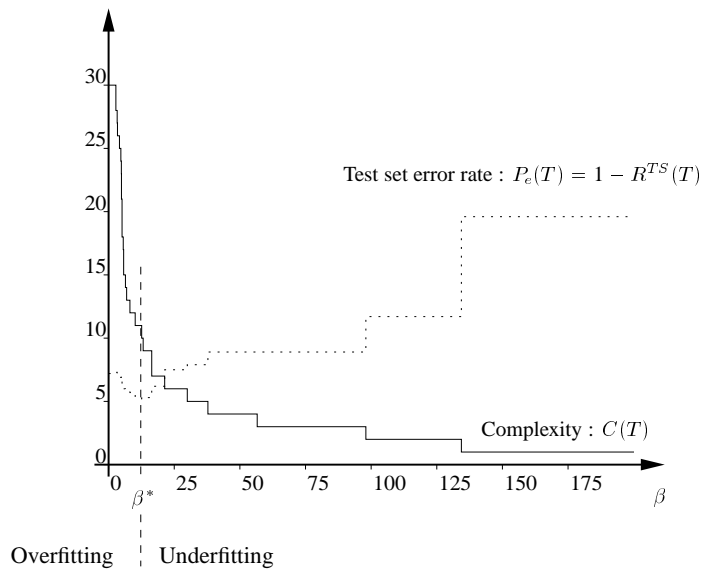


Figure 3.5 Characteristics of pruned trees for increasing β

Table 3.5 Pruned tree selection algorithm

1. Define a tree quality measure $Q_\beta(T)$.
2. Let β increase from 0 to ∞ , and generate the corresponding sequence of β -pruned trees, $Q_\beta(T)$ being estimated on the learning set.
3. Compute an unbiased estimate of the latter trees' reliabilities (e.g. on the basis of an independent set of pre-classified examples); let P_e^* be the corresponding minimal error rate estimate, and let σ denote an estimate of its standard error.
4. Select the final tree T_s in the sequence as the tree of minimal complexity, and such that $P_e(T_s) \leq P_e^* + \sigma$.

distinguish this set from the true *test set* TS which is supposed to be truly *independent* of a tree, and may be used to provide unbiased estimates. Although in many practical situations the error rates obtained on the pruning set are found to be unbiased, there is no guarantee and the bias of this estimate may well depend on the pruning algorithm or on the selection rule. Clearly, the “1 standard error rule” prevents the selected tree from fitting too perfectly the PS and thus is in favor of a low bias.

3.2.4 Conclusions

We have discussed above the three major subproblems within the general framework of tree induction, concerning (i) the choice of candidate splits in order to develop the test nodes of a tree, (ii) the hill-climbing tree growing algorithm, and its optimal and stop splitting rules, and finally (iii) the various pruning strategies proposed to cope with difficulties related to overfitting.

The list of variants to which we have referred gives only a limited - and far from exhaustive - account of all the research work done in this field for almost three decades. In the last few years several comparative studies have been published, looking at various aspects of the methodology, from theoretical and practical viewpoints. Quinlan's synthesis of the work done until 1986 is very informative [QU 86b] and may be usefully complemented by the rather extensive review of tree methodologies given by Safavian and Landgrebe [SA 91a]. Finally, in his Ph.D. thesis, Buntine has made some very incisive theoretical contributions from the Bayesian point of view [BU 90].

From the practical side, Mingers has made an extensive comparison of splitting criteria in [MI 89b] and pruning approaches in [MI 89a]. Within the recently completed ESPRIT project Statlog, extensive simulation studies have been carried out on 23 different practical problems, including two of our power system security data sets, comparing as many as 22 classification methods, including 5 decision tree induction algorithms. Let us quote some of their conclusions concerning decision trees [TA 94].

There is a confusing diversity of Decision Tree algorithms, but they all seem to perform at about the same level. There are no indications that this or that splitting criterion are best, but the case for using some kind (!) of pruning is overwhelming, although, again, our results are too limited to say exactly how much pruning to use . . .

Similar impressions are reported in several other publications [BR 84, MI 89a, MI 89b].

Our experience in the context of power system security problems is that pruning allows us to significantly reduce tree complexity (frequently by factors of 3 or more) while preserving near optimal reliability. Further, considering that one of the main objectives of tree induction is to provide easily interpretable information, for the purposes of data exploration and analysis, the simplicity becomes an even more important feature of the trees.

3.3 MAIN VARIANTS

Before proceeding with the description of the decision tree algorithm which we have developed for our experiments in the context of power system problems, we will briefly discuss some other interesting questions about possible variants or enhancements of the standard TDIDT method.

3.3.1 Variable combinations

The computational efficiency of the TDIDT algorithm is due to the fact that it searches in a reduced space of candidate trees, developing one node at a time and looking at a single attribute at a time. While this works nicely in many problems, in some situations it may be inappropriate and tend to produce very complex trees of low reliability.

In this case, one possible enhancement of the method may be to combine several attributes in a single test while splitting a node. For example, numerical attributes may be combined in a linear combination and binary attributes in a logical combination. The appropriate combinations might be chosen a priori, either manually or by using standard statistical feature extraction techniques [DU 73, FR 77]. They may also be determined automatically at the time of developing a node, taking advantage of the tree growing approach to adapt the optimal combination at each tree node. Various, more or less complex strategies may be thought of in order to define an appropriate set of variables to be combined and to choose the parameters defining their combination. For example Breiman et al. propose to use a sequential forward selection procedure [BR 84]. Utgoff [UT 88] has proposed to build decision trees with perceptrons implementing linear combinations used to predict classes at the terminal nodes; similar techniques used to define linear combinations at the test nodes are also discussed in [MU 93, WE 94b].

Another, complementary possibility would be to use look ahead techniques, so as to search for high order correlations among *several* attributes and the objective function, while keeping the “single attribute per test node” representation. The latter approach would be appropriate if symbolic attributes are important.

To fix ideas, Figure 3.6 illustrates two examples where these strategies could be useful.

In the left part of Fig. 3.6, a two step look ahead technique would allow us to identify the optimal decision tree, comprising four terminal nodes. The regions shown on the diagram correspond to the partition obtained by the standard (one step look ahead) TDIDT method described above. The first split at the root of the tree actually depends strongly on random variations in the learning set. Nevertheless, the resulting approximation, although overly complex, remains perfectly accurate. For example, in a simulation the tree obtained from the 1000 learning states shown on the diagram yields indeed a 100% correct classification on an independently generated test set.

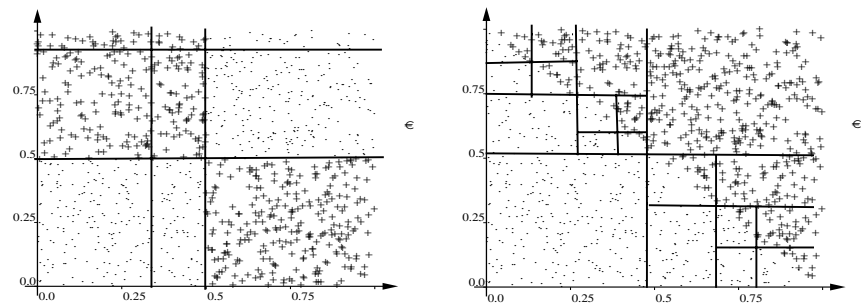


Figure 3.6 *Difficult examples for the standard TDIDT approach*

For the problem illustrated in the right part of Fig. 3.6, the difficulty of the TDIDT approach is not related to its suboptimal search procedure but rather to fundamental limitations in the representation capabilities of standard trees. As shown by the staircase approximation, the resulting standard TDIDT tree is likely to be quite inaccurate. In a simulation similar to the above one, such a tree was obtained and resulted in 97-98% reliability in the test set. On the other hand extending the basic tree procedure so as to search for linear combination splits allowed us again to reach 100% accuracy.

Notice that several limitations exist in the above possibilities of enhancing trees. The first, rather obvious one is related to computational costs. Clearly, look ahead search time will increase exponentially with the number of combined attributes; similarly the time required to determine the linear combinations rapidly increases with the number of combined attributes.

Another, and at least as important drawback, is related to the fact that the interpretability of the trees will rapidly decrease if too complex tests or search criteria are used. A final limitation is due to the overfitting problem which may become worse when too powerful - almost exhaustive - search techniques are used. In particular current pruning counter measures may require adaptations so as to remain effective [WE 94b].

3.3.2 Batch vs incremental learning procedure

In the above description we have assumed the *batch* learning approach, where the complete learning set is required when the tree building starts, and is used at each step to take the splitting, stop splitting and pruning decisions.

This is appropriate when all the learning states are available at the same time. However, if the learning states become available in a sequential fashion, then an incremental scheme is more appropriate. This allows the tree building to start as soon as the first observations are obtained, beginning with a very simple approximate model. Subsequently, the tree structure is enhanced with more details when additional information

Table 3.6 *Weighted object propagation*

-
1. Let o be an object, and $w(\mathcal{N}, o)$ its weight in a tree node \mathcal{N} .
 2. The weight at a successor node of \mathcal{N} is obtained by $w(\mathcal{N}', o) = w(\mathcal{N}', \mathcal{N}, o) * w(\mathcal{N}, o)$, where $w(\mathcal{N}', \mathcal{N}, o)$ denotes the arc strength.
 3. Initial weights at the top-node would be usually equal to one. However, available prior information may also be used to bias initial weights.
 4. Arcs strengths are 1 for arcs corresponding to a test outcome which is known to be true for the object o , 0 for an outcome which is known to be false, and proportional to the arc probabilities otherwise.
 5. Arc probabilities are estimated as conditional probabilities of their outcome being true, on the basis of the available examples for which the attribute value is known.
-

becomes available. Further, the statistical distributions of data would be monitored and the tree parameters would be *adapted* as soon as significant changes are observed.

Such an incremental TDIDT method has been proposed and is discussed in [UT 89]. While this may be a key feature in some problems, we don't think that it would be very useful in the context of power system security assessment applications, since trees may be easily reconstructed from scratch as soon as a new data base becomes available.

3.3.3 Missing attribute values

Another, often quoted practical problem occurs when attribute values are unknown for some learning states. For example, in many medical problems, attribute values determined by lengthy or potentially harmful analyses would typically be obtained only if the practitioner has good reasons to believe it will indeed provide interesting information. In these problems a high percentage of attribute values are generally unknown.

A number of methods have been proposed in order to adapt the TDIDT procedure for the treatment of unknown values. Actually there are two different situations where this problem arises. The first is when during the tree induction process some of the examples are incompletely specified. The other is when we actually use the tree to predict output information.

In both situations, a good strategy turns out to be the weighted propagation algorithm illustrated in Table 3.6. At a tree node, if the test attribute value is unknown, we

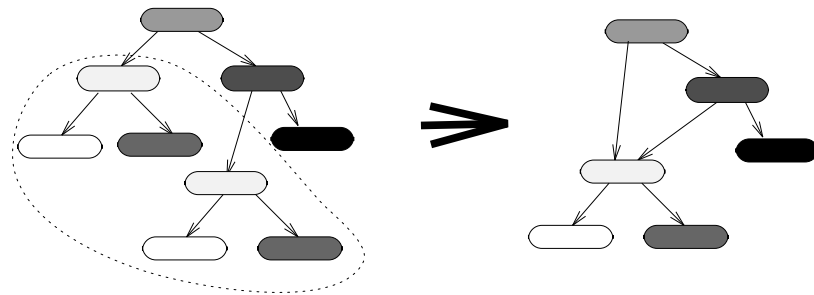


Figure 3.7 Example of trellis structure resulting from node merging

estimate the probability of each outcome and we propagate the object down to every successor, along with its corresponding weight [QU 86b].

At the tree induction step, the various object countings used to evaluate probabilities or impurities are replaced by (non-integer) sums of weights. If the tree is used to predict some information for an unseen object, the latter weight is propagated through the tree and the information is collected and averaged over all relevant terminal nodes. The same technique may also be extended to the case of partial information on the attribute values.

A similar technique was proposed in [CA 87], where within a limited interval around the thresholds used to test continuous attributes, objects are propagated to both successors proportionally to the difference in their attribute value and the test threshold. This actually results in a kind of *fuzzification* of the tree tests, and allows the obtained class probabilities to vary continuously, rather than in a stepwise fashion, when the attribute vector moves from one terminal node to another in the attribute space.

3.3.4 Generalized “tree” structures

A final possible enhancement of the TDIDT approach would be to allow more general structures than simple trees. The two possibilities which have mainly been studied concern the use of “trellis” structures allowing a node to have more than one parent node, and option trees which allow information to be averaged over several possible trees.

Trellises

Extension of the tree structure to trellises has been proposed in [ZI 92] and [CH 88b].

While the tree structure allows us only to decompose a set of objects into subsets of objects, the trellis structure illustrated in Fig. 3.7 allows us also to merge similar subsets

Table 3.7 *SIPINA algorithm. Adapted from [ZI 92]*

-
1. *Starting with a single node trellis, iterate;*
 2. (a) *find the best possible merge of two terminal nodes; if this improves the information provided by the trellis, then merge, and proceed at step 2 (a);*
 (b) *find the best combination of merge and split, defined by any merging of two nodes followed by splitting the resulting node into successors; if this is successful in improving the information provided by the trellis, then merge and split, and proceed at step 2 (a);*
 (c) *find the best split of a terminal node; if this is successful in improving the information provided by the trellis, then split, and proceed at step 2 (a).*
 3. *The obtained structure has reached a local optimum of information and is returned as result.*
-

during the tree construction process. This aims at keeping the sample size sufficiently large, and also at avoiding replications of similar structures in a tree.

In order to be able to balance the reduction in tree complexity resulting from merging some nodes with the incumbent increase in impurity, appropriate non-convex quality measures have to be used. For example, the method described in [ZI 92] uses “ λ -centered” estimates of class-probabilities discussed in appendix A.5. With these estimates, information quantity no longer decreases necessarily when merging nodes, and the SIPINA algorithm proposed in this reference proceeds in the fashion described in Table 3.7.

Option trees

Buntine has proposed an approach which basically consists of inducing a set of class probability trees instead of a single “optimal” tree, and further uses probability averaging over these trees.

In principle, using a Bayesian framework, a posteriori probabilities may be computed from given prior tree probabilities (depending on the number of nodes and values of probabilities attached at terminal nodes) and from the learning set information [BU 92].

Thus, the approach proposed by Buntine consists of identifying a small number of dominant trees : those of nearly maximal posterior probability, i.e. all trees which seem to provide a reasonable explanation of the learning data. Further, the method

computes, for an unseen object o , an average class probability over the dominant trees, using the following type of formula :

$$P(c_i|o, \mathbf{LS}) \approx \frac{\sum_T P(c_i|T, o, \mathbf{LS})P(T|o, \mathbf{LS})}{\sum_T P(T|\mathbf{LS})}, \quad (3.9)$$

where

$P(c_i|T, o, \mathbf{LS})$ is the class probability predicted according to tree T , and
 $P(T|\mathbf{LS}) = P(T|o, \mathbf{LS})$ the posterior probability of this tree, given the \mathbf{LS}
information and the current object, which is supposed to be independent of o .

Option trees are a compact representation of a set of trees which share common parts. Thus, the technique proposed in [BU 92] consists of considering several “dominant” possibilities to develop a tree node, rather than investigating only the most promising one. This includes in particular the trivial subtree, which consists of pruning the current subtree.

Thus, at each internal node of such an option tree several splits and corresponding subtrees are stored together with the corresponding posterior probability updates. During classification, an object is propagated down to each option subtree and the class probabilities inferred by the latter are averaged via a simple recursive scheme.

3.4 THE ULg METHOD

In this section we will describe in detail the tree construction algorithm that we have developed for and applied to various power system problems [WE 86, WE 89b, WE 91a, WE 93b, WE 93h].

To fix ideas, we will use throughout many numerical and graphical illustrations taken from an illustrative real life transient stability problem, introduced below. Further, we will consider only the case where the learning objective is to define decision regions or class probabilities. Note that the method could be generalized to general regression problems, but we believe that other techniques, such as the hybrid DT-ANN approach introduced in chapter 6, would be more appropriate in this context.

3.4.1 Description of a real illustrative problem

We consider the practical problem of preventive transient stability assessment of the 735kV system of Hydro-Québec, depicted in Fig. 3.8. A normal operating condition of this power system is considered as secure from the transient stability viewpoint, if it is able to withstand any permanent single-phase to ground fault, followed by line tripping, fast reclosure and final permanent tripping. It is interesting to recognize that

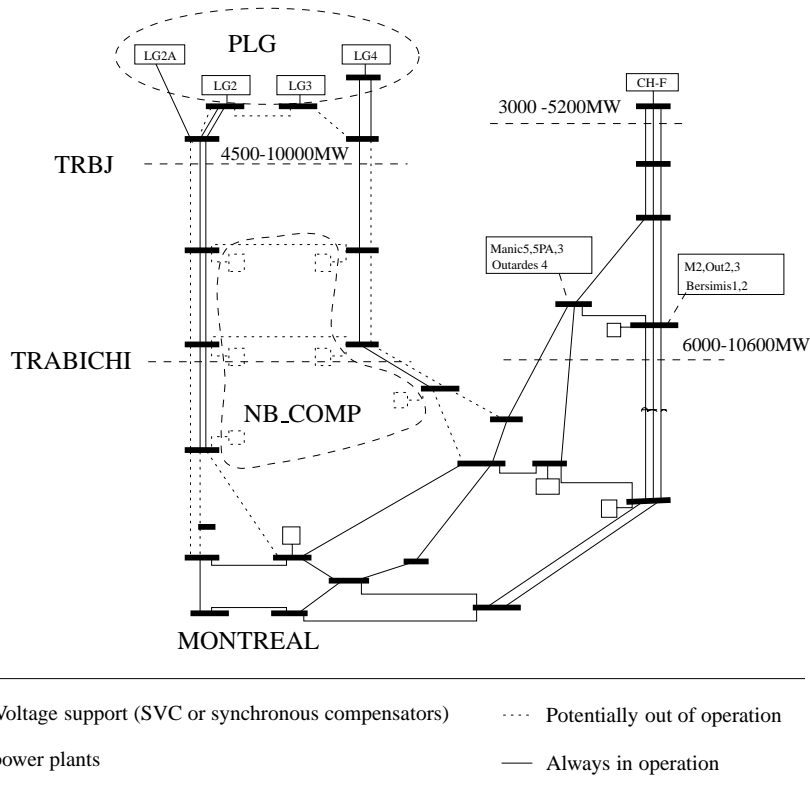


Figure 3.8 One-line diagram of 735kV Hydro-Québec system

this system, due to the very large power flows and long transmission lines, is mainly constrained by its transient stability limits.

For the sake of simplicity, we have looked at a subproblem of the overall system stability problem, considering only faults occurring within the James' Bay transmission corridor in the left part of the one-line diagram. With respect to these faults, the stability is mainly influenced by the power flows and topology within the same corridor.

For this system, a set of transient stability limits have previously been developed, in a manual approach, where operation planning engineers have determined off-line on the basis of carefully chosen simulation scenarios, a set of approximate limit tables relating the system topology and power flows to a Stable/Unstable classification. These limit tables have been implemented on the real-time computer of Hydro-Québec, via an ad hoc data base tool called LIMSEL, which is presently in use for operation [VI 86].

A data base, composed of 12497 normal operating states was generated via random

sampling; it comprises more than 300 different combinations of up to 6 line outages, and about 700 different combinations of reactive voltage support equipment in operation, and a wide variety of power flow distributions. A precise description of the random sampling tool developed for this purpose will be given in §13.4.

For each state, the corresponding classification Stable/Unstable was obtained from LIMSEL running on the backup on-line computer. This yielded 3938 stable states and 8559 unstable states, among which 393 are marginally unstable and 8166 are fairly unstable.

To describe the operating states, and in order to characterize their stability, the following types of candidate attributes were computed for each state.

Power flows. The active power flow through important lines and cutsets in the James' Bay corridor.

Power generations. Total active power generated in the 4 LaGrande (LG) power plants and various combinations.

Voltage support. The number of SVCs or synchronous compensators in operation within the six substations in the James' Bay corridor.

Topological information. Logical variables indicating for each line whether or not it is in operation in the pre-fault situation.

This set, composed of 67 candidate attributes was determined with the help of an expert in charge of transient stability studies at Hydro-Québec. From previous studies it was already known that the total power flow through the corridor would be an important attribute, together with the topological information and the total number of SVCs and synchronous compensators.

The diagram of Fig. 3.9 shows the statistical distribution in the data base of the total power flow in the James' Bay corridor, and the corresponding stability distribution. The height of each vertical bar represents the number of states among the 12497, for which the power flow belongs to the interval corresponding to the basis of the bar. Each bar, is further subdivided into regions of different grey shade, in proportion to the corresponding number of stable, marginal and fairly unstable states. We observe that all states which have a power flow larger than 8700 MW are unstable states, while there exist unstable states in the full range of power flows, down to 4500 MW.

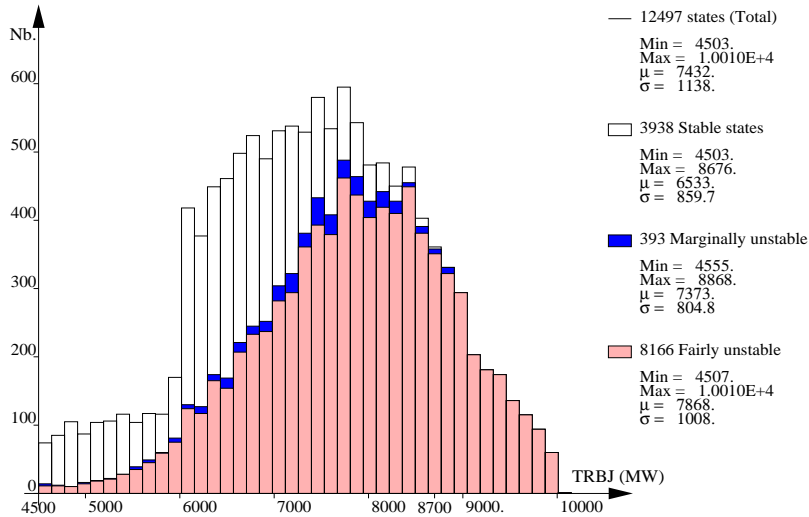


Figure 3.9 Empirical distribution of TRBJ : total James' Bay power flow

Table 3.8 Deriving classification from class probabilities

-
- Let nd_i denote the non-detection cost of class c_i , i.e. the cost assigned to deciding class $c_j \neq c_i$ for an object of class c_i .
 - Let $\hat{p}^i(\mathcal{N}_{t_j})$ denote the conditional class probabilities attached to a terminal node \mathcal{N}_{t_j} .
 - Then, associate the decision $c_i(\mathcal{N}_{t_j})$ such that the product $\hat{p}^i(\mathcal{N}_{t_j}) * nd_i$ is maximal.
-

3.4.2 Quality evaluation

The optimal splitting rule, as well as the stop splitting rule and the pruning criteria used in our method are derived from the entropy concept from information theory, defined in §2.5.4.

Decision trees were obtained indirectly via the construction of class probability trees and the specification of a non-detection cost vector, via the rule given in Table 3.8.

Intuitively, the objective of building a class probability tree is to provide a maximum amount of information about the classes. This is measured by the reduction in classification entropy provided by a tree.

Let us consider a classification problem and a class probability tree T . $H_C(\mathbf{X})$ denotes the initial or prior classification entropy of any subset of \mathcal{U} , defined in eqn. 2.26, and let $\{\mathcal{U}(\mathcal{N}_{t_1}), \dots, \mathcal{U}(\mathcal{N}_{t_q})\}$ denote the partition induced on \mathcal{U} by T , assuming that there are q terminal nodes in T .

Let us define the residual entropy of a tree, in a subset \mathbf{X} of \mathcal{U} , as the expected classification entropy at its leaves

$$H_{C|T}(\mathbf{X}) \triangleq \sum_{i=1, \dots, q} P(\mathcal{N}_{t_i} | \mathbf{X}) * H_C(\mathcal{N}_{t_i} \cap \mathbf{X}). \quad (3.10)$$

Then the mean information quantity provided by the tree in \mathbf{X} is defined as the mutual information of these two partitions in \mathbf{X}

$$I_C^T(\mathbf{X}) \triangleq H_C(\mathbf{X}) - H_{C|T}(\mathbf{X}). \quad (3.11)$$

In particular the overall mean information of a tree is defined by

$$I_C^T(\mathcal{U}) \triangleq H_C(\mathcal{U}) - H_{C|T}(\mathcal{U}), \quad (3.12)$$

and simply denoted by I_C^T .

Ideally, the information provided by a tree would be total, i.e. equal to the prior entropy. In practice this is not necessarily possible. In particular for many problems, characterized by residual uncertainty, the upper bound of information is significantly lower than H_C .

Given a learning set, we will estimate the *apparent* information of a tree, by replacing probabilities by relative frequencies estimated in the learning set

$$I_C^T(\mathbf{LS}) \triangleq H_C(\mathbf{LS}) - H_{C|T}(\mathbf{LS}), \quad (3.13)$$

and the *total* apparent information quantity is obtained by multiplying the latter by the size N of the learning set.

The apparent information of a tree tends to overestimate systematically its actual information. In particular, in many circumstances it is possible to build trees with total apparent information, even if there is some residual uncertainty. Intuitively, large complex trees tend to overfit the data more strongly and their apparent information thus tends to be more optimistic than for smaller trees. Thus, in a quality measure it would be appropriate to compensate for this effect by penalizing in some fashion proportional to the tree complexity.

On the other hand, for a given tree complexity, it seems reasonable to assume that the bias of apparent information will decrease with the size of the learning set, or equivalently the quality should increase in proportion to the total amount of apparent

information and it should decrease in proportion to the tree complexity. This suggests the following form for an empirical quality measure

$$Q(T, \mathbf{LS}) \triangleq N * I_C^T(\mathbf{LS}) - \beta * C(T), \quad (3.14)$$

where $C(T)$ denotes the tree complexity, which is by definition equal to one less than the number of terminal nodes of the tree³.

Thus, the quality of a tree is a compromise between its complexity and its total apparent information quantity. The quality of the initial, trivial tree composed of a single root node is equal to zero, whatever the learning set, since both its complexity and its apparent information are equal to zero.

This quality measure, which we have justified heuristically, may be derived from a theoretical *maximum a posteriori probability* (MAP) computation or equivalently from a *minimum encoding length* (MEL) computation, assuming either that a priori tree probabilities will decrease exponentially with its complexity or (equivalently) that its encoding will require a number of bits increasing linearly with complexity. This and other more theoretical considerations are discussed in detail in the references [WE 90a, WE 93h, WE 94b]. An interesting property of the quality measure is its additivity, which is a consequence of the additivity of the total information quantity and of the complexity measures. For any decomposition of a tree into subtrees, the quality of the total tree is equal to the sum of the qualities of its subtrees.

Exploiting the quality measure, for a given choice of β , the various subtasks considered in the tree induction process may be reformulated in the following way.

Growing. At each step, develop a node in such a way as to maximize the improvement of quality.

Stop splitting. Stop splitting as soon as a (local) maximum of quality is reached.

Pruning. Extract the pruned subtree of maximal quality.

In the following sections we will further discuss the variants of this approach which have been implemented.

3.4.3 Optimal splitting

The optimal splitting rule consists of a search for a locally optimal test maximizing a given score function. This implies finding for each candidate attribute its own optimal split and identifying the attribute which is overall optimal. This calls for the definition

³For binary trees, the total number of nodes is related to the complexity by the formula $\#\mathcal{N} = 2 * C(T) + 1$.

of a score measure and the design of appropriate search algorithms allowing us to handle each type of candidate attributes.

We will not speak about the optimal search of binary partitions of qualitative attributes, since for power system problems the discrete attributes are generally binary topological indicators, which allow only a single partition.

However, before we define the score measure used in our algorithm, we will discuss in detail the case of numerical, essentially real valued attributes which are most important in the case of security assessment problems, as well as linear combinations of two numerical attributes which may yield an important improvement in reliability, as we will illustrate.

Optimal thresholds for ordered attributes

For a numerical attribute we proceed at each node according to the optimal threshold identification procedure described in Table 3.9 to generate the corresponding optimal partition.

This search requires, in addition to the sorting of the learning subset, about N computations of the score function. Although it may seem bulky at first sight, it may be done rather efficiently with available computing hardware. For instance, to sort the 12497 states of our example data base of §3.4.1 with respect to the values of the TRBJ attribute, it would take about 2 seconds⁴, and the overall time required to identify the optimal score within this very large subset would take about 6 additional seconds. At a tree node corresponding to “only” 1000 learning states, these times would shrink to respectively a fraction of a second and 1 second.

It is important to realize that this search procedure is applied repeatedly, for each numerical attribute and at each tree node. It will identify systematically the optimal threshold, whatever the definition of the score measure. Typically, on a 28 MIPS computer the method will not spend more, on average, than a minute at each internal node of the growing tree, even for very large learning sets and a high number of candidate attributes.

Linear combinations of attributes

It is frequently found, in the context of power system security problems that there are two important complementary attributes which share most of the information provided by a tree. In such situations, one could manually define a composite attribute as a function, or try to identify a linear combination attribute on the basis of the learning

⁴Within this work, illustrative CPU times are determined on a 28 MIPS SUN Sparc2 work station. Our research grade TDIDT software was implemented in Lucid CommonLisp.

Table 3.9 *Optimal threshold identification*

-
1. For an attribute a and threshold v , let us denote as the left subset at a node as the set of its learning states such that $a < v$ holds, and the right subset its complement.
 2. Sort the learning subset at the current node, by increasing order of the candidate attribute considered.
 3. Start with an empty left subset and a right subset equal to the complete learning subset of the node.
 4. Sweep through the sorted list of states, removing at each step a state from the right subset and adding it to the left subset.
 5. At each step, update the number of states of each class in the left and right subsets.
 6. Let us denote by v_i the attribute value of the last object moved in the left subset; thus the left subset states are such that $a \leq v_i$.
 7. Similarly, let us denote by v_{i+1} the attribute value of the next object to be moved, but still in the right subset; thus the right subset states are such that $a \geq v_{i+1}$ and $v_i \leq v_{i+1}$.
 8. Only if $v_i < v_{i+1}$, we define a new candidate threshold by $v_{th} = \frac{v_i + v_{i+1}}{2}$, and compute the score of the candidate test $a(o) < v_{th}$ on the basis of the class counts in the left and right subsets.
 9. If the score of the newly evaluated test is better than the previous optimum, we update v_{th} along with its score, as the current best test.
-

set. This amounts to identifying at a tree node a test of the form

$$a_1(o) + \lambda * a_2(o) < v_{th}. \quad (3.15)$$

In our software this is done by a simple nested optimization procedure, which is indicated in Table 3.10. The computational cost of this procedure is equivalent to the treatment of about 10 to 20 real valued attributes.

An interesting generalization of the procedure would be to allow handling a higher number of attributes combined in a linear combination involving the identification of several parameters. With the above algorithm, this would, however, imply a very rapid increase in computational complexity; a more efficient numerical optimization technique should be used. In the following two chapters we will illustrate and compare various methods able to determine hyperplanes in the context of supervised learning, and in chapter 6 we will mention a hybrid technique which could allow us to combine

Table 3.10 *Linear combination search*

-
1. Compute the optimal threshold v_{th1} corresponding to λ_1 and v_{th2} to λ_2 ; λ_1 and λ_2 are specified by the user as the lower and upper bound for the search of λ ; by default $[-\infty \dots \infty]$ is used.
 2. For each candidate value of λ , the corresponding threshold $v_{th}(\lambda)$ is determined by applying the optimal threshold search described previously to the values of the function $a_1(o) + \lambda * a_2(o)$; the corresponding optimal score is thus determined as a function of λ .
 3. The “optimal” value of λ is searched by using a dichotomous search in the interval $[\lambda_1 \dots \lambda_2]$, with a number of iterations generally fixed a priori to less than 20.
-

these methods with the TDIDT approach, so as to determine linear combination trees.

Remark. The above two simplistic search procedures may seem to be rather naive and inefficient. However, they are easy to implement and are not tied to any particular score evaluation function properties, such as continuity and differentiability. They may therefore exploit any kind of appropriate score measure.

Evaluation of candidate splits

In addition to the above described search algorithms, we need to specify the evaluation function or *score* used to select the best split. In the tree induction literature, an apparently very diverse set of measures have been proposed to select an appropriate candidate split. In appendices A.1 to A.5 we discuss carefully these different measures and the purity (or uncertainty) measures from which they are generally derived. As we see, many of these apparently very different possibilities turn out to be not so different and perform rather equivalently in practice.

A convenient way to measure the impurity is to use the *entropy* function well known from thermodynamics and information theory. Among other nice properties let us mention the fact that the entropy function is the only uncertainty measure which is additive [DA 70] : the entropy of a system composed of independent subsystems is equal to the sum of the subsystems’ entropies; similarly, the uncertainty of the outcome of independent events is equal to the sum of the uncertainties of each event taken alone. The other interesting thing about entropy is its probabilistic interpretation, which suggests that reducing entropy amounts to increasing posterior probabilities [WE 90a, WE 92b, WE 93h].

Thus, a simple and in practice appropriate solution consists of using the total amount of apparent information provided by a candidate partition at a node, as the criterion for selecting the most appropriate partition. This is evaluated for each test t , according to the formulas given in §2.5.4, by

$$I_C^t(\mathbf{LS}(\mathcal{N})) = H_C(\mathbf{LS}(\mathcal{N})) - H_{C|t}(\mathbf{LS}(\mathcal{N})), \quad (3.16)$$

Here $H_C(\mathbf{LS}(\mathcal{N}))$ denotes the prior classification entropy estimated in the learning subset at the node, which is obtained by

$$H_C(\mathbf{LS}(\mathcal{N})) = - \sum_{i=1,m} \frac{n_i}{n_{..}} \log \frac{n_i}{n_{..}}, \quad (3.17)$$

where n_i denotes the number of learning states of class c_i at the current node and $n_{..}$ its total number of learning states.

On the other hand, $H_{C|t}(\mathbf{LS}(\mathcal{N}))$ denotes the posterior classification entropy estimated in the learning subset at the node, given the information provided by the test, which is evaluated by

$$H_{C|t}(\mathbf{LS}(\mathcal{N})) = - \sum_{j=1,p} \frac{n_{.j}}{n_{..}} \sum_{i=1,m} \frac{n_{ij}}{n_{.j}} \log \frac{n_{ij}}{n_{.j}}, \quad (3.18)$$

where n_{ij} corresponds to the learning states of class c_i which correspond to the outcome t_j , and $n_{.j}$ correspond to all the states corresponding to outcome t_j .

In practice, rather than using the information quantity directly, we prefer to normalize, in order to obtain values belonging to the unit interval $[0 \dots 1]$, independently of the prior entropy $H_C(\mathbf{LS}(\mathcal{N}))$. The normalized values may be interpreted as an “absolute” measure of the *correlation* between the test outcome and the classification, a value of 1 corresponding to total correlation and a value of 0 to statistical independence. In particular, information quantities obtained at different nodes of a tree, or with various classifications, may still be compared thanks to the normalization property.

In appendix A.3 we compare several possibilities mentioned in §2.5.4 to normalize the information quantity. It turns out that the resulting tree performance, in terms of complexity and reliability, is not very sensitive to the particular choice of score measure. Even in a much larger class of purity measures, not necessarily derived from the logarithmic entropy concept, the resulting tree performances remain very stable. In our method we have chosen to use the normalization I_C^t by the mean value of H_C and H_t indicated in §2.5.4, which was suggested by Kvålseth [KV 87].

Thus our score measure is defined by

$$SCORE(t, \mathbf{LS}(\mathcal{N})) \triangleq C_C^t(\mathbf{LS}(\mathcal{N})) \triangleq \frac{2 * I_C^t(\mathbf{LS}(\mathcal{N}))}{H_C(\mathbf{LS}(\mathcal{N})) + H_t(\mathbf{LS}(\mathcal{N}))}, \quad (3.19)$$

where H_t is the uncertainty or entropy related to the outcome of the test, and is estimated by

$$H_t(\mathbf{LS}(\mathcal{N})) \triangleq - \sum_{j=1,p} \frac{n_{.j}}{n_{..}} \log \frac{n_{.j}}{n_{..}}. \quad (3.20)$$

Table 3.11 *Splitting of the data base by a test*

TRBJ	Stable	Unstable	Total
< 7308.5	3234	2408	5642
> 7308.5	704	6151	6855
Total	3938	8559	12497

Illustration. Let us consider our example problem, and let us compute the score obtained by the test $TRBJ < 7308.5MW$, used to partition the complete data base composed of the 12497 states. This test splits the data base into two subsets composed respectively of 3234 stable and 2408 unstable states for which the condition is true, and 704 stable and 6151 unstable states. This is graphically represented in Table 3.11.

Using logarithms in base two, the prior classification entropy of the complete learning set is computed by

$$\begin{aligned} H_C(\mathbf{LS}(\mathcal{R})) &= -\left[\frac{3938}{12497} \log_2 \frac{3938}{12497} + \frac{8559}{12497} \log_2 \frac{8559}{12497}\right] \\ &= 0.899bit, \end{aligned}$$

and the posterior entropy is computed by

$$\begin{aligned} H_{C|t}(\mathbf{LS}(\mathcal{N})) &= -\left[\frac{5642}{12497} \left\{ \frac{3234}{5642} \log_2 \frac{3234}{5642} + \frac{2408}{5642} \log_2 \frac{2408}{5642} \right\} \right. \\ &\quad \left. + \frac{6855}{12497} \left\{ \frac{704}{6855} \log_2 \frac{704}{6855} + \frac{6151}{6855} \log_2 \frac{6151}{6855} \right\} \right] \\ &= 0.706bit. \end{aligned}$$

Thus, the apparent information provided by this split is obtained by $I_C^t = 0.899 - 0.706 = 0.193bit$.

Finally, the entropy related to the test outcome is obtained by

$$\begin{aligned} H_t(\mathbf{LS}(\mathcal{R})) &= -\left[\frac{5642}{12497} \log_2 \frac{5642}{12497} + \frac{6855}{12497} \log_2 \frac{6855}{12497}\right] \\ &= 0.993bit, \end{aligned}$$

and thus the score associated to the above test is obtained by

$$SCORE(t, \mathbf{LS}(\mathcal{N})) = \frac{2 * 0.193}{0.993 + 0.899} = 0.204.$$

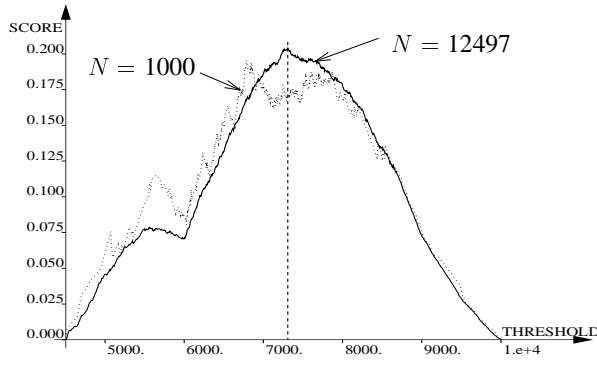


Figure 3.10 Variation of the score of the test $TRBJ < THRESHOLD$

Curves representing the score as a function of the test threshold are indicated in Fig. 3.10. The dotted line curve is obtained for a random sample composed of 1000 learning states drawn in the data base. The plain curve corresponds to the scores obtained when using the complete data base, as in the above derivation. We observe that the shape of the latter curve is much smoother than of the former. We can also check that the value of 7308.5 MW, used in our example computation, actually corresponds to the maximum score, and thus represents the optimal threshold. On the other hand, for the dotted curve a maximum score of 0.196 is obtained for a threshold of 6767.5 MW.

The comparison of the two curves of Fig. 3.10 provides an idea of the dependence of the optimal threshold as well as the corresponding optimal score value on the random nature of a learning sample.

Therefore, it is interesting to provide information about the sampling distribution of the score measure $C_C^t(\mathbf{LS}(\mathcal{N}))$. This is shown to be asymptotically Gaussian and its standard deviation is estimated by [KV 87]

$$\sigma_{C_C^t} = \sqrt{\left(\frac{C_C^t}{n_{..}I_C^t}\right)^2 \sum_{i=1,m} \sum_{j=1,p} n_{ij} \left[\log n_{ij} + \left(\frac{C_C^t}{2} - 1\right) \log(n_{i.}n_{.j}) + (1 - C_C^t) \log n_{..} \right]^2}. \quad (3.21)$$

For example, applying this formula to the test of Table 3.11 yields a standard deviation of $\sigma_{Score} = 0.006$, when $N = 12497$. In the case of the optimal test obtained in the smaller random sample of $N = 1000$ of Fig. 3.10 we get a larger value $\sigma_{Score} = 0.024$.

To further illustrate this random behavior, we have generated 500 random samples composed of 1000 states drawn from the above 12497. On each sample, we have computed the optimal threshold, its score and its standard deviation, according to the

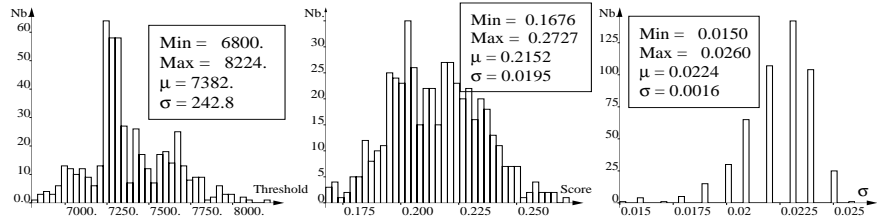


Figure 3.11 *Random variations of optimal thresholds and scores*

above theoretical formulas. The results are summarized in Fig. 3.11, which shows the empirical distributions obtained for these three parameters.

The leftmost diagram shows the distribution of the optimal threshold, which is distributed with a standard deviation of 243 MW, around the mean value of 7382 MW, close to the value of 7308.5 MW obtained in the complete data base. The central curve shows the corresponding distribution of scores and the rightmost curve the distribution of its standard deviation, computed by the above formula. We can observe that the mean value, $\mu = 0.0224$ of the latter diagram, is in good agreement with the sampling standard deviation, $\sigma = 0.0195$ observed on the central diagram.

Comparison of candidate splits and selection

At a given test node, in order to select a test so as to develop this node, the algorithm proceeds in the following fashion.

First, for each candidate attribute it identifies the best partition for this attribute, using the appropriate search algorithm, according to the type of the attribute : discrete, ordered or linear combination of two ordered attributes.

Second, the attributes along with their optimal partitions are sorted by decreasing order of optimal score. Let $Score^*$ denote the optimal score of the best attribute, and σ^* the corresponding standard deviation computed by eqn. (3.21). The list of candidate attributes is supposed to be sorted by the user, in decreasing order of attribute preference.

Then the finally selected attribute is the first one found in the candidate attribute list, obtaining a score at least equal to $Score^* - \beta' \sigma^*$, where β' is a parameter chosen by the user. For example, using $\beta' = 0$ will always lead to selecting an attribute obtaining the highest score.

Illustration. To fix ideas, let us consider our example problem, and look at the selection of an optimal split within the complete data base and the following list of 28 candidate attributes, in the given order of preference

Power generations. PLG, PLG34, PLG2C, PLG23, PLG3, PLG4.

Global power flows. TRBJ, TRBJO, TRBJE, TRCHI, TRCHA, TRNEM, TRALB, TRABICHI, TRQMT, TRMIC, TRMAN, TRCHU.

Topological information. L7057, L7060, L7079, L7090.

Individual power flows. TR7060, TR7057, TR7079, TR7090.

Voltage support devices. NB_COMP, N_CHA.

Assuming that a value of $\beta' = 1.0$ was chosen, we obtain the information shown in Table 3.12, concerning the attribute scores in the complete data base⁵. Only the three first attributes belong to the interval of scores considered to be equivalent. Accordingly, among these the one with the highest priority in the list of candidate attributes is chosen. This is PLG, the total active power generation of the 4 LaGrande power plants (see Fig. 3.8).

Notice that the score of the two other attributes is very close. Actually, a closer look at these attributes shows that they are very strongly correlated with PLG, thus they provide similar information on the stability. TRBJ is the total power flow in the James' Bay corridor, measured nearby the generation plants and TRABICHI denotes the total power through a cross-section in the middle of the corridor. They are clearly strongly correlated with PLG.

This is confirmed by the values given in the last column of Table 3.12 which indicate the correlation of each attributes' optimal test with the optimal test of the selected attribute PLG. The correlation coefficient used here to evaluate the similarity of two tests t_1 and t_2 is defined, similarly to the the score measure, by the following formula

$$Correl(t_1, t_2) \triangleq \frac{2 * I_{t_1}^{t_2}(\mathbf{LS}(\mathcal{N}))}{H_{t_1}(\mathbf{LS}(\mathcal{N})) + H_{t_2}(\mathbf{LS}(\mathcal{N}))}. \quad (3.22)$$

Let us illustrate the use of a linear combination attribute. Although in the above table the attribute NB_COMP, denoting the total number of compensation devices in operation in the James' Bay corridor, obtains a rather low score, it is known from prior expertise that this attribute influences very strongly the stability limits of the corridor. Thus, it is presumed that a linear combination of this attribute together with the total power flow attribute TRBJ would provide increased discrimination power. Indeed, proposing this linear combination attribute to the algorithm, results in the following optimal linear combination

$$TRBJ - 227 * NB_COMP < 5560MW \quad (3.23)$$

corresponding to a score of 0.3646, which is significantly higher than the above optimal score without linear combination attribute.

⁵Among the candidate attributes, L7090, L7060, TRMIC, N_CHA, TR7090, TR7079, TRMAN, TRQMT, L7057, TRCHU which obtain a score smaller than 10% of the best score are not shown in the table.

Table 3.12 Detailed information about attribute scores and correlations

```

Expanding TOP-NODE : N=12497, UNSTABLE=8559, STABLE=3938,
Total Prior Entropy N*Hc : 11234.7
.....
--> A test node : TOP-NODE
=====
CANDIDATE ATTR. EVALUATION:  SCORE SIGMA  N*INFO cor PLG
=====
*  TRBJ < 7307.5             0.2037 0.006  2408.8   1.00
** PLG < 7376.5             0.2037 0.006  2408.5   0.99
*  TRABICHI < 6698.5        0.2035 0.006  2401.7   0.89
-----
TRBJO < 4193.0              0.1437 0.006  1586.6   0.22
TRNEM < 4257.5              0.1349 0.006  1483.6   0.22
PLG23 < 6029.5              0.1238 0.005  1436.9   0.31
PLG34 < 3265.5              0.0913 0.005  1082.7   0.14
PLG4 < 1592.5               0.0727 0.004   817.6   0.11
PLG2C < 4394.5              0.0673 0.004   787.8   0.18
PLG3 < 1418.5               0.0653 0.004   764.4   0.09
TRCHI < 1338.5              0.0582 0.004   475.9   0.11
TR7060 < 956.5              0.0581 0.004   623.3   0.01
TRCHA < 1331.5              0.0578 0.004   472.0   0.11
TRALB < 1717.5              0.0563 0.004   495.7   0.16
L7079 < 1.0                  0.0388 0.003   346.3   0.00
TRBJE < 2232.5              0.0376 0.003   412.9   0.10
NB_COMP < 4.0                0.0299 0.003   277.6   0.01
TR7057 < 1888.5             0.0235 0.002   163.8   0.05
=====
CHOSEN TEST : PLG < 7376.5 (Outcomes : YES NO)
=====

```

The parameters of the linear combination test, which may be rewritten as $TRBJ < 5560 + 227 * NB_COMP$, translate the beneficial effect of the number of compensation devices on the threshold of the total power flow attribute. The line in the $(TRBJ, NB_COMP)$ plane corresponding to the above threshold is represented in Fig. 3.12, along with a random sample of 500 operating states. One can observe that on the right side of the line there are almost only unstable states, whereas on the left side there is a mixture of stable and unstable states.

3.4.4 Stop splitting and pruning

In our initial investigations with the tree growing algorithms, in the context of transient stability assessment, we have experimented with various stop splitting criteria, using

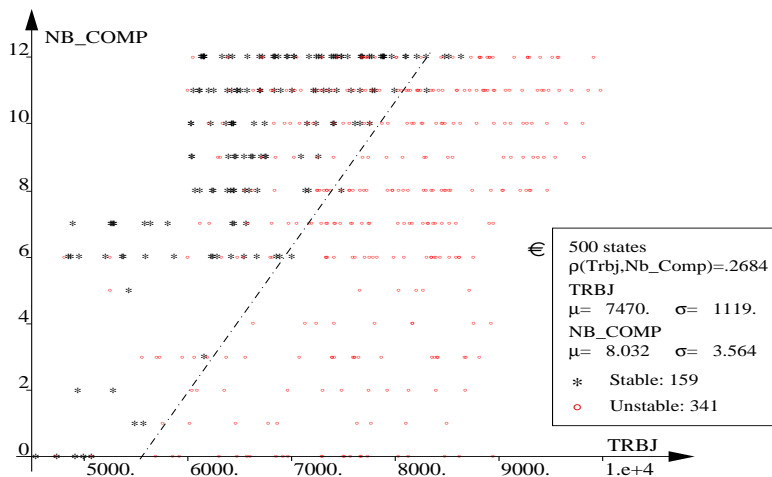


Figure 3.12 Illustration of linear combination attribute

for example lower bounds on the number of learning states and/or on the residual entropy at a terminal node [WE 86, WE 87b].

These experiments led us to the conclusion that in order to obtain good reliability it was necessary to develop the trees almost completely. This strategy unfortunately gave overly complex trees, mostly composed of random splits and which were very difficult to interpret. Thus, a strong need was felt for an approach able to distinguish among random splits and splits significantly correlated with the stability classification. We therefore proposed the hypothesis testing approach, in order to identify the situations where the apparent reduction in entropy due to a split was indeed significant [WE 89b]. The observation that the hypothesis testing approach was equivalent to detecting a local maximum of quality became clear later, and allowed a more elegant formulation of the pruning criterion.

Stop splitting via hypothesis testing

It is important to notice that the hypothesis testing was proposed by many researchers, not the least of which is Quinlan [QU 86a], in order to handle the case of noisy attributes and noisy classifications.

Our problems however were formulated as deterministic problems, without any noise and our difficulties were related to the necessity of providing a simple approximation of a very complex problem, due to the limited amount of information provided by any learning set of reasonable size. Indeed, although we knew that a correct decision tree for most transient stability problems would be infinitely complex, we were trying to find a good compromise allowing us to represent this in a simplified fashion as far as

is confirmed by the learning data.

In order to remain as coherent as possible with the strategy we used to identify the most interesting test, we decided to use the so-called G -statistic proposed by [KV 87]. Indeed, one can show that under the hypothesis of statistical independence of the test issue and goal classification, the sampling distribution of the following quantity

$$G^2 \triangleq 2n_{..} * \ln 2 * I_C^t(\mathbf{L}\mathcal{S}(\mathcal{N})), \quad (3.24)$$

which is directly proportional to the total apparent information provided by a test, follows a χ -square distribution with $(m - 1) * (p - 1)$ degrees of freedom.

Thus, conforming to the general scheme of Table 3.3, the stop-splitting rule amounts to fixing a priori a value of the nondetection risk α of the independence hypothesis and to comparing the value of $2n_{..} * \ln 2 * I_C^t(\mathbf{L}\mathcal{S}(\mathcal{N}))$ obtained for the optimal test, with the threshold value obtained from the χ -square table. A value of $\alpha = 1.0$ would amount to systematically rejecting the independence hypothesis, and to considering even the smallest increase in apparent information as significant. This would lead to fully growing the trees, so as to separate completely their learning states of different classes. On the other extreme, using a too small value of α would lead to develop only nodes with a very large increase in apparent information, and would produce overly simple trees.

A very large number of simulations, for a very diverse range of problems, mainly from power system transient stability and voltage security, have shown that optimal values of α are in the range of $10^{-3} \dots 10^{-4}$, which in terms of total apparent information $N * I_C^t$ leads to a threshold value in the interval of $7 \dots 15$. These simulations have also shown that the resulting trees are generally close to optimal in terms of reliability, sometimes slightly suboptimal, but always significantly less complex than fully grown trees. To fix ideas, the ratio of the number of nodes of the full tree to the number of nodes of a pruned one with $\alpha = 10^{-4}$ lies generally between 2 and 10 [WE 90a].

Thus, we conclude that the hypothesis testing approach successfully prevents trees from overfitting their learning set, and leads to much simpler and less random trees. In terms of practical outcomes, these in general are more reliable and much easier to interpret than the trees obtained without using the hypothesis test.

Stop splitting via quality criterion

As we mentioned above, another approach to define a stop splitting criterion is based on the quality measure.

Since the objective of the hill-climbing tree growing algorithm is to maximize the tree quality, a good criterion of stop splitting would be to detect a local maximum of quality. For a given value of β , the quality variation of a tree T resulting from splitting

a terminal node \mathcal{N} with a test t , is computed by

$$\Delta Q(T, \mathbf{LS})_{t\mathcal{N}} \triangleq n_{..} I_C^t(\mathbf{LS}(\mathcal{N})) - \beta * (p - 1), \quad (3.25)$$

where $(p - 1)$ represents the variation of the number of terminal nodes due to the node development. This is always equal to 1 in the case of binary trees. Thus, the detection of a local maximum of quality at a terminal node of a tree amounts to comparing the value of the total apparent increase in information provided by the optimal test t^* , $n_{..} I_C^*(\mathbf{LS}(\mathcal{N}))$, with the value of the threshold β .

Similarly to the risk α of the hypothesis testing approach, β is a user defined parameter and should be tuned according to problem specifics. A value of $\beta = 0$ would consist of not taking into account the tree complexity in the quality measure; this is equivalent to assuming $\alpha = 1.0$ in the hypothesis testing approach and produces fully grown trees. On the other hand, using very large values of β would lead to oversimplified trees. In particular, for $\beta > N * H_C(\mathbf{LS})$ the tree will shrink to its root node, since no test will be able to provide enough information to yield an overall positive variation of Q .

Pruning and pruning sequences

There are two possible difficulties with the above described stop-splitting approaches.

The first is related to the fact that the stop-splitting criterion is only able to detect a *local* maximum of quality. As soon as a node development is not sufficiently promising, one irrevocably stops splitting this node. There are however situations where it would be possible to improve the tree, provided that at least two or more successive node developments are considered. In other words, we have reached a local maximum which is not the global maximum.

The second difficulty, which is probably more often encountered in practice, is due to the fact that the stop-splitting approaches require the user to predefine the pruning parameter, α or β , the optimal value of which may depend on problem specifics, and on the complexity vs reliability compromise which is desired.

Each time a new learning problem is considered, the value of this parameter should be tuned to the problem characteristics. This may be done during some initial tree growing trials for various values of the parameter. Each one of the obtained trees may be evaluated on the basis of an independent test set, and the value of the parameter corresponding to the most appropriate tree would be retained for further tree building.

One of the questions of such a strategy is how often one must adapt the pruning parameter. For example, should it change as soon as the learning set size changes, or when candidate attributes are modified or only when considering a completely new learning problem. Actually, as we have already mentioned, it has been observed in practice that the optimal value of α (and thus of β) is not very sensitive to problem

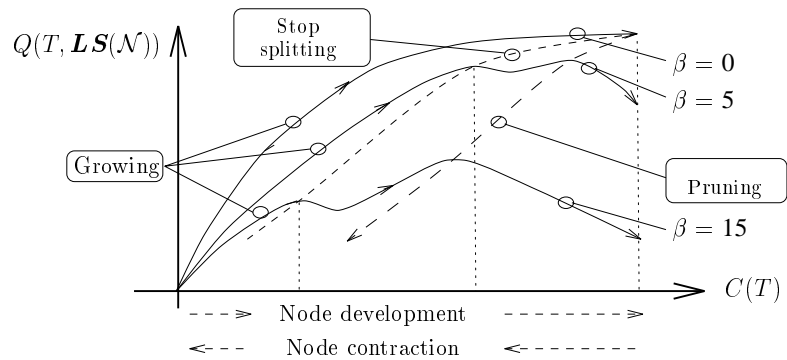


Figure 3.13 *Quality variation : growing and pruning (adapted from [WE 93h])*

specifics, at least within the limited area of power system security problems, which we have studied extensively. Nevertheless, it is interesting to define a more systematic approach to identify the optimal pruning degree of a tree.

Figure 3.13 intuitively suggests the behavior of tree quality curves for variable values of β . Each one of the plain curves shows the variation of tree quality as terminal nodes are progressively developed⁶. The left hand dotted line suggests that the stop splitting approach provides a local maximum along these curves, whereas the right hand dotted line, which represents the optimally pruned trees, by definition corresponds to the global maximum along the curve. While both curves indicate that for increasing value of β the resulting tree complexity decreases, the optimally pruned tree is always of slightly higher quality and complexity than the tree obtained by the stop splitting approach.

Both of the above problems may thus be tackled by replacing the stop splitting complexity control by the tree pruning approach. This amounts to growing a tree completely, i.e. along the curve in Fig. 3.13 corresponding to $\beta = 0$ (or $\alpha = 1.0$), and then simplifying this tree by contracting its test nodes, so as to extract its pruned subtree of maximal quality, for β increasing from 0 to ∞ .

This yields the nested sequence of shrinking trees represented by the right hand dashed line in Fig. 3.13. Using an independent pruning set to estimate the test set error rates for each one of these pruned trees will allow us to appraise their generalization capability to unseen objects. This is illustrated in Fig. 3.14, which shows the variation of the test set error rate and of the complexity of the optimally pruned trees for increasing values of β . On the basis of these latter curves, one may then select an appropriate pruning level β^* , for example by using the “1 standard error rule” as is suggested in Fig. 3.14. This consists of selecting the pruned tree as the most simple tree for which the test set

⁶We suppose that at each step the node whose optimal test leads to the maximal increase in information is chosen to be developed.

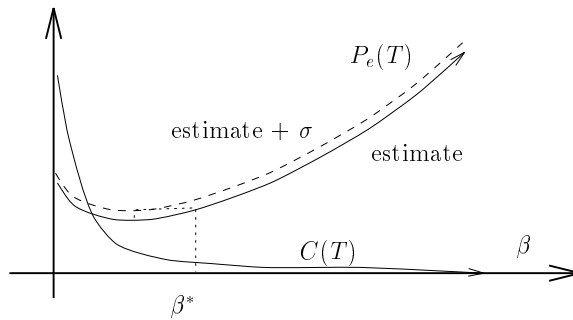


Figure 3.14 Test set error of pruned trees and “1 standard error rule”

error rate is not larger than the minimal test set error rate along the pruning curves plus its standard deviation.

The above algorithm may be implemented quite efficiently, allowing us to generate the complete pruning sequence with a reasonable overhead of computing time with respect to the stop splitting approach.

Illustration. Considering again our example transient stability problem, we have built a completely grown tree on the basis of the first 8000 states of the data base and 87 candidate attributes, including in addition to the 67 attributes proposed by the utility engineer, four linear combination attributes and some other combined attributes.

This yields a very large initial tree T_0 composed of 435 nodes, corresponding to a complexity of $C(T_0) = 217$. This tree was evaluated on the basis of a PS composed of 2000 other states of the data base, yielding an error rate of 4.35%. Starting with this initial tree, its pruning sequence was computed and an optimal tree selected using the “1 standard error rule”⁷. The resulting tree T_{β^*} corresponds to $\beta^* \in [12.235 \dots 12.654[$ and is composed of 115 nodes (i.e. $C(T_{\beta^*}) = 57$) and has an error rate in the above test set of 3.95%.

Figure 3.15 shows the curves of the tree complexity and test set error rate along the sequence of shrinking trees for increasing values of β . For the sake of clarity the graphs are zoomed on the interesting range of β values. The vertical line shows the location of the optimal tree, on the left side of which it is possible to observe the slight overfitting of the more complex trees, which translates into an increased error rate. On the right side of this line one can observe that pruning the tree further would lead to removing some significant tests, resulting in a rapid increase in error rate.

Since the pruned tree was selected on the basis of the pruning set error rate, it is

⁷The standard deviation of the error rate is computed by the formula $\sqrt{\frac{P_e(100-P_e)}{M}}$; in most of our simulations it is approximately equal to 0.5%

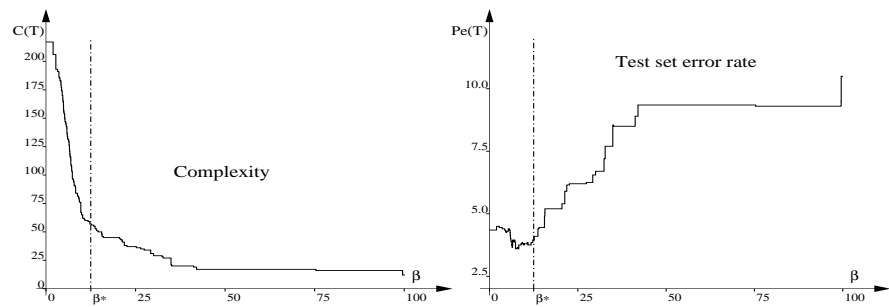


Figure 3.15 Pruning sequences for a transient stability assessment tree

legitimate to suspect the latter to be optimistically biased. Thus, we have re-tested the pruned tree, as well as the initial one, on the basis of an independent test set, composed of the 2497 remaining states of the data base. This yielded respective error rates of 4.21% for the pruned tree and 4.17% for the initial tree, which are not significantly different from the above two error rates. This is in good agreement with our overall experience, suggesting that using the “1 standard error rule” indeed produces in general quite simple trees, which are close to optimal and for which the pruning set error rate is not strongly biased. Thus in practice, it is not necessary to reserve an extra independent test set, for estimating the reliability of the pruned tree.

In terms of computational cost, the pruning approach presents an overhead with respect to the stop splitting approach, mainly due to the increase in CPU time required to grow the initial tree fully. In the present example, the total computing time required to grow this tree was of 3hrs 31min CPU time. Then it took about 187 seconds to generate the complete pruning sequence and to select the optimally pruned tree, and some 20 additional seconds to test the latter tree’s reliability on the basis of the 2497 independent states.

In comparison, using the hypothesis testing stop splitting rule, together with a value of $\alpha = 5 * 10^{-5}$ (corresponding to the above optimal value of β) yields in this case exactly the same tree, but requires only 2hrs 16min CPU time, i.e. a reduction of about 35% with respect to the above figure.

Thus, if the optimal level of pruning is known a priori, it is of course more efficient to use the stop splitting approach. However, in order to determine this optimal value, for example during initial trials, it is much more systematic and efficient to use the pruning approach than repetitive tree building with different settings of α .

Figure 3.16 provides a partial view of the above pruned tree, showing its most important parts nearby the top-node. The notation used for a typical node is also represented at the top left hand side of the tree; one can see that each tree node is represented by a box, the upper part of which corresponds to the proportions of stable and unstable

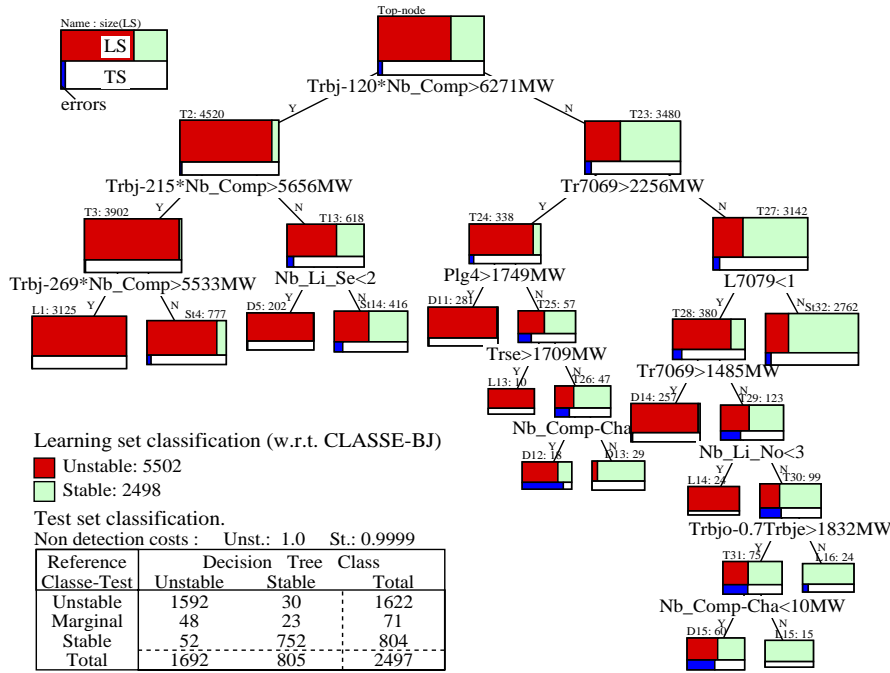


Figure 3.16 Decision tree : $N = 8000$, $M = 2497$, $\alpha = 5 * 10^{-5}$, $P_e = 4.2\%$

learning states relative to this node. In addition to the label indicating the type of a node, the number of learning states of the node is indicated next to it. Test nodes are identified by the label “Ti” or “STi”, the latter corresponding to subtrees which have not been drawn on the picture. Terminal nodes are identified by a label “Li” for leaves and “Di” for deadends. A leaf is a terminal node with a sufficiently class pure learning subset, i.e. a learning subset of mean entropy lower than a predefined threshold (H_m) value taken here equal to 0.01bit, whereas a deadend is a node which corresponds to a pruned subtree.

The test results obtained when classifying the 2497 test states are shown in the table next to the tree. The non-detection costs used to assign a classification to the terminal nodes of the tree are almost identical, and the majority class is used. When there is a tie, the slightly lower non-detection cost of the stable class ensures that the unstable class is systematically chosen. The table indicates the number of stable, marginally unstable and fairly unstable states, as they are classified by the tree. The 23 marginally unstable states classified stable correspond to the so-called “normal” (i.e. unavoidable) non-detections whereas the 30 fairly unstable states classified stable are the “dangerous” non-detections. The false alarms are the 52 stable states which are classified unstable by the tree. Notice that only 30 out of 1622 fairly unstable states are not detected, which yields a rather low non-detection rate of the dangerous situations of 1.85%.

Table 3.13 Percentage of $N * I_C^T$ provided by each test attribute

TRBJ+B*Nb_CO:	51.8	TR7069	:	9.7	L7079	:	6.2	
TRSE	:	5.8	NB_COMP	:	4.6	TR7062	:	4.3
NB_LI_SE	:	3.4	NB_LI_NE	:	1.9	PLG4	:	1.9
L7090	:	1.8	NB_COMP-CHA	:	1.6	TRSO	:	1.2
CLASSE-BASE	:	1.0	TR7094	:	1.0	PLG+B*TRBJ	:	0.8
TRNEM	:	0.6	NB_LI_NO	:	0.5	TR7025	:	0.4
TRNO	:	0.3	TRABI	:	0.3	TRBJO+B*TRBJ	:	0.3
TR7044	:	0.2	PLG3	:	0.2	TR7016	:	0.2

In addition, at each node of the tree the proportion of erroneous classifications of test states are indicated for the corresponding subtree. At the terminal nodes this corresponds to the proportion of its test states of the minority class. At intermediate nodes, it corresponds to the mean error rate of the complete subtree, and at the top-node it corresponds to the overall error rate of the tree (i.e. 4.2%).

Finally, although this is hardly apparent from the above picture, we mention that the decision tree building has identified among the 87 candidate attributes 24 relevant ones. The tree allows us to reduce the initial total entropy of the learning set from $N * H_C = 7166bit$ to a residual entropy value $N * H_{C|T} = 965bit$. This amounts to a total information quantity provided by the tree of 86.53%.

Table 3.13 provides detailed information of the way this information is shared among the different test attributes. The attributes are sorted by decreasing values of their information quantity which is defined as the sum of the total information quantities $N(\mathcal{N}) * I_C^t(\mathcal{N})$ of the test nodes corresponding to a given attribute, expressed as a percentage of the total information of tree $N * I_C^T$. One may observe that more than 50% of the tree information is provided by the linear combination attribute used at the top-node, and another 40% by the following eight attributes which involve the topology (L7079, NB_COMP, NB_LLSE, NB_LLNE) and power flows (TR7069, TRSE, TR7062) in the James' Bay corridor as well as the active power generated in one of the LaGrande power plants (PLG4). This gives a first impression on the way various pieces of interesting information may be provided by a tree.

Remark. The information obtained via the tree portrayed and described above was obtained with the basic TDIDT approach. A more in depth investigation and adaptation allows us to improve the latter information from various viewpoints, as we will show in §13.4. In particular, it is possible to decompose the overall stability problem into subproblems yielding simpler and more accurate trees, which are also easier to analyze from the physical point of view. Further, the tree building process may be biased so as to reduce further the number of non-detections of unstable situations.

3.5 OTHER CLASSES OF MACHINE LEARNING METHODS

While most of the real world applications of machine learning use one of the TDIDT methods, there exists also a large variety of other machine learning approaches.

Among them we may distinguish between methods which stick to the same simple attribute based representation of objects and rules as the TDIDT methods, and those which aim at exploiting more complex higher level, relational representation languages.

For the latter kind of methods, the objective is to tackle situations where interesting information is provided by the structure of objects and the relations among their components. Many real-life problems may involve the representation of such complex information. Among the methods trying to operate with the high level representation languages required for these problems, let us quote the recent work of Quinlan, using first order predicate calculus to represent objects and rules [QU 90, QU 91].

This is certainly a promising long term research area, but for the time being the resulting methods are still at the experimental stage, able to handle only rather small problems, and lacking many features required for real world applications, such as for example the ability to cope with numeric as well as incomplete or contradictory information. On the other hand, in the context of our power system security problems it is not sure whether they have a true potential of outranking the presently available methods, since the simple attribute based representation presently used fits nicely into these problems.

Among the former category of machine learning methods (i.e. those which use the attribute based representation), we will briefly describe two complementary approaches concerning respectively *instance based learning* and *rule induction*. As we will see below, the rule induction techniques might be able to improve the interpretability of the information provided by decision trees. On the other hand, instance based learning techniques allow one to identify in a large data base the reference cases relevant for drawing conclusions about the current situation, providing thereby potentially useful guidelines for an operator.

In addition to these two learning methods, we will give a brief overview of the genetic algorithms, which have recently received increased interest. These could be applied as an auxiliary tool, for solving some of the difficult combinatorial search problems arising within any of the above machine learning methods.

3.5.1 Rule induction

Practical motivation

A decision tree decomposes its attribute space into a set of exhaustive and mutually exclusive regions, corresponding to its terminal nodes. It may be translated into a corresponding set of decision or production rules. Each rule corresponds to a terminal node and associates a *conjunction* of attribute tests, encountered on the path from the top-node of the tree to the terminal node, with the majority class (or class probabilities) attached at the terminal node. Thereby a class is represented as a *disjunction* of the mutually exclusive rules corresponding to the terminal nodes of this class.

It is a straightforward task to translate a decision tree into a corresponding set of production rules. In general it is also possible to further simplify the resulting set of rules without loss of accuracy [QU 87a], by relaxing the condition of mutual exclusiveness and exhaustiveness. This may greatly improve the human intelligibility of the information carried by a decision tree while maintaining an optimal level of accuracy.

Another approach to machine learning consists of building the rules directly on the basis of the learning set without requiring the intermediate building of a decision tree. Probably the most well known such rule induction methods correspond to the AQ family of algorithms [MI 83]. This method was initially developed for the deterministic case and was later adapted to allow the consideration of uncertain, unreliable or incomplete information. From this evolution some new - essentially probabilistic - rule induction methods have emerged which are now able to compete with the decision tree induction techniques in terms of their simplicity vs. accuracy compromise. At the same time, these methods are of sufficient computational efficiency to handle real-world problems [WE 90c, CL 89].

We briefly describe below the CN2 algorithm [CL 89], which is quite representative of the rule induction methods, and which has obtained promising results in the context of the Statlog project with two different power system security data sets [TA 94].

In practice we hope to be able to further improve the interpretability with respect to decision trees. On the one hand, with the rule induction methods it could be easier to generate simpler and more selective rules to detect insecurity. On the other hand, it may be possible to restrict their scope to those regions of the attribute space which are sufficiently well represented in the learning set, in particular so as to avoid an optimistic classification in the regions which have not been sampled. Further, the rule induction methods use a more general search strategy allowing them to trade computational efficiency and rule quality in a more flexible way than the TDIDT procedures.

Table 3.14 *The CN2 induction algorithm. Adapted from [CL89]***CN2(LS)**

1. Start with an empty list of rules and let initially $E = LS$.
2. Let $RULE^*(E)$ denote the best rule found for E , i.e. the most informative and significant conjunctive rule.
3. If $RULE^* = \emptyset$ then stop and return the current rule list as solution.
4. Otherwise, add $RULE^*$ to the current list of rules, and remove from E the objects covered by $RULE^*$, and continue at step 2.

 $RULE^*(E)$

1. Let initially $STAR$ be the set containing only the empty rule, S be the set of all possible selectors, and $RULE^*$ be the empty rule.
2. If $STAR$ or E are empty then return $RULE^*$.
3. Let $NEWSTAR$ denote the set obtained by specializing all rules in $STAR$, in all possible ways by adding a single selector of S , and remove all the rules of $NEWSTAR$ which either are in $STAR$ (i.e. they are not a proper specialization of the previously considered rules), or are null (they are an overspecialization).
4. For every rule C_i in $NEWSTAR$, if C_i is statistically significant and better than $RULE^*$ when tested on E , then replace $RULE^*$ by C_i .
5. Set $STAR$ to the K best rules of $NEWSTAR$ for the next iteration.

The CN2 algorithm [CL 89]

CN2 is a direct descendant of the AQ family of rule induction algorithms, which constructs a set of conjunctive rules in a sequential fashion. The dependence of AQ on particular instances has been removed in CN2 and the conditions of perfect consistency and coherency with the learning set have also been relaxed. This enables CN2 to properly cope with noise and uncertainties.

The algorithm is described in Table 3.14. It is composed of an outer loop $CN2(LS)$, which grows a list of rules and an inner loop $RULE^*(E)$ which improves gradually the rules by specializing them, i.e. by adding some conditions on attribute value so as to restrict the set of objects covered by a rule in order to improve its information. An attribute condition is built from the set of possible values of an attribute by using the following relations $\{=, \leq, >, \neq\}$. A rule is a conjunction of such tests, which are selected sequentially by the beam search procedure $RULE^*(E)$. The set of rules covering the learning set is constructed sequentially, whereby each rule is evaluated

only on the learning states not yet covered by the preceding rules. Thus the resulting set of rules should be applied in the same order as they have been generated.

The notable fact about the CN2 algorithm is that it is a result of combining interesting features of both AQ and ID3. In particular, it uses an information theoretic criterion, similar to ID3, to assess the quality of a rule and a χ -square like hypothesis test, similar to our stop-splitting rule, to test the significance of rules so as to avoid overfitting. On the other hand, the main advantage of the method drawn from AQ is its improved beam search strategy, which allows us to search along a set of most promising search directions, rather than a single one, as in the hill-climbing approach. The computational complexity of the algorithm is directly proportional to the number K of search directions investigated in parallel, which allows to trade computing times and expected rule quality.

In the Statlog project this method, compared to the tree induction algorithms, has obtained quite similar accuracy results although it was significantly slower.

3.5.2 Instance based learning (IBL)

Practical motivation

In the decision tree or rule learning approaches the aim is to derive via an appropriate inductive inference technique a general rule from a set of specific learning examples. This is a model driven approach, where learning consists of searching in a space of possible rules in order to replace the information of a learning set by a set of general rules, which are then used later for predicting classes of new objects. The objective is mainly to find an explicit model which is based only implicitly on the relevant similarities and differences among objects.

In contrast, the instance or object driven approach to learning consists of storing the individual learning objects and modelling explicitly their relevant similarity relationships, so as to allow generalization to unseen objects. In this framework, learning will essentially consist of deriving distance functions appropriate for generalization [ST86, CO91, AH91, SA91b]. Objects and distances are then used to find the best reference case matching a new object, and the information stored together with the reference case is extrapolated, taking into account the differences between the two objects and possibly exploiting prior knowledge about the problem.

This is clearly also one of the mechanisms by which human experts - in particular in the context of security analysis - use their experience to solve difficult problems. One of the interesting possibilities of instance based learning is that if an operator wants to check the security information derived for the current state, the system may simply present the relevant instances of the data base and their main differences with the current situation. The operator may then focus his analysis on these differences so

as to determine the confidence he can have in the extrapolated security information. Further, at this step it would be relatively easy to exploit domain specific knowledge in order to bound the possible influence of attribute differences on the security margin.

The model-driven and object-driven approaches to learning are certainly in general complementary and could be used together so as to make a better use of the information available in a data base. This is discussed further in the context of hybrid approaches in chapter 6. Below we will merely provide a hint on the so-called PEBLS *instance based learning* paradigm [CO 91].

PEBLS [CO 91]

PEBLS (standing maybe for *Practical Exemplar Based Learning System*) extends the nearest neighbor methods of statistical pattern recognition discussed in §4.3.1 to symbolic attributes. The main problem when applying the nearest neighbor idea is to define an appropriate similarity measure which is used to compare different objects. This involves the definition of attribute differences and their weighted combination in a distance measure. In the context of statistical pattern recognition, techniques have mainly been developed to handle real valued attributes. Thus the basic purpose of PEBLS is to extend these techniques to the case of symbolic information.

This results in the definition, on the basis of a learning set, of *value difference* tables [ST 86], producing a non-Euclidean metric and the idea of *exception spaces* which attach weights to individual objects in a data base, allowing one to control the size of the region around an object where its information may be reliably used for extrapolation.

The value difference metric [ST 86]

The idea of the value difference metric is to take into account the overall similarity of the classification information of the different values of an attribute, in order to define the relative importance of differences of an attribute's values. To set up the value difference tables, the attributes are analyzed one by one, which implicitly consists of neglecting, when defining the differences among attributes values, the cross-correlations among *several* attributes and classes.

The distance between two values v_1, v_2 of a *qualitative* attribute a is defined in [ST 86] by

$$\delta_a(v_1, v_2) \triangleq w_a * \sum_{i=1, \dots, m} \left| \frac{n_{i1}}{n_{.1}} - \frac{n_{i2}}{n_{.2}} \right|^r, \quad (3.26)$$

where

n_{ij} denotes the number of learning states o such that $c(o) = c_i$ and $a(o) = v_j$

$n_{.j}$ denotes the total number of learning states such that $a(o) = v_j$,

r is a constant, usually set to 1, and

w_a is a weight controlling the importance of an attribute in the overall distance.

Notice that in [CO91] the weight w_a is always set to 1, which consists of weighting the attribute differences proportionally to the r -norm of the corresponding difference in conditional class probability vectors. Thus, if two attributes are highly correlated the corresponding information will be taken into account twice in the overall distance.

For an *ordered* attribute, we could define a similar distance

$$\delta_a(v_1, v_2) \triangleq w_a * \sum_{i=1, \dots, m} \left| \frac{n_{i,v_1}}{n_{.,v_1}} - \frac{n_{i,v_2}}{n_{.,v_2}} \right|^r, \quad (3.27)$$

where

n_{i,v_j} denotes the number of learning states such that $c(o) = c_i$ and $a(o) \leq v_j$, and $n_{.,v_j}$ denotes the total number of learning states such that $a(o) \leq v_j$.

Notice that eqns. (3.26) and (3.27) measure the difference among attribute values by the distance of conditional class probability distributions; thus other such measures comparing probability distributions, e.g. based on the entropy concept, could as well be used to derive alternative attribute value distances.

Another possibility would consist of defining the value distance by

$$\delta_a(v_1, v_2) \triangleq w_a * \sum_{i=1, \dots, m} \left| \frac{n_{i1}}{n_{i.}} - \frac{n_{i2}}{n_{i.}} \right|^r, \quad (3.28)$$

where $n_{i.}$ denotes the total number of states of class c_i .

Then the corresponding distance for an ordered attribute would be defined by

$$\delta_a(v_1, v_2) \triangleq w_a * \sum_{i=1, \dots, m} \left| \frac{n_{i,v_1}}{n_{i.}} - \frac{n_{i,v_2}}{n_{i.}} \right|^r, \quad (3.29)$$

which takes into account the requirement that

$$v_1 \leq v_2 \leq v_3 \implies \delta(v_1, v_2) \leq \delta(v_1, v_3).$$

Finally the total distance between two objects is defined by

$$\Delta(o_1, o_2) \triangleq w(o_1) * w(o_2) \sqrt[k]{\sum_{i=1, \dots, n} \delta_{a_i}(a_i(o_1), a_i(o_2))^k}, \quad (3.30)$$

where

k denotes the order of the distance, and

$w(o)$ is a weight controlling the importance of an object in the data base.

While the value difference tables are directly computed from the learning set classification applying either eqn. (3.26) or (3.27) as appropriate, the remaining weights used in the distance measure must be adapted in an iterative fashion.

Table 3.15 *Iterative adaptation of object weights*

-
1. Compute the value distance tables for each attribute from the complete learning set, according to eqns. (3.26) and (3.27).
 2. Start with an initial data base composed of a small number of objects picked at random from the learning set, and set their initial values of $n(o)$, $correct(o)$, $w(o) (\triangleq \frac{n(o)}{correct(o)})$ to 1.
 3. Consider the learning objects sequentially and insert them one by one in the data base.
 4. Let o denote the next object to insert, and o' its nearest neighbor in the current data base, i.e. minimizing the distance $w(o') \sqrt{\sum_{i=1, \dots, n} \delta_{a_i}(a_i(o), a_i(o'))^k}$.
 5. Increment $n(o')$ by one; if $c(o) = c(o')$ increment also $correct(o')$.
 6. Initialize $n(o)$ to $n(o')$ and $correct(o)$ to $correct(o')$.
 7. Continue at step 4, until the learning set is empty.
-

In the algorithm described in references [CO91] and [SA91b] the weights w_a of individual attributes are kept constant and equal to one while the individual weights of objects are updated in a sequential fashion, as indicated in Table 3.15. The weight of an object is proportional to the ratio of the number of times $n(o)$ it has been used as a nearest neighbor, to the number of times it has been used while leading to a correct decision, $correct(o)$. This allows the influence of exceptional states to be restricted to a small neighborhood.

Equal w_a weights of the attributes corresponds to the assumption that the different attributes provide independent and complementary information on the classification, which may not be valid in practice. In this case a more elaborate technique would consist of determining optimal relative weights of attributes on the basis of the learning set. One of the possible techniques to help choosing the optimal set of weights is discussed in the next section.

3.5.3 Genetic algorithms

Genetic algorithms have been proposed some twenty years ago, as a general model of adaptive behavior for artificial systems, and are loosely based on an analogy with population genetics derived from the Darwinian principle of natural selection [HO 75].

This has led to a general optimization technique which combines ideas from random sampling and hill-climbing methods with the notion of competition. These heuristic methods have shown to be able to provide high quality solutions for many difficult combinatorial optimization problems and are seen as a promising alternative to knowledge directed heuristic search when the prior knowledge is not sufficiently strong to effectively guide the search [GO 89a]. For such problems, the genetic algorithms offer a possibility of collecting and exploiting global problem specific knowledge on-line, during the search process, which is exploited to orient the search into interesting directions.

In addition to the general optimization methods, an important part of the research on genetic algorithms has concentrated on its application to machine learning, yielding a class of genetic algorithm based machine learning techniques [DE 90]. Although these methods are of interest, below we will merely describe the basic idea of the genetic algorithm based *optimization* technique, and then provide some examples of potential applications in the context of the machine learning methods described earlier in this chapter.

Genetic algorithms for general purpose optimization

Our description of Genetic Algorithms (GA) is a summary of a more detailed discussion given in [DE 90]. The interested reader should refer to this reference or to the book by Goldberg [GO 89a] for additional technical details.

Generally speaking, an optimization problem is defined as the search for a solution x^* in some predefined space \mathcal{X} , so as to maximize the value of an optimality criterion $f(\cdot)$ defined in \mathcal{X} . Particularly interesting such problems arise in practice when either the function $f(\cdot)$ is not differentiable or not convex, or when \mathcal{X} is not convex, and there are many local maxima of $f(\cdot)$ of highly variable quality.

In the context of a GA, the elements of \mathcal{X} are represented as strings of characters, and the algorithm manipulates successive generations of *populations* of such strings of characters, trying to find values of maximal $f(\cdot)$. An initial generation population, say $P(0)$, is chosen by sampling strings in \mathcal{X} at random, and the value of $f(\cdot)$ is computed for each such element. At step k , a new generation $P(k)$ is derived from the generation $P(k-1)$ by altering selected states via *genetic operators* (crossover and mutation). Further, mutation operators are chosen at random according to a priori defined parameters, and states are selected according to a random scheme, where the probability of selection is higher for states with higher values of $f(\cdot)$.

The crossover operator consists of selecting two individuals from the current population and combining their string representations to produce a new element. This operator will preserve the similarities among elements and it is necessary to use a mutation operator to generate elements which are significantly different from the current population.

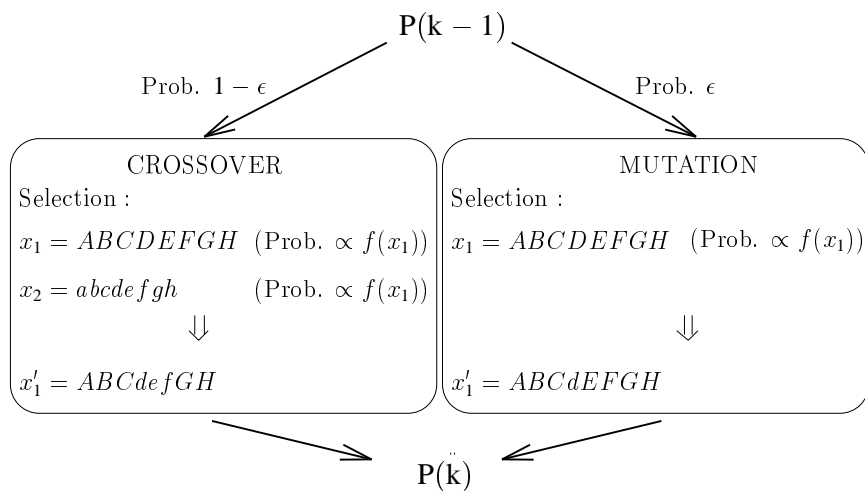


Figure 3.17 Illustration of crossover and mutation operators

These two operators are illustrated in Fig. 3.17. At each step of the basic algorithm a state x_1 is replaced by the result x'_1 obtained by applying the selected operator. The probability ϵ of the mutation operator is generally rather small, for example $\epsilon = 0.1$. The points within the strings where the crossover or mutation operators are applied are chosen at random and the number of different elements in the population is generally kept between 100 and 200.

Of course, many variants of the basic GA have been proposed in the literature, depending on the precise definition of operators, the way operator probabilities are chosen, and the method used to generate a new population at each step, deciding how many and which states are replaced at each step. However, the most important task in applying either variant to a practical problem consists of choosing an appropriate representation of elements in \mathcal{X} in terms of strings and the determination of selection probabilities in terms of the values of the function $f(\cdot)$.

Application to machine learning

As we have indicated in the preceding sections, the machine learning problem is basically an optimization problem, where the objective is to maximize the quality of a rule set or of a decision or class-probability tree. To avoid combinatorial explosion most practical machine learning algorithms use a hill-climbing - at best beam search - strategy and are able only to provide a local optimum of quality with respect to a set of predefined rule modification operators. Since they search only a rather sparse subset of the complete set of possible models, they may be unable to find high quality rules in some practical situations.

Thus, one possible way of applying the GA would “merely” consist of replacing the hill-climbing search strategy by a genetic algorithm operating on an appropriate string encoding of all possible rules or decision trees. This leads to the family of genetic algorithm based machine learning methods discussed in [DE 90].

Another approach consists of using an existing machine learning method to construct a first guess rule, and then to apply the genetic algorithm so as to modify only the *parameters* of the rule.

For example, in the context of the decision trees, a rather straightforward application of this idea would allow us to adapt the attribute thresholds defined at the test nodes of an existing tree in order to improve the tree quality (e.g. defined by eqn. (3.14)) or any other kind of optimality criterion. One could use the same kind of technique as is proposed in [MO 90], where a genetic algorithm is applied to refine the rules defined by experts. The thresholds of the latter rules are adapted on the basis of a sample of pre-classified states so as to yield a pre-specified “false alarms vs non detection” compromise.

The main advantage of this latter approach is that the attribute selection and structure of the rules would be determined once and for all, on the basis of algorithms which are by now well validated in practice. The threshold values could then be adapted via the genetic algorithm for example so as to produce biased versions of the rules, e.g. with reduced non-detection probability of certain classes, or minimizing the expected cost.

A second class of problems, which have not yet received a completely satisfactory solution, concerns the choice of a distance or similarity function, to be used in the context of the instance based learning algorithms. We have seen in the preceding section that this problem may be solved by choosing a set of weights and difference tables $w_a, w(o), \delta_a(v_i, v_j)$ so as to optimize the generalization capabilities of the nearest neighbor rule.

3.6 CONCLUDING REMARKS

Tree induction methods are by now mature techniques, able to handle very large machine learning problems with very good efficiency. Tree quality evaluation, optimal and stop splitting and pruning have been explored in depth by many researchers and satisfactory solutions exist. Possible ways to improve decision trees are still under research : they concern mainly the relaxation of representation constraints (e.g. using several different attributes in a compound test at an interior node of a tree; generalized tree structures) and the use of more powerful optimization techniques able to get closer to the global optima. One of the potential limitations with these techniques is due to the fact that a decision tree provides by construction a complete model of the full attribute space and is unable to restrict its own domain of validity to the regions which are well

enough represented in the learning set.

Complementary to the “general model” philosophy of tree induction methods, are the instance based learning approaches which offer the possibility of using domain knowledge so as to enable local reasoning about differences between the closest matching reference cases in a data base and the current situation. Further, the distance of the current object to its nearest neighbor in a data base may provide information on the degree of confidence one may have in extrapolated information. In the context of security assessment applications this could be applied in order to define indicators to detect situations where it is necessary to use another, more detailed model to assess the security. These possibilities offer admittedly a very promising research avenue.

The rule learning paradigms provide a kind of intermediate compromise between these two extremes. They synthesize the information of a data base up to a certain level, while allowing for the proper handling of exceptions, and should also be able to avoid the construction of overly general rules, not validated in a data base. Thus, together with the instance based learning methods they show promise in further improving the quality of information drawn from the data bases.

On the other hand, while the research on exploiting high level structural and relational descriptions in the context of inductive learning is progressing, the application of such techniques to real-world problems has still a long way to go.

4

Statistical methods

In this chapter we will give a description of some of the classification, regression and clustering techniques grouped under the banner of statistical methods, while the following chapter is devoted to the so-called neural network approaches. One of the common characteristics of these methods is that they handle input information in the form of numerical attributes. Thus all non-numerical information must be translated into numbers via an appropriate coding scheme. In the case of power system problems, this concerns mainly the topological information, which is assumed to be coded by binary 0/1 indicators.

Before starting with the description of the statistical methods, we would like to stress the fact that the distinction between statistical and neural network approaches becomes nowadays quite irrelevant. This will become more obvious in the course of this and the next chapter; it will be further discussed at the end of chapter 5.

Our second remark concerns the choice of methods we have decided to describe and the level of technical details provided for each one. Our choice has been mainly driven by the needs of power system security applications, and our own perception of which methods show promise to fit these needs. This perception is based on the experimentation of the methods on real and synthetic power system security data sets, and in addition, on valuable practical feedback gained from the comparative study made in the STATLOG project with two of our data sets. However, while this is clearly a subjective choice, we believe we have included in our description a representative sample of the methods which show some true potential.

Although our discussion is clearly biased by the specifics of security assessment, we will leave most of the practical considerations to later chapters. Our aim is to provide a mere overview of capabilities of approaches, and not to give a highly technical description. Implementation details have been included only as far as this may clarify some of the ideas and help understand the basic principles.

4.1 INTRODUCTION

Statistical learning techniques have been developed for more than sixty years, for classification, regression and clustering. Since the pioneering work of the late sixties [DY 68], there have also been repetitive attempts to apply these methods to power system security assessment, mainly for fast transient stability analysis [HA 92].

Overall, the statistical approach to learning (or statistical inference) consists of three conceptual steps : (i) probabilistic description of the regression or classification problem by a set of joint probability distributions $p(\mathbf{a}, \mathbf{y})$ or $p(\mathbf{a}, c)$, as appropriate, and formulation of simplifying assumptions, if any, about the structure of the underlying probabilistic process; (ii) estimation of the conditional probability distributions $p(\mathbf{y}|LS, \mathbf{a})$ (resp. $p(c|LS, \mathbf{a})$) of the output variables \mathbf{y} (resp. c) given the learning set information and the value of the attribute vector of a new observation; (iii) use of the latter model for decision making. An in depth discussion of the various approaches and techniques to the estimation problem is given in [DU 73].

In our description we have chosen to classify the statistical methods into parametric and nonparametric ones. The former category concerns methods based on strong hypotheses about a problem and which exploit them in order to define a simplified model in terms of a fixed number of parameters. The latter category concerns methods which make only very non-restrictive assumptions in order to be as general as possible.

Most of the statistical methods require some pre-processing of the data so as to optimize their performance; we will therefore briefly comment the feature pre-whitening, selection and extraction methods which go together with the statistical pattern recognition techniques. They may of course also be useful in the context of the neural network approaches discussed in the next chapter.

4.2 PARAMETRIC METHODS

We will stick to the tradition, according to which the so-called parametric methods concern only the linear and the quadratic models, although other approaches could as well be termed parametric. We consider first the case of linear classification boundaries and discuss two different ways of obtaining this type of classifier. Then we will derive the quadratic discriminant functions from the standard normality assumption and show how this degenerates into a linear discriminant under the hypothesis of identical class-conditional covariance matrices.

4.2.1 Linear discriminant functions

In order to simplify our discussion, we will only consider the case of classification problems with two classes $c(o) \in \{c_1, c_2\}$, and will assume we are searching for a linear classification rule of the form,

$$g(\mathbf{a}(o)) \triangleq \sum_{i=1,n} a_i(o) * w_i + w_0, \quad (4.1)$$

which assigns class c_1 if $g(\mathbf{a}(o)) \geq 0$ and class c_2 otherwise.

Then the practical learning problem amounts to defining the coefficients w_0, \dots, w_n , on the basis of the learning set, so as to maximize the quality of the decision rule.

Although the two methods which we will describe will provide the same criterion under the restricted hypothesis of normal class conditional distributions with an identical covariance matrix, they are characterized by different learning criteria and thus result in different classification boundaries when the latter hypothesis is not verified, which is most often the case in practice.

Let us also notice that there are plenty of other approaches to the design of linear or generalized linear models, as for example the various perceptron models discussed in chapter 5, within the context of neural networks. The interested reader may also consider the references [DU 73, HA 81, DE 82], and the references therein for a more extensive account of such methods.

Fisher's linear discriminant

The basic idea behind the Fisher's linear discriminant is to replace the multi-dimensional attribute vectors by a single feature resulting from a linear transformation of the attributes. Thus the objective is to define a linear transformation maximizing the separation of objects of different classes, on the new feature axis.

Let us define the class conditional means of the attribute vector by

$$\bar{\mathbf{a}}_i \triangleq \sum_{o \in LS; c(o)=c_i} \mathbf{a}(o), \quad (4.2)$$

and the class conditional scatter of the linear projection on vector $\mathbf{w} = (w_1, \dots, w_n)^T$ of the samples, by

$$\tilde{s}_i^2 \triangleq \sum_{o \in LS; c(o)=c_i} (\mathbf{w}^T \mathbf{a}(o) - \mathbf{w}^T \bar{\mathbf{a}}_i)^2. \quad (4.3)$$

Then *Fisher's linear discriminant* is defined by the weight vector \mathbf{w}^* maximizing the following criterion function

$$J(\mathbf{w}) \triangleq \frac{(\mathbf{w}^T \bar{\mathbf{a}}_1 - \mathbf{w}^T \bar{\mathbf{a}}_2)^2}{p_1 \tilde{s}_1^2 + p_2 \tilde{s}_2^2}, \quad (4.4)$$

which is the ratio of the distance of the projected mean vectors to the mean class-conditional standard deviation of projected feature values. (In the pure Fisher's linear discriminant, the classes are supposed to be equiprobable.)

Let us also define the mean class-conditional sample covariance matrix $\hat{\Sigma}_W$ by

$$\hat{\Sigma}_W \triangleq p_1 \hat{\Sigma}_1 + p_2 \hat{\Sigma}_2, \quad (4.5)$$

where the matrices $\hat{\Sigma}_i$ are the sample estimates of the class conditional covariance matrices, obtained by

$$\hat{\Sigma}_i \triangleq \frac{1}{n_i} \sum_{o \in LS; c(o)=c_i} [\mathbf{a}(o) - \bar{\mathbf{a}}_i] * [\mathbf{a}(o) - \bar{\mathbf{a}}_i]^T. \quad (4.6)$$

Then, it may easily be shown that an explicit form is obtained for \mathbf{w}^* by the following formula,

$$\mathbf{w}^* = \hat{\Sigma}_W^{-1} (\bar{\mathbf{a}}_1 - \bar{\mathbf{a}}_2), \quad (4.7)$$

provided that the matrix $\hat{\Sigma}_W$ is non-singular. Otherwise, the optimal direction may be determined by an iterative gradient descent least squares technique [DU 73].

To obtain a decision rule, in addition to choosing \mathbf{w} it is required to define an appropriate threshold w_0 . In the standard Fisher's linear discriminant method, this threshold is chosen directly on the basis of the distribution parameters, in the following way

$$w_0 \triangleq -\frac{1}{2} (\bar{\mathbf{a}}_1 + \bar{\mathbf{a}}_2)^T \hat{\Sigma}_W^{-1} (\bar{\mathbf{a}}_1 - \bar{\mathbf{a}}_2) + \log \frac{p_1}{p_2}. \quad (4.8)$$

However, once the vector \mathbf{w}^* has been fixed, it is a simple scalar optimization problem to choose the appropriate value of w_0 . Therefore, it may be done easily by an optimal threshold search similar to the one described in Table 3.9 for the tree induction methods in chapter 3. The advantage of this method is that it is appropriate for every possible practical optimization criterion.

Illustration. Let us apply Fisher's linear discriminant to the transient stability example of §3.4. We consider the problem of determining an optimal linear classification boundary in the two-dimensional attribute space (*TRBJ, NB_COMP*). Let us determine the optimal linear combination according to Fisher's criterion, and compare it to the optimal linear combination found by the tree algorithm.

The class-conditional means and covariance matrices have been determined in the complete data base composed of 12497 states, and are given by

$$\begin{aligned} \bar{\mathbf{a}}_{Stable} &= \begin{pmatrix} 6533 \\ 8.977 \end{pmatrix} & \hat{\Sigma}_{Stable} &= \begin{pmatrix} 739166.0 & 1733.8 \\ 1733.8 & 8.483 \end{pmatrix} \\ \bar{\mathbf{a}}_{Unst} &= \begin{pmatrix} 7845 \\ 7.542 \end{pmatrix} & \hat{\Sigma}_{Unst} &= \begin{pmatrix} 1009284.0 & 1475.3 \\ 1475.3 & 13.775 \end{pmatrix}. \end{aligned}$$

On the other hand the prior class probabilities are respectively

$$\hat{p}_{Stable} = \frac{3938}{12497} = 0.315; \quad \hat{p}_{Unst} = \frac{8559}{12497} = 0.685$$

and thus the mean class conditional covariance matrix is obtained by

$$\hat{\Sigma}_W = 0.315 * \hat{\Sigma}_{Stable} + 0.685 * \hat{\Sigma}_{Unst} = \begin{pmatrix} 924165.7 & 1556.8 \\ 1556.8 & 12.1077 \end{pmatrix}$$

and the optimal projection direction is obtained by

$$\hat{\Sigma}_W^{-1}(\bar{\mathbf{a}}_{Stable} - \bar{\mathbf{a}}_{Unst}) = \begin{pmatrix} -0.00206617 \\ 0.38413971 \end{pmatrix}.$$

Thus the optimal linear combination direction of the attributes is given by

$$TRBJ + \frac{0.38413971}{-0.00206617} * NB_COMP = TRBJ - 186 * NB_COMP.$$

To determine the corresponding threshold providing the highest *score* in the complete data base, we have used the optimal threshold search for this linear combination attribute. This yielded a threshold of 5903MW corresponding to a test of

$$TRBJ - 186 * NB_COMP < 5903MW,$$

and a score of 0.344. We can compare this with the optimal test of

$$TRBJ - 227 * NB_COMP < 5560MW,$$

found by the linear combination search algorithm of Table 3.10, which obtained a slightly higher score of 0.3646. Notice also that using eqn. (4.8) to compute the threshold results in a value of 5465 MW, corresponding to a score of 0.3291.

Thus in this particular case, Fisher's linear discriminant is only slightly suboptimal provided that the threshold is determined optimally so as to maximize the score value. The slight difference between the two linear combinations is illustrated graphically in Fig. 4.1, showing both boundaries together with a sample of 500 random states drawn from the data base.

While the linear combination attribute is by construction optimal for any used score measure, up to the effectiveness of our iterative search method, the Fisher linear discriminant is not in general optimal with respect to usual score measures. However, its main advantage lies in its direct, non-iterative computation, since, in addition to the mean attribute vectors in each class, it requires only the computation of the inverse of the mean class-conditional covariance matrix. This is actually quite straightforward, provided that the number of attributes is not too large. On the other hand, the iterative linear combination search of Table 3.10 is not applicable to more than two dimensions.

A justification of Fisher's linear discriminant is obtained for the case of Gaussian class-conditional attribute densities with identical covariance matrices. Below, in §4.2.2, we show that in this case Fisher's linear discriminant coincides (asymptotically) with the optimal Bayes decision boundary.

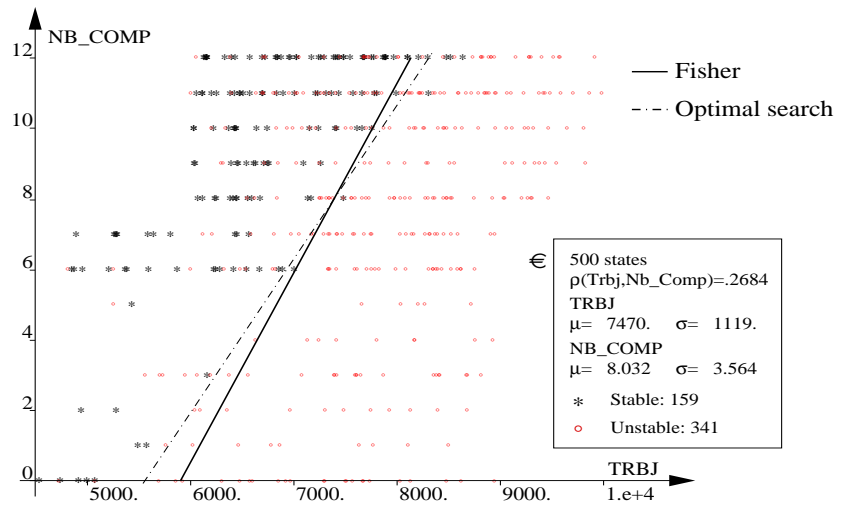


Figure 4.1 *Difference between Fisher and optimal linear discriminant*

Logistic discriminant

Similarly to the above technique, logistic discrimination determines a hyperplane in order to separate classes as well as possible. The main difference comes from the optimality criterion, which is here to maximize the conditional likelihood rather than the quadratic cost function of Fisher's linear discriminant. For convenience, we will again describe the method in the simple two class case.

The working hypothesis behind the logistic discriminant is that the “log odds” of the two classes may be approximated by a linear function

$$\log \frac{P(c_1|\mathbf{a})}{P(c_2|\mathbf{a})} \triangleq \mathbf{w}^T \mathbf{a} + w_0. \quad (4.9)$$

This, together with the constraint $P(c_1|\mathbf{a}) + P(c_2|\mathbf{a}) = 1$, leads to the following parametric expression for the conditional class probabilities

$$P(c_1|\mathbf{a}) = \frac{\exp(\mathbf{w}^T \mathbf{a} + w_0)}{1 + \exp(\mathbf{w}^T \mathbf{a} + w_0)}, \quad (4.10)$$

$$P(c_2|\mathbf{a}) = \frac{1}{1 + \exp(\mathbf{w}^T \mathbf{a} + w_0)}. \quad (4.11)$$

Given a learning set of preclassified examples, the optimality criterion used for estimating the parameters w_0 and \mathbf{w} is to maximize the conditional likelihood

$$L(LS|w_0, \mathbf{w}) = \prod_{o \in LS; c(o)=c_1} P(c_1|\mathbf{a}(o)) \prod_{o \in LS; c(o)=c_2} P(c_2|\mathbf{a}(o)), \quad (4.12)$$

Table 4.1 *Fisher vs logistic linear discriminant. Adapted from [TA 94]*

Problem	Fisher		Logistic	
	P_e (test)	Train CPU	P_e (test)	Train CPU
Transient stability	4.1%	107.5 sec	2.8%	336.0 sec
Voltage security	2.5%	73.8 sec	0.7%	130.4 sec

or equivalently its logarithm, which is equal to the residual entropy in the learning set

$$\log L(LS|w_0, \mathbf{w}) = \sum_{o \in LS; c(o)=c_1} \log P(c_1|\mathbf{a}(o)) + \sum_{o \in LS; c(o)=c_2} \log P(c_2|\mathbf{a}(o)). \quad (4.13)$$

The logistic discriminant is determined, using an iterative gradient descent or Newton approach to search for the optimal values of u_0 and \mathbf{w} .

Again, in the case of Gaussian class-conditional attribute distributions $p(\mathbf{a}|c_i)$ with equal covariance matrices, it may be shown that this method will produce the same optimal, linear discriminant as the preceding technique. However, the logistic discriminant covers also the case of class-conditionally independent binomial (0/1) attributes.

The logistic discriminant may be seen as a particular case of a generalized linear regression model, where the regression variable is the class indicator variable [MC 52].

Difference between logistic and linear discriminants

It is interesting to notice that in practice there may be important differences in performance between the above two approaches. To illustrate this, we give in Table 4.1 the results obtained in the context of two power system security problems, where the normality or independence assumptions are clearly violated. They have been obtained within the Statlog project, using the two power system security data sets described respectively in §13.3 and §14.2. As it was put by the statistician in charge of the project, “the difference in performance is undoubtedly due to the non-Gaussian nature of some of the variables”.

4.2.2 Quadratic and generalized linear discriminants

Quadratic discriminants are optimal in the case of Gaussian class conditional attribute distributions. Otherwise, they present a generalization of linear discriminants, by allowing us to take into account correlations among attributes by second order terms. However, not all second order terms need to be incorporated in the discriminant function and thus the number of parameters may be controlled.

Gaussian class conditional attribute distributions

Let us consider the case where the class conditional attribute distributions are Gaussian, defined by

$$p(\mathbf{a}|c_i) \triangleq \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp \left[-\frac{1}{2} (\mathbf{a}(o) - E\{\mathbf{a}|c_i\})^T \Sigma_i^{-1} (\mathbf{a}(o) - E\{\mathbf{a}|c_i\}) \right]. \quad (4.14)$$

Then the Bayes decision rule, yielding minimum error rate is obtained by choosing the class c_j such that the value of its posterior probability

$$P(c_j|\mathbf{a}) = \frac{p(\mathbf{a}|c_j) * p_j}{p(\mathbf{a})}, \quad (4.15)$$

is maximal, or equivalently such that

$$g_j(\mathbf{a}) = \log p(\mathbf{a}|c_j) + \log p_j, \quad (4.16)$$

is maximal, since $p(\mathbf{a})$ is independent of the class c_j . This is equivalent to maximizing

$$g_j(\mathbf{a}(o)) \triangleq -\frac{1}{2} (\mathbf{a}(o) - E\{\mathbf{a}|c_j\})^T \Sigma_j^{-1} (\mathbf{a}(o) - E\{\mathbf{a}|c_j\}) - \frac{1}{2} \log |\Sigma_j| + \log p_j, \quad (4.17)$$

where we have dropped the term $\frac{n}{2} \log 2\pi$ which is independent of the class.

Thus, to each class c_i corresponds a quadratic function $g_i(\mathbf{a}(o))$, and the equalities $g_i = g_j$, lead to quadratic decision boundaries in the general case.

However, these quadratic hypersurfaces degenerate into linear hyperplanes when the class-conditional covariance matrices are identical. Indeed, let us consider the two-class case, when $\Sigma_1 = \Sigma_2 = \Sigma$. Then the Bayes optimal decision rule is to decide class c_1 , whenever $g_1(\mathbf{a}) - g_2(\mathbf{a}) > 0$, namely when

$$\mathbf{a}^T(o) \Sigma^{-1} (E\{\mathbf{a}|c_1\} - E\{\mathbf{a}|c_2\}) > \frac{1}{2} (E\{\mathbf{a}|c_1\} + E\{\mathbf{a}|c_2\})^T \Sigma^{-1} (E\{\mathbf{a}|c_1\} - E\{\mathbf{a}|c_2\}) + \log \frac{p_2}{p_1}. \quad (4.18)$$

In particular, the direction of the optimal hyperplane is identical to Fisher's linear discriminant direction.

In practice, the quadratic discriminant may in principle be directly determined by estimating the class-conditional mean vectors and covariance matrices and substituting in the above formula. However, it is often preferable to use an iterative gradient descent least squares technique, which appears to be more robust than the direct approach, and allows for some interesting generalizations, such as the classical sequential forward or backward iterative least squares techniques.

Generalized linear discriminants

It is a well known fact that in the context of high dimensional attribute spaces, the quadratic discriminant may fail due to its very high number of parameters. In particular, obtaining a reasonable estimate of the covariance matrices would often require too many data points. Classical approaches to solve this “curse of dimensionality” problem are the feature selection and extraction techniques briefly discussed below in §4.5.

Other approaches consist of simplifying the quadratic model, either in a backward or in a forward approach. The former starts with the full quadratic model and removes iteratively the terms in the discriminant functions $g_i(\mathbf{a})$ which do not significantly improve the accuracy. The latter approach complicates the linear model sequentially, by introducing the quadratic terms progressively in the discriminant function and stopping as soon as the performance stops improving. A further generalization of these approaches uses arbitrary (e.g. orthonormal) polynomials at each step which leads to the general family of sequential or stepwise least squares techniques.

Although these - actually nonparametric - methods certainly have much potential, they have become less popular in the recent years, in particular due to the recent emergence of the neural network approaches, which are similarly general.

Below, within the class of nonparametric methods, we will describe the *projection pursuit* technique, which is a very powerful and attractive approach to generalized linear discrimination or regression.

4.2.3 Conclusion

The high non-linearity, variability and dimensionality of power systems, which we have to face in the context of our security problems would certainly prevent the above discussed parametric methods from being very general useful tools. In addition, as we have illustrated, since different learning criteria may lead to completely different results in terms of performances, it could be difficult to select an appropriate criterion for each new power system and each new security problem.

In other words, some of these methods may work quite well in some particular situations, but we don't expect them to be robust enough to become a general stand alone tool. However, since they are standard and easy to apply techniques, a reasonable approach could be to include them in a tool-box, and when a new problem is encountered try them out in a preliminary study. If they don't work properly, a more powerful nonparametric approach must be used instead, otherwise they may be used as an auxiliary tool, for instance to determine interesting linear combinations of attributes.

The results corresponding to the voltage security problem indicated in Table 4.1 above, show that the logistic discriminant may occasionally be very accurate. Indeed, the test

set error rate of 0.7% outperforms 20 of the 21 other tested methods in the Statlog project. The only method which could reach the performance of the logistic discriminant in this problem was the projection pursuit method SMART, which obtained a test set error rate of 0.6%.

4.3 NONPARAMETRIC METHODS

We will mainly describe two popular approaches to nonparametric classification or regression.

On the one hand, the nearest neighbor methods are very simple to implement, but also very sensitive to the choice of attribute representation. Their main attractive feature is that they provide information about the distance of an object to the nearest neighbor in the data base, and this distance may provide some information about the confidence with which information may be extrapolated.

On the other hand, the projection pursuit technique is an iterative, and computationally intensive procedure to derive a non-linear model to represent the data. This method, as we will see, offers also some data exploration features. While its principle is closely connected to the neural network approaches discussed in the next chapter, it seems more powerful in terms of accuracy and able to provide easier interpretable information, in a fashion similar to the machine learning approaches.

Finally, we will briefly indicate the principle of some other frequently used nonparametric techniques, such as the kernel density estimators and the naive Bayes approach.

4.3.1 The nearest neighbor class of methods

Nearest neighbor (*NN*) methods have been applied both for density estimation, classification and regression. We discuss only the latter two applications.

Classification

Given a learning set LS and a distance Δ defined in the attribute space, the nearest neighbor classifier consists of classifying an object o in the class $c(o')$ of the learning state o' of minimal distance, i.e.

$$o' = \arg \min_{LS} \Delta_a(o, o'). \quad (4.19)$$

Asymptotically, when the LS size $N \rightarrow \infty$, the nearest neighbor o' converges towards the object o . Thus its class $c(o')$ has an expected asymptotic probability of being the

correct class $c(o)$ equal to

$$\sum_{i=1,m} p_i(\mathbf{a}(o)) * p_i(\mathbf{a}(o)). \quad (4.20)$$

From this, it may be derived that in an m class problem the asymptotic error rate of the nearest neighbor rule is upper bounded by [DE 82]

$$P_e^{NN} \leq P_e^{Bayes} \left(2 - \frac{m}{m-1} P_e^{Bayes} \right). \quad (4.21)$$

This suboptimality of the nearest neighbor is a kind of overfitting problem. It is indeed related to the fact that the NN rule extrapolates the classification of the sample without any smoothing. It is interesting to observe that this overfitting suboptimality remains a problem even for very large samples.

The first approach to solving this problem consists of reducing the locality of the NN information by using more than one nearest neighbor. This leads to the so-called $K - NN$ or $(K, L) - NN$ rules [DE 82].

The basic $K - NN$ rule consists of searching for the K nearest neighbors of an attribute vector and estimates the class probabilities by

$$\hat{p}_i(\mathbf{a}(o)) \triangleq \frac{n(K, o, c_i)}{K}, \quad (4.22)$$

where $n(K, o, c_i)$ denotes the number of learning states of class c_i among the K nearest neighbors of o .

Asymptotically, the $K - NN$ is Bayes optimal, strictly speaking if the number K increases with N , such that

$$\lim_{N \rightarrow \infty} K(N) = \infty \text{ and} \quad (4.23)$$

$$\lim_{N \rightarrow \infty} \frac{K(N)}{N} = 0. \quad (4.24)$$

Indeed, $\lim_{N \rightarrow \infty} \frac{K(N)}{N} = 0$ guarantees that the K nearest neighbors still converge towards the object o in the attribute space, while $\lim_{N \rightarrow \infty} K(N) = \infty$ guarantees that the class-probability estimates converge towards the true values. In practice, in the finite sample case there exists generally an optimal value of K , above which the smoothing effect becomes too strong and leads to a decrease in performance.

The second approach to improve the NN rule consists of editing the learning set by removing those learning states which are surrounded by states of a different class. This consists of increasing the probability of the nearest neighbor to belong to the majority class, and thus leads to nearly optimal decision rules.

In addition to these editing techniques, condensing algorithms may be used to dramatically reduce the size of the required data base, by removing the states which do not

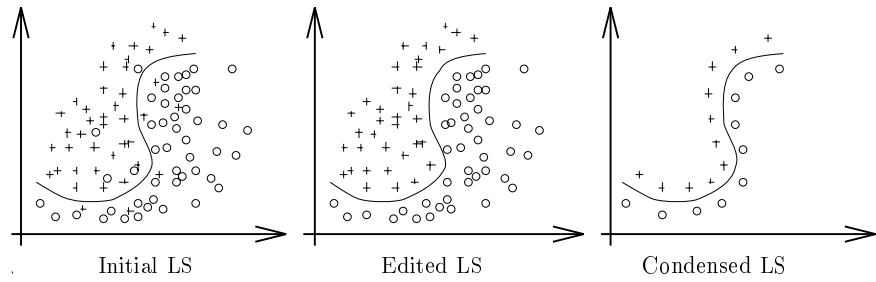


Figure 4.2 *Nearest neighbor, editing and condensing*

contribute to defining the decision boundary. Figure 4.2 illustrates graphically the editing and condensing techniques, which are discussed in full detail in [DE 82]. It should be noted that while these techniques improve error rates and particularly dramatically reduce the CPU times, they unfortunately strongly reduce the locality of the nearest neighbor classifier, which is however a desirable practical feature of the NN method, as we discuss below.

Regression

Another possible use of the nearest neighbor approach is for regression problems. In this case the following type of regression function may, for example, be used

$$\mathbf{r}(o) \triangleq \frac{\sum_{o' \in K - NN(LS, o)} \mathbf{y}(o') \Delta^{-1}(o, o')}{\sum_{o' \in K - NN(LS, o)} \Delta^{-1}(o, o')} \quad (4.25)$$

where we have denoted by $K - NN(LS, o)$ the set of the K nearest neighbors of o .

Discussion

The nearest neighbor rule is a very simple and easy to implement approach. It has the main disadvantage of requiring a very large number of learning states to become robust with respect to the definition of distances. In particular, in the case of high dimensional attribute spaces the method may rapidly require prohibitively large samples. Thus, to be effective it must in general rely on prior feature selection and/or extraction techniques, so as to reduce the attribute space dimensionality.

At the same time, while the learning of the basic nearest neighbor rule merely consists of storing the data base, the complexity of using this information for new classifications is directly proportional to the product of the number N of learning states and the dimension n of the attribute space. This may be several orders of magnitude higher than the time required by competing techniques and only rather sophisticated search algorithms can allow us to reduce the CPU time. Nevertheless, in the context of power

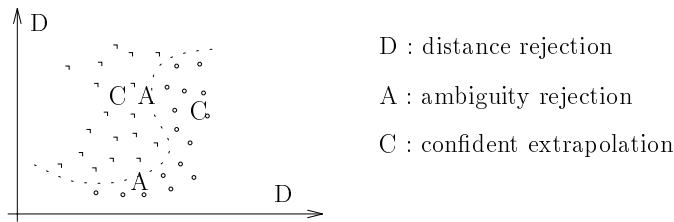


Figure 4.3 *Nearest neighbor ambiguity and distance rejection*

Table 4.2 *Error rates (%) of $K - NN$ classifiers*

K	1	3	5	7	9
28 attributes	5.92	5.20	4.80	4.06	4.48
3 attributes	4.24	3.12	3.04	2.80	2.96

system security assessment this would not be a very strong limitation, thanks to the increased CPU speeds and the relatively limited size of data bases to several thousand states. However, in applications like image or printed character recognition, where data bases of several millions of objects are frequent, this becomes one of the main concerns of this method.

As we have already mentioned for the instance based learning approaches discussed in the preceding chapter, the main practical attractiveness of this approach is related to the local identification of the reference cases of a data base, on the basis of which a diagnostic is made. In a practical security assessment environment, these nearest reference cases may be supplied to the operator as an explanation or justification of the security characterization made for the current system state. In particular, the main differences with the current situation may be analyzed so as to decide whether and how their information may be extrapolated. This could, for example, allow us to use local linear approximation techniques, so as to infer security margins, and provide rejection options for states either too close to the classification boundary (ambiguity rejection) or too far away from any reference case (distance rejection) as illustrated in Fig. 4.3.

As an illustration of the typical behavior of the $K - NN$ method, let us look at the voltage security assessment example of §10.2 considered in Table 4.1. Table 4.2 shows the influence of K on the test set error rates obtained for two different sets of attributes. In each case the standard Euclidean distance was used, and the attributes were normalized by dividing their value by their standard deviation, as is described in §4.5.1.

These results illustrate how increasing the value of K allows us to reduce the error rate. They suggest that for both sets of attributes the optimal value of K is equal to 7. It is also interesting to note that reducing the number of attributes has allowed us to significantly improve the performance, both in terms of reliability and efficiency.

In this particular example, we have used the TDIDT method to build a tree so as to identify among the 28 attributes the 3 most significant ones.

4.3.2 Projection pursuit

The projection pursuit regression technique models a regression function $r(\cdot)$ as a linear combination of smooth functions of linear combinations (or projections) of the attribute values. Thus the model assumes the following formulation

$$\mathbf{r}(\mathbf{a}) \triangleq \bar{\mathbf{y}} + \sum_{i=1,k} \mathbf{v}_i f_i(\mathbf{w}_i^T \mathbf{a}), \quad (4.26)$$

where the order k , the r -vectors \mathbf{v}_i , the n -vectors \mathbf{w}_i and the scalar functions $f_i(\cdot)$ are determined on the basis of the learning set, in an iterative attempt to minimize the mean square error

$$MSE(\mathbf{r}) \triangleq \sum_{o \in LS} \|\mathbf{y}(o) - \mathbf{r}(\mathbf{a}(o))\|^2. \quad (4.27)$$

For classification problems, the standard class-indicator encoding is used, which is defined by

$$y_i(o) = \delta_{c(o), c_i}, \quad \forall i = 1, \dots, m. \quad (4.28)$$

In the basic approach the functions f_i are special scatter-plot smoothers, which are normalized in the following way

$$\sum_{o \in LS} f_i(\mathbf{w}_i^T \mathbf{a}(o)) = 0 \quad \text{and} \quad \sum_{o \in LS} f_i^2(\mathbf{w}_i^T \mathbf{a}(o)) = 1, \quad (4.29)$$

and the projection vectors \mathbf{w}_i are normed

$$\sum_{j=1,n} w_{ij}^2 = 1. \quad (4.30)$$

The striking similarity of this model with a single hidden layer feed-forward neural network is shown in Fig. 4.4. However, the originality of the projection pursuit regression technique is that both model complexity (the order k) and the smooth activation functions $f_i(\cdot)$ are determined on the basis of the learning set data, while in the basic multi-layer perceptron they are chosen a priori by the user, which leads in general to overly complex structures with many redundant parameters.

Forward growing of projection pursuit

At each step j of the procedure, the order of the model is increased by one unity, by adding an additional projection direction w_j and smooth function f_j and determining

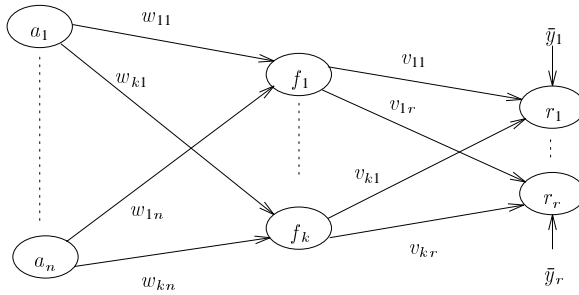


Figure 4.4 Graphical representation of the projection pursuit model

the vector v_j . During this first step, the parameters of the preceding directions are kept constant.

The second step consists of adjusting in a backfitting approach all the parameters of all directions $k \leq j$ in a cyclic fashion, so as to minimize the MSE (4.27).

Finally, the model growing procedure stops when the MSE is sufficiently low or when it does not improve sufficiently anymore.

A complementary approach to the above growing consists of generating the models in decreasing order of their complexity k , by starting with a sufficiently high value of k and pruning at each step the least active part of the model, corresponding to the projection direction which influences the least strongly the output values. This is defined as the direction i which minimizes the sum

$$I_i \triangleq \sum_{j=1,r} |v_{ij}|. \quad (4.31)$$

Backfitting

The heart of the algorithm consists of backfitting a group of parameters, w_i , f_i , and v_i , corresponding to one of the current projection directions $i \leq j$. This is done in an iterative fashion.

1. Adjusting v_i is done directly by setting the derivatives of the MSE to zero with respect to each component of v_i . This yields a linear equation, since the MSE is quadratic in v_i .
2. To adjust the smooth functions $f_i(\cdot)$, we proceed in two steps. First, non-smooth function values $f_i(\mathbf{w}_i^T \mathbf{a}(o))$ are determined for each object $o \in \mathbf{LS}$. Again, since the MSE is quadratic in f_i , this can be done in a direct linear computation, setting the partial derivatives of the MSE w.r.t. $f_i(\mathbf{w}_i^T \mathbf{a}(o))$ ($\forall o \in \mathbf{LS}$) to zero. Second,

the resulting “optimal” values

$$\left(\mathbf{w}_i^T \mathbf{a}(o), f_i^*(\mathbf{w}_i^T \mathbf{a}(o)) \right), \quad \forall o \in LS, \quad (4.32)$$

are used as target values to determine the smooth interpolation function. We refer the interested reader to [HW 93] for a further discussion of various alternative schemes for this unidimensional smoothing.

3. Finally, to adjust the projection direction \mathbf{w}_i , an iterative gradient descent or Newton method should be used, since the MSE is not a quadratic function of \mathbf{w}_i .

Discussion

One of the advantages of the *projection pursuit* regression method with respect to standard feed-forward neural network techniques lies in the greater simplicity of the resulting structure. This is due to the automatic determination of the neuron activation function together with the adaptation of the model complexity to the data. While similar neural network growing techniques have been proposed in the literature, the projection pursuit approach has been found to be superior in performance to the cascade correlation techniques proposed by Fahlman and Lebière for neural networks [FA 90]. Actually the main motivation of cascade correlation is to increase the speed of learning convergence and not so much to improve the model accuracy.

Admittedly, in high dimensional attribute spaces the projection directions found by this method may become difficult to interpret. Thus, Friedman and Stuetzle have proposed various extensions to the basic method to improve its data exploration features [FR 81]. For example, by restricting the number of attributes combined in any projection, the method may provide interesting two or three dimensional directions for data exploration. With these extensions this method would provide similar features to the TDIDT approaches discussed in the preceding chapters, with the additional capability of providing a *smooth* non-linear input/output modelling capability, which would be particularly interesting for the estimation of power system security *margins*.

The SMART implementation of the projection pursuit regression technique was applied, in the context of the Statlog project, on the two above-mentioned power system security classification data sets. In both cases this method scored best in terms of reliability (but also slowest in terms of learning CPU time). This, in addition to the possibility of exploiting the continuous security margins, provides a strong motivation for further exploration of the capabilities of these projection pursuit approaches in the context of power system security problems.

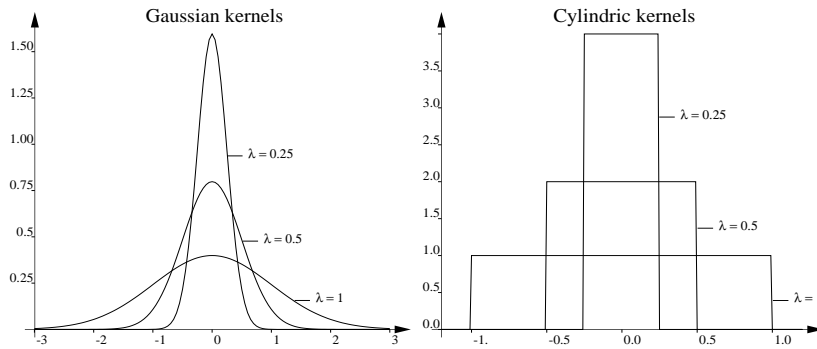


Figure 4.5 Various kernel functions and smoothing parameters

4.3.3 Other techniques

Below we group together three other classical nonparametric techniques. Although we do not believe they would be very useful in our practical context (i.e. more accurate or complementary in terms of functionality), we provide a brief description since some of the results discussed later pertain to one of these methods.

Kernel density estimation

While the $K - NN$ and projection pursuit methods aim at directly modelling the conditional class-probabilities $p_i(\mathbf{a})$, the kernel density estimation approach operates indirectly by providing a nonparametric estimate of the class conditional attribute densities $p(\mathbf{a} | c_i)$.

This approach uses the following expansion

$$\hat{p}(\mathbf{a} | c_i) \triangleq \frac{1}{n_{i.}} \sum_{o \in LS; c(o)=c_i} \phi(\mathbf{a}, \mathbf{a}(o), \lambda), \quad (4.33)$$

where the function $\phi(\cdot, \mathbf{a}(o), \lambda)$ is a kernel function centered at $\mathbf{a}(o)$, and λ is a smoothing parameter. Various kernel functions are suggested in Fig. 4.5 together with the effect of the smoothing parameter.

In addition to different possible choices for the kernel function, discussed in most pattern recognition textbooks [DU 73, HA 81, DE 82], it is also important to choose the smoothing parameter to adapt the method to the data. Actually, it turns out that the choice of the smoothing parameter, which is the kernel density version of our by now familiar overfitting problem, is much more important in practice than the choice of the type of kernel function.

Various techniques have been proposed to estimate the value of λ on the basis of the data. One possibility consists of maximizing the “leave-one-out” sample likelihood,

defined by

$$L(LS|\lambda) \triangleq \prod_{o \in LS} \hat{p}'(\mathbf{a}(o)|c(o)) \quad (4.34)$$

where $\hat{p}'(\mathbf{a}(o)|c(o))$ is the density estimate at point $\mathbf{a}(o)$ for class $c(o)$, obtained when removing the object o from the learning set, i.e.

$$\hat{p}'(\mathbf{a}(o)|c(o)) \triangleq \frac{1}{n_i - 1} \sum_{o' \in LS; c(o')=c(o); o' \neq o} \phi(\mathbf{a}(o), \mathbf{a}(o'), \lambda). \quad (4.35)$$

The expression (4.34) may then be optimized with respect to λ by a one dimensional numerical search technique in the semi space $\lambda \in]0 \dots \infty[$.

Histogram

A very simple approach to nonparametric density estimation is the histogram approach.

Basically, this method consists of dividing a priori the attribute space into subregions and counting the relative number of states of each class falling into each subregion. In the simplest case the regions are defined by dividing the range of each interval into a fixed number of regular sub-intervals. The advantage of this approach with respect to kernel density estimation or nearest neighbor is that it does not require to store any of the learning states.

However, in order to make this approach applicable in the case of multidimensional attribute spaces, the size of the elementary regions must be adapted to the learning set representativity, in particular to avoid empty regions and to minimize the variations among neighboring cells. This is the histogram version of the overfitting problem, for which, not surprisingly, smoothing solutions have been proposed in the literature [HA 81]. In spite of these improvements, we believe that the approach is mainly useful in one, two or three dimensional situations. A particular situation where this is useful is discussed in the next paragraph, in the context of the “naive” Bayes approach.

Illustration. Figure 4.6 illustrates the two-dimensional histograms in the $(TRBJ, NB_COMP)$ space, for the 3938 stable and 8559 unstable states of the data base corresponding to our transient stability example of §3.4. Using these histograms as a classifier amounts to classifying into the stable class if an object belongs to a cell where the number of stable states is higher than the number of unstable states. Thus, the histogram classifier basically consists of dividing the attribute space into a number of regularly distributed *predefined* cells, counting the number of learning states of each class belonging to each such cell, and associating the majority class in the corresponding learning subset to each cell.

This classification is shown in Fig. 4.7, where the regions corresponding to empty cells are labelled “unknown”. It is interesting to notice that these regions, falling outside of

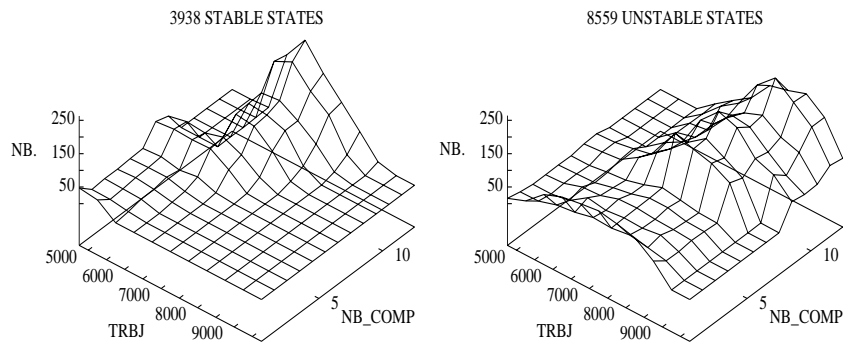


Figure 4.6 Example two-dimensional histograms

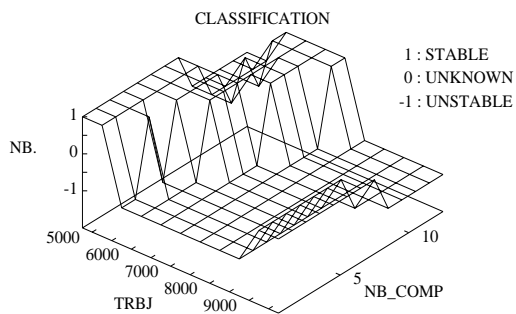


Figure 4.7 Classification corresponding to the histograms of Fig. 4.6

the sampled region cannot be identified straightforwardly with a tree classifier, which would extrapolate the classification of the neighboring cells.

Naive Bayes

An interesting situation occurs when the attributes are independently distributed in each class. Then the class conditional probability densities may be factorized in the following way

$$p(\mathbf{a}|c_i) = \prod_{j=1,n} p(a_j|c_i), \quad (4.36)$$

and the multi-dimensional estimation problem reduces to $m \times n$ uni-dimensional density estimations, $p(a_j|c_i)$ for each attribute.

In particular, for discrete attributes this amounts to counting the number of occurrences of each possible attribute value in each class and use these values in order to estimate the corresponding probabilities in a maximum likelihood or Bayesian approach.

For real valued attributes either a parametric estimator or one of the above nonpara-

metric techniques ($K - NN$, kernel density estimation or histograms) may be used in order to estimate the scalar probability density function $p(a_i|c_i)$. The most straightforward nonparametric technique is in this particularly simple uni-dimensional case the histogram approach, due to its simplicity and computational efficiency.

It is interesting to notice that the above naive Bayes classifier leads to generalized linear discriminant functions. Indeed, replacing the probabilities $p(a_j|c_i)$ by their estimate $\hat{p}(a_j|c_i)$ and taking the logarithm yields

$$\log p(c_i|\mathbf{a}) \propto \log p_i + \log p(\mathbf{a}|c_i) = \log p_i + \sum_{j=1,n} \log \hat{p}(a_j|c_i). \quad (4.37)$$

4.4 CLUSTERING METHODS

While supervised learning techniques obviously aim at producing a model for a particular relationship which is assumed to exist between the input attributes and the output classification or regression function, clustering or unsupervised learning aims essentially at uncovering such relationships among groups of data points or among groups of attributes used to describe them.

The clustering methods are therefore one of the basic pre-processing techniques used in the context of statistical data analysis and learning approaches. They aim at identifying groups of correlated variables or regions of similar objects in the attribute space. Discovering such similarities may allow us to compress the information by replacing individual objects by prototypes and individual parameters by representative features. This simplification may have very drastic implications, for example in terms of supervised learning speed and effectiveness.

Unfortunately the theoretical justifications of the various practical clustering techniques are rather weak [DU 73]. Thus, below we will first describe some classical approaches in the context of clustering of *objects* in a given number of groups and then give some indications on how to determine an appropriate number of clusters. The same methods may also be applied to the clustering of attributes, as we will illustrate on the basis of our example problem of transient stability assessment.

4.4.1 Algorithms of dynamic clusters

Given a set of objects, and a number K fixed a priori by the user, the *ISODATA* and *K-means* procedures determine a set of K clusters, so as to represent most effectively the prior distribution in the attribute space $p(\mathbf{a})$ by the cluster prototypes or centers.

In these methods, a cluster is defined by its *prototype* and its members are the learning states which are most *similar* to the cluster prototype. The iterative algorithms stop as

soon as a stable partition of the data has been found.

In the basic algorithm, a prototype is defined as the mean attribute vector of a cluster and the similarity is defined as the Euclidean distance. This leads to the basic ISODATA and K -means algorithms searching for clusters minimizing the following quadratic quantization error criterion

$$J_e = \sum_{i=1, K} J_i, \quad (4.38)$$

where J_i denotes the quantization error of the cluster i , defined by

$$J_i \triangleq \sum_{o \in LS: o \in \text{Cluster}_i} \|\mathbf{a}(o) - \bar{\mathbf{a}}_i\|^2, \quad (4.39)$$

$\bar{\mathbf{a}}_i$ denoting the center or prototype of the i -th cluster.

This criterion is clearly sensitive to the normalization of the attributes, and thus the clusters found may strongly depend on the normalization. In order to achieve invariance, one should therefore transform the attributes using one of the techniques described in §4.5. This may however also be detrimental in some situations. Thus the definition of a clustering criterion is essentially a problem solved in an empirical, pragmatic trial and error fashion.

The so-called *dynamic clustering algorithm* is a generalization of the ISODATA method, which allows us to use a general class of kernels for representing prototypes and employs a more general similarity based optimality criterion [DE 82].

ISODATA

In the ISODATA algorithm, the cluster centers are adapted iteratively in the following *batch* fashion.

1. Choose the initial cluster prototypes randomly or on the basis of prior information.
2. Classify all the learning states by allocating them to the closest cluster.
3. Recompute the K prototypes on the basis of their corresponding learning states.
4. If at least one cluster prototype has changed, return to step 2, otherwise stop.

K -means

This quite similar approach starts with the definition of the initial clusters as given sets of objects, and operates schematically in the following sequential fashion.

1. Start with a random partition of the data into K clusters, and compute the corresponding cluster centers as the means of each cluster's objects' attribute vectors.

2. Select the next candidate object o from the learning set, and let i be its current cluster.
3. (a) If o is in a single object cluster then this remains unchanged.
 (b) Otherwise find the cluster j which results in a minimum overall quantization error J_e , if object o is moved from cluster i to cluster j . If $i \neq j$ move the object and adapt both cluster centers.
4. If J_e has remained unchanged during a complete pass through the learning set then stop, otherwise return to step 2.

This latter approach has the advantage of being sequential and thus may be applied in real time, in order to adapt the clustering to new incoming objects. Its main disadvantage, with respect to the ISODATA batch algorithm is its higher susceptibility of being trapped in local minima [DU 73].

Determining the right number of clusters

In practice the number of clusters is often unknown and must also be determined on the basis of the data. The classical approach to this problem consists of applying either of the above algorithms repeatedly with a growing (or decreasing) number of clusters K .

In practice, for each value of K a performance measure is computed for the corresponding clusters obtained. For example, in the above mean square error framework, the overall quantization error $J_e(K)$ could be used for this purpose. The $J_e(K)$ criterion decreases towards zero when K increases and an appropriate number of clusters may be selected by detecting the value of K corresponding to a “knee” in the $J_e(K)$ curve, above which J_e decreases much more slowly.

4.4.2 Hierarchical agglomerative clustering

Hierarchical clustering aims at defining a sequence of clusterings for $K \in [1 \dots N]$, so that clusters form a nested sequence, i.e. such that objects which belong to a same cluster at step K remain in the same cluster at step $K - 1$.

The top down or divisive approach consists of generating this sequence in the order of increasing values of K . In the bottom up or agglomerative approach, objects are progressively merged in a step-wise fashion. We briefly describe and illustrate the latter.

The agglomerative algorithm starts with the initial set of N objects, considered as N singleton clusters. At each step it proceeds by identifying the two most similar clusters and merging them to form a single new cluster. This process continues until all objects

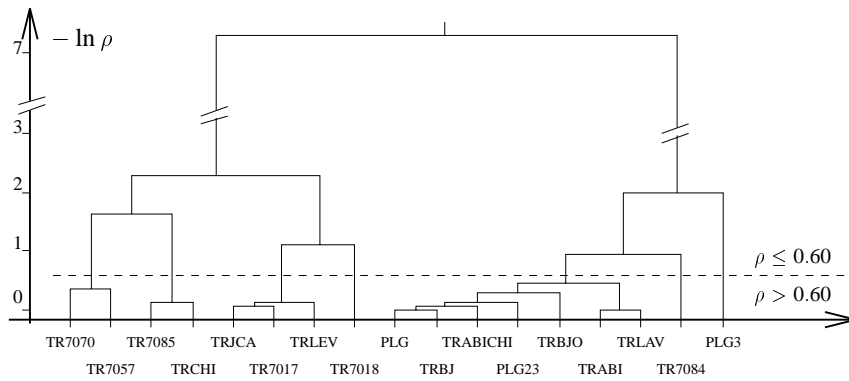


Figure 4.8 Hierarchical attribute clustering example

have been merged together in a single cluster. Cluster similarity may be defined in various ways, for example combining object similarities in the following way

$$SIM_{\min}(\text{Cluster}_i, \text{Cluster}_j) = \min_{o_i \in \text{Cluster}_i; o_j \in \text{Cluster}_j} SIM(o_i, o_j), \quad (4.40)$$

$$SIM_{\max}(\text{Cluster}_i, \text{Cluster}_j) = \max_{o_i \in \text{Cluster}_i; o_j \in \text{Cluster}_j} SIM(o_i, o_j). \quad (4.41)$$

The resulting hierarchical clustering may be represented by a dendrogram which shows graphically the hierarchical groupings of objects along with the cluster (dis)similarities. This is particularly interesting for the analysis of attribute similarities or when the number of objects to cluster is small.

Illustration. An example dendrogram, built for the hierarchical clustering of attributes for the transient stability example of §3.4 is represented in Fig. 4.8.

For this illustration, we have considered a selection of 14 power flow and 3 power generation attributes chosen among the 67 candidate attributes. The similarity among the attributes was defined as their correlation coefficient (see eqn. (2.11)) which was estimated for each pair of attributes on the basis of the 12497 operating states of the data base.

The similarity of two subsets of attributes S_1 and S_2 was defined as the minimum similarity of pairs of attributes of the two subsets, using eqn. (4.40). To improve the graphical rendering we have used the logarithm of the inverse of this similarity measure to draw the dendrogram in Fig. 4.8. Thus the vertical position of the line merging to clusters represents the following quantity

$$\text{Distance}(S_1, S_2) \triangleq -\ln \left(\min_{a_1 \in S_1; a_2 \in S_2} |\hat{\rho}(a_1, a_2)| \right). \quad (4.42)$$

It is interesting to observe from the dendrogram that PLG, TRBJ, TRABICHI, PLG23,

TRBJO, TRABI and TRLAV form a rather homogeneous group of similar attributes, the correlation being at least equal to 0.60 for each pair. Actually, a closer look at these attributes shows that they all correspond either to generations of the LaGrande power plant or to North to South power flows in James's Bay corridor.

Similarly, the group of attributes TRJCA, TR7017 and TRLEV are related to lines within the Québec to Montréal corridor, which are shared by the North to South and the West to East corridors of the Hydro-Québec system.

4.4.3 Mixture distribution fitting

To conclude our brief introduction to clustering techniques let us mention an important family of methods which approach the clustering problem in a probability distribution fitting paradigm.

In this framework one considers the hypothesis that the learning sample was generated by a probability distribution $p(\mathbf{a})$ which is supposed to be a mixture of K elementary probability densities corresponding to the elementary underlying classes under investigation.

To illustrate this idea, we will merely describe the basic principle of the recent *AutoClass* algorithm, but many other methods have been proposed within this framework; for further information the interested reader may refer to [DU 73, HA 81, DE 82].

AutoClass [CH 88a]

AutoClass is based on a Bayesian approach to clustering, proposed by Cheeseman [CH 88a]. Its main advantages are its ability to determine automatically the most likely number of clusters and to handle both numerical and symbolic attributes, including hierarchically structured ones. The main assumption of AutoClass is that the attributes values are independent in a given cluster. Thus each cluster is described by a product of elementary attribute distributions. Real-valued attributes are modelled by unidimensional Gaussian distributions and discrete attributes by probability tables.

The AutoClass approach is based on the Bayesian theory of finite mixtures. Each learning state is assumed to be drawn from one of K mutually exclusive and exhaustive classes, described by a probability distribution as indicated above :

$$p(\mathbf{a}|M) = \sum_{j=1,K} p_j * p(\mathbf{a}|c_j, M_j), \quad (4.43)$$

where M denotes the model hypothesis which is composed of the vector of class probabilities p_j and one set of model parameters M_j for each class c_j .

For a given choice of the model parameters M , each observation \mathbf{a} will have a probability of belonging to each class computed by

$$p(c(o) = c_i | \mathbf{a}(o), M) = \frac{p_i * p(\mathbf{a}(o) | c(o) = c_i, M_i)}{p(\mathbf{a}(o) | M)}. \quad (4.44)$$

To learn the model parameters and its order K , the joint probability density of the LS under the model assumption and independence hypothesis is computed

$$p(LS | M) = \prod_{o \in LS} p(\mathbf{a}(o) | M). \quad (4.45)$$

From this, the posterior distribution of the model parameters may be computed, under the hypothesis of known order K by

$$p(M | LS, K) = \frac{p(M | K) p(LS | M, K)}{p(LS | K)}, \quad (4.46)$$

where $p(LS | K)$ is the normalizing constant obtained by

$$p(LS | K) = \int_{\mathcal{M}_K} p(M_K | K) p(LS | M_K, K) dM_K, \quad (4.47)$$

where M_K denotes the parameter choice for a model of order K , and \mathcal{M}_K the space of possible such models.

The posterior distribution of the number of classes is then obtained by

$$p(K | LS) = \frac{p(K) p(LS | K)}{p(LS)}. \quad (4.48)$$

The optimal order is the one maximizing the above probability.

It is important to notice that there are two prior distributions, $p(K)$ and $p(M_K | K)$, which must be filled in the above reasoning in order to define the algorithm.

In particular, we may for example assume that the prior distributions of the model complexity are uniform and that the model parameters are conditionally distributed uniformly, i.e. $p(M_K | K)$ is uniform in an a priori defined parameter interval. In this case, the prior probability of a particular choice of parameters $p(M, K)$ will automatically decrease when the number of parameters increases. And this decrease in prior probability will trade off the increased model fit $p(LS | M, K)$ in eqn. (4.46) and prevent overfitting. Of course, the algorithm may also take into account the user's prior beliefs about model complexity and parameters.

Thus, the apparently inconsequent hypothesis of a *conditional* uniform prior model probability given its complexity, leads to the cost complexity tradeoff. This should be compared with the maximum likelihood strategy, which is equivalent to assuming a priori that all models are equally likely, *independently* of their complexity.

As we discuss in [WE 94b], a very similar reasoning leads to the Bayesian justification of the tree quality measure explained in §3.4. This gives an “a posteriori” explanation of our choice of describing the AutoClass method.

4.5 DATA PREPROCESSING

Before reaching the conclusion of this chapter, it is our duty to provide some hints about a certain number of classical data preprocessing techniques, which are often used to transform an initial representation into a set of more appropriate attributes, and which belong to the established statistical pattern recognition auxiliary tools.

These techniques provide an intermediate tool between the manual choice of an ad hoc representation which is more of an “art”, and the fully integrated automatic learning methods such as the machine learning and neural network methods.

4.5.1 Pre-whitening

Pre-whitening or normalization of data consists of linearly transforming each attribute, to obtain a zero mean and unit variance

$$a' = \frac{a - \bar{a}}{\sqrt{(a - \bar{a})^2}}. \quad (4.49)$$

This is the least one can do in order to render numerical techniques, such as nearest neighbor computations and clustering, independent of arbitrary attribute scalings.

4.5.2 Feature selection

Feature selection consists of reducing the dimensionality of the input data by selecting a small number of the most discriminant or the most representative features. There may be two motivations for reducing the dimension of the input space. The first one is purely related to computational efficiency of the subsequent learning tasks. The second reason is more related to the problem of overfitting.

Although there is a whole bunch of complicated feature selection algorithms described in the literature [DE 82], we will only describe some basic, very simple techniques which could allow us to remove the redundant information, since many of the modern techniques for classification or regression have some built in feature selection or extraction capabilities.

Attribute clustering

As we have suggested above, the clustering analysis of attributes allows us to identify groups of attributes which are very strongly correlated, i.e. which share the same physical information. In the context of power system security problems this is very frequent for variables such as power flows (see Fig. 4.8) or voltages (see §14.4). A simple dendrogram may be drawn to suggest which groups of such variables may be represented by a single prototype, e.g. a mean value. In practice, this may lead to a more efficient and more robust classification.

But since the attribute clustering technique does not take into account the classification information, it is not very selective in identifying the discriminant attributes.

Decision tree building

The next step for feature selection could be to build a decision tree, on the basis of the available pre-classified data or a regression tree, as appropriate. The detailed information on the scores obtained by each candidate attribute and their estimated standard deviation and correlation, make it in general quite easy to determine a much smaller subset of the most discriminant variables.

This technique has been used above in the context of the voltage security example of Table 4.2, in the discussion of the nearest neighbor rule in §4.3.1. This method was found to provide an important reduction in dimensionality in several other power system security applications.

Simple sequential feature selection

One method of feature extraction, which has often been used for its great simplicity consists of selecting the features sequentially according to the following figure of merit

$$J \triangleq \frac{|S_B(a_1, \dots, a_k)|}{|S_W(a_1, \dots, a_k)|}, \quad (4.50)$$

where $|S_B(a_1, \dots, a_k)|$ schematically represents a between class scatter index and $|S_W(a_1, \dots, a_k)|$ stands for a mean within class scatter index. Both are supposed to be computed in the attribute sub-space (a_1, \dots, a_k) .

The above figure of merit can then be determined for each single attribute to choose the first attribute a_1^* , and then for each pair of attributes a_1^*, a_2 , to determine the most complementary attribute a_2^* , and so on This is the sequential forward selection approach. Another, dual scheme, consists of starting with the complete list of candidate attributes and deleting at each step the least useful one, i.e. leading to the highest value of the performance index for the remaining set of attributes.

A simplification of the above scheme consists of computing the index in a scalar attribute by attribute approach. E.g. assuming attribute independence and restriction to the two-class case, an index may be computed for each attribute in terms of the ratio

$$J_a \triangleq \frac{|\mu_{a_1} - \mu_{a_2}|^2}{p_1 \sigma_{a_1}^2 + p_2 \sigma_{a_2}^2}, \quad (4.51)$$

of the square difference of the class-conditional mean values to the weighted sum of the class-conditional standard deviations. Excluding strongly correlated attributes, one may select the n' best attributes according to the above criterion.

4.5.3 Feature extraction

While the feature selection methods search for an optimal *subset* of the initial attributes, the feature extraction methods aim at defining a set of - generally linear - feature combinations.

We will merely indicate the basics of the *Karhunen-Loève* expansion of *principal components analysis*. The objective of this technique is to linearly transform the initial attributes in order to concentrate the maximum of information in a minimum number of transformed attributes.

In the following we suppose that each attribute has been centered by subtracting its mean value. Moreover, we will use expectation operators to manipulate population quantities. The same derivations may then be applied to the finite sample case by replacing expectation operators by sample mean values.

Thus we assume that

$$E\{\mathbf{a}\} = \mathbf{0}. \quad (4.52)$$

Then, let us consider an orthonormal system of vectors $\mathbf{u}_1, \dots, \mathbf{u}_n$, i.e. such that

$$\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}, \quad (4.53)$$

and express the attribute vectors as a linear combination of these vectors

$$\mathbf{a} \triangleq \sum_{i=1,n} \tilde{a}_i \mathbf{u}_i, \quad (4.54)$$

where

$$\tilde{a}_i = \mathbf{u}_i^T \mathbf{a}, \quad (4.55)$$

since the vectors \mathbf{u}_i form an orthonormal basis.

If we take a smaller number $d < n$ of terms, truncating the above series, we obtain an approximate representation of the vector \mathbf{a} ,

$$\hat{\mathbf{a}} = \sum_{i=1,d} \tilde{a}_i \mathbf{u}_i. \quad (4.56)$$

We define our “optimal representation problem” as the choice of the vectors \mathbf{u}_i , $i = 1, d$ which minimize the following mean squared representation error

$$\epsilon = E \left\{ (\mathbf{a} - \hat{\mathbf{a}})^T (\mathbf{a} - \hat{\mathbf{a}}) \right\}. \quad (4.57)$$

In other words, we search for a d -dimensional subspace of the initial attribute space, which is closest to the probability distribution in the Euclidean distance sense.

Substituting the expressions (4.54) of \mathbf{a} and (4.56) of $\hat{\mathbf{a}}$ into eqn. (4.57), we obtain

$$\epsilon = E \left\{ \sum_{i=d+1,n} \tilde{a}_i^2 \right\}, \quad (4.58)$$

where we have exploited the orthonormality conditions (4.53).

Thus, using the expression (4.55) of the expansion coefficients, we obtain

$$\epsilon = E \left\{ \sum_{i=d+1,n} \mathbf{u}_i^T \mathbf{a} \mathbf{a}^T \mathbf{u}_i \right\}, \quad (4.59)$$

or

$$\epsilon = \sum_{i=d+1,n} \mathbf{u}_i^T \left[E \left\{ \mathbf{a} \mathbf{a}^T \right\} \right] \mathbf{u}_i, \quad (4.60)$$

exploiting the fact that the vectors \mathbf{u}_i are independent of \mathbf{a} , to interchange the summation and the expectation operators. Notice that the matrix $E \left\{ \mathbf{a} \mathbf{a}^T \right\}$ is the covariance matrix (cf. assumption (4.52)).

It can be shown that the stationary points of the above expression correspond to choosing for \mathbf{u}_i eigenvectors of the covariance matrix $E \left\{ \mathbf{a} \mathbf{a}^T \right\}$, i.e. such that

$$E \left\{ \mathbf{a} \mathbf{a}^T \right\} \mathbf{u}_i = \lambda_i \mathbf{u}_i. \quad (4.61)$$

Under this condition the mean square representation error ϵ is computed by

$$\epsilon = \sum_{i=d+1,n} \lambda_i, \quad (4.62)$$

and will be minimal if the λ_i 's are chosen as the $n - d$ smallest eigenvalues of the covariance matrix.

Reciprocally, the optimal truncation is obtained by using the eigenvectors corresponding to the d largest eigenvalues. In principle, the truncation error would also be minimized by using any orthonormal basis of the subspace spanned by the eigenvectors corresponding to the d largest eigenvalues.

However, choosing the eigenvectors rather than an arbitrary orthonormal combination of them, has the additional feature of decorrelating the transformed attribute values. Indeed, it is easy to show that for this particular choice of \mathbf{u} vectors $E \left\{ \tilde{a}_i \tilde{a}_j \right\} = \lambda_i \delta_{ij}$.

4.6 CONCLUDING REMARKS

In this chapter we have aimed at providing an overview of the classical and also the more recent statistical techniques, able to provide some interesting tools in the context of our power system security assessment problems. Our main objective was to give an intuitive understanding of the principles.

The second objective was to suggest possible practical uses; it led us to support our description with several illustrative results from real power system problems. Although doing so has introduced the additional difficulty of explaining, up to a certain degree, the considered power system problems, anticipating thereby later chapters, we hope that this has been useful in supporting our message.

This message might be summarized in three sentences.

Parametric approaches, though too simple to be stand-alone methods, may provide very useful auxiliary tools, for example to define interesting attribute combinations or to provide quickly a simple first order model.

Non-parametric approaches, in particular the projection pursuit techniques drawing their inspiration from neural networks, often yield less transparent black-box models, but they may be very powerful in terms of modelling capabilities.

Finally, a very important aspect in applying either of these techniques, is the proper exploration and analysis of the data, using the various parametric, non-parametric, supervised and non-supervised approaches as tools.

To conclude, we notice that an important part of the work of applying statistical, neural network or machine learning methods to power systems security, consists of analyzing the data base contents in order to check representativity assumptions so as to validate the resulting criteria. In this context, graphical representations, such as scatter plots, one or two-dimensional histograms or dendrograms may provide very useful tools, and one of our objectives has been to provide some practical examples of such graphical information representations.

5

Artificial neural networks

5.1 INTRODUCTION

While the learning systems based on artificial neural networks became popular only recently, they have already a very long research history and some have evolved towards quite mature techniques. Considering the early work on neural networks, it is interesting to observe the analogy with the first machine learning research. Both were mainly motivated by the study and modelling of the human learning ability. However, while the machine learning research was mainly aimed at providing a phenomenological simulation model of the high-level capacities of the brain, the neural network approach aimed at reproducing these latter capabilities, starting from a low-level model reflecting the structure of the brain, in a bottom up fashion.

The emergence of artificial neural network models dates back to the 1940's, with the work by McCulloch and Pitts [MC 43] on modelling the brain neuron behavior. The second wave of the research reached its peak in the early sixties with the perceptron learning theorem of Rosenblatt [RO 63] and the negative results concerning the perceptron's representation capability limitations of Minsky and Papert [MI 69]. Finally, the last wave has started from the conjunction of the rapid increase in available computing power in the early 1980's, the theoretical work of Hopfield, and the improvements of multi-layer perceptrons culminating with the (re)publication of the back-propagation algorithm by Rumelhart, Hinton and Williams [RU 86].

Since the mid 1980's, an almost exponentially growing amount of theoretical and practical work has been published, leading to the creation of new journals and conferences, and several textbooks. Even if we were restricting our focus to the field of power system applications, it would still be very difficult to give a reasonably representative account of the ongoing research. Thus, although there are many other potentially interesting power system problems for neural network applications, such as adaptive control and

load forecasting to mention only the most popular ones, we will restrict our attention to security assessment applications, and discuss some of the most promising techniques in this context only.

In the first part of this chapter we describe in some detail the single and multi-layer perceptrons which are representative of the family of feed-forward neural network architectures for supervised learning. In §5.2.6 we will mention the functional link network, which is another feed-forward structure which has been applied in security assessment problems [PA 89b]. These correspond probably to the most well known and mature neural network techniques, which have shown some true potential in the context of real large scale power system problems [WE 93a].

The second part of our description is devoted to the non-supervised neural network approach of Kohonen, which we consider also as an attractive technique for data analysis and graphical interpretations. However, while several research projects are progressing in this context, these applications have not yet reached the maturity of multi-layer perceptrons.

As we have already mentioned, a lot of interesting research is currently going on in applying computational learning theory, as well as Bayesian and classical statistical frameworks to the neural network paradigm. The aim of the latter work is to provide theoretical foundations and unifications among the neural, statistical and machine learning frameworks. But, it is still too early to assess the practical outcomes of this work in terms of improved learning algorithms and/or more effective architectures. Thus, as we have done in the preceding chapters we will merely point out for the interested reader the reference book by Hertz, Krogh and Palmer which considers the stabilized part of neural network theory [HE 91].

We will provide some practical illustrations on the basis of our standard example of power system transient stability assessment of §3.4.

5.2 MULTI-LAYER PERCEPTRONS

To give a historical perspective of the work on perceptrons, we will start by describing the single layer perceptron or linear threshold unit (LTU) and its learning algorithm. Then we will consider the use of soft threshold units and the gradient descent mean squared error (MSE) correction algorithm, which is the parent of the well known back-propagation algorithm.

Further, we will proceed with multiple layer feed-forward network structures, and general neuron activation functions, and after describing briefly the basic stochastic gradient descent method, we will give a short description of more efficient batch oriented second order optimization techniques, and conclude with some remarks concerning the

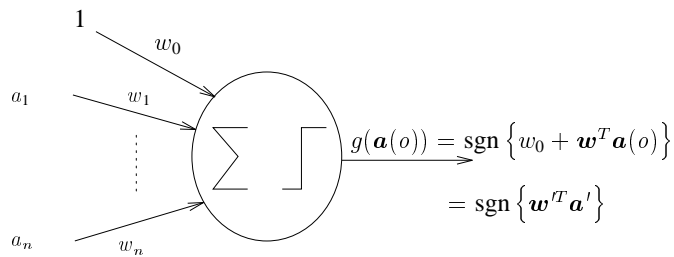


Figure 5.1 Basic linear threshold unit

choice of network architectures.

5.2.1 Single layer perceptron

Note. For convenience and coherence with the other preceding and following descriptions, we use the $-1/+1$ binary encoding rather than the often used $0/1$ encoding. Further, we will also use the extended attribute vector notation

$$\mathbf{a}' \triangleq \begin{pmatrix} 1 \\ \mathbf{a} \end{pmatrix}. \quad (5.1)$$

Linear threshold units

The single layer perceptron, or simply perceptron, is a hyperplane model similar to the linear discriminants of §4.2.1. It is implemented by the linear threshold unit represented in Fig. 5.1, which assigns a value of -1 or $+1$ depending on which side of the hyperplane corresponding to its weight vector the attribute vector is located, and may be used in order to solve a two-class classification problem. In particular, in the case of boolean functions, the attribute values are themselves binary $-1/+1$ indicators.

Supposing that the learning set classification information has been encoded in the above binary fashion, i.e. $c(o) \in \{-1, 1\}$, the ideal objective of the perceptron learning algorithm is to reproduce the learning set classification perfectly, or equivalently to choose a set of weights $\mathbf{w}'^T = (w_0, \mathbf{w}^T)$ such that

$$\sum_{o \in LS} [g(\mathbf{a}(o)) - c(o)]^2 = 0. \quad (5.2)$$

Learning problems for which this is possible are called linearly separable.

The perceptron learning algorithm is indicated in Table 5.1; it is a sequential method considering successive passes through the learning states, and adjusting the weights at

Table 5.1 *Perceptron learning algorithm*

-
1. Consider the objects of the learning set in a cyclic or random sequence.
 2. Let o be the current object, $c(o)$ its class and $\mathbf{a}(o)$ its attribute vector.
 3. Adjust the weight by using the following correction rule,

$$\mathbf{w}^{new} = \mathbf{w}^{old} + \eta (c(o) - g(\mathbf{a}(o))) \mathbf{a}'(o). \quad (5.3)$$

each step so as to improve the classification. Notice that the corrections are equal to zero for objects which are already classified correctly; for incorrectly classified objects the correction of the weight vector is parallel to the object's attribute vector, and the direction is chosen so as to bring the output closer to the correct output value $c(o)$.

The parameter η denotes the learning *rate* of the algorithm, and various strategies have been proposed to choose its value. It may be shown that if the learning set is separable, then the fixed learning rate perceptron learning rule converges to a solution in a finite number of steps, but the speed of convergence may depend on the values of η . In addition, if the learning set is not separable, then the algorithm will never stop changing the weight values. Thus, one of the techniques used to ensure convergence consists of using a decreasing sequence of learning rate values $\eta_k \rightarrow 0$.

The structure of the single LTU may be generalized to a single layer of LTUs, allowing them to learn a boolean vector or binary coded integer output function, as would for example be required for multi-class classification problems.

It was a great scientific deception at the time when Minsky and Papert published their work on the representation capability limitations of the LTU. In particular, it is a well known result that the perceptron is unable to represent an as simple function as the two-dimensional logical XOR (exclusive OR) operator, or the general n -dimensional parity function. It was noted quite early that the solution to this problem calls for more complex, multi-layer structures. Unfortunately, the discrete perceptron learning rule does not generalize to multi-layer structures.

The solution to this problem calls for multi-layer structures with non-linear but differentiable input/output relations, to allow the use of the error back-propagation learning algorithm. Therefore, we will first consider the *soft* threshold units which provide the elementary brick to build up such general powerful multi-layer models.

Soft threshold units and minimum mean squared error learning

The soft threshold unit is a slight modification of the perceptron, which considers a nonlinear differentiable activation function applied to a linear combination of input attributes, instead of a hard threshold.

The input/output function $g(\mathbf{a})$ of such a device is computed by

$$g(\mathbf{a}) \triangleq f(w_0 + \mathbf{w}^T \mathbf{a}) = f(\mathbf{w}'^T \mathbf{a}') \quad (5.4)$$

where the *activation* function $f(\cdot)$ is assumed to be differentiable. Classical examples of activation functions are the sigmoid and hyperbolic tangent functions, but other types of general non-linear smooth functions may also be considered.

Considering output values varying continuously between -1 and 1, and the possibility of non-separable problems, we now reformulate the learning objective as the definition of a weight vector $\mathbf{w}' = (w_0, \mathbf{w})$ minimizing the mean squared error (MSE) criterion

$$MSE(\mathbf{w}') \triangleq \sum_{o \in LS} [g(\mathbf{a}(o)) - c(o)]^2. \quad (5.5)$$

The gradient of the MSE with respect to the augmented weight vector \mathbf{w}' is computed by

$$\nabla_{\mathbf{w}'} MSE = 2 \sum_{o \in LS} [g(\mathbf{a}(o)) - c(o)] f'(\mathbf{w}'^T \mathbf{a}'(o)) \mathbf{a}'(o). \quad (5.6)$$

Thus, using a *fixed* step gradient descent approach for minimizing the mean squared error, in a sequential object by object correction setting, would consist of using the following weight update rule

$$\mathbf{w}'^{new} = \mathbf{w}'^{old} - \eta \nabla_{\mathbf{w}'} MSE \quad (5.7)$$

$$= \mathbf{w}'^{old} + \eta [c(o) - g(\mathbf{a}(o))] f'(\mathbf{w}'^T \mathbf{a}'(o)) \mathbf{a}'(o). \quad (5.8)$$

This is analog to the perceptron learning rule of Table 5.1, where the learning rate η is adapted proportionally to the derivative f' of the activation function.

An alternative, *batch* learning approach consists of computing the full gradient (5.6) of the MSE with respect to the complete learning set before correcting the weight vector.

A further improvement would then consist of using a *variable* step gradient descent method, for example the *steepest descent* approach. This consists of using a line search so as to determine at each stage the step in the gradient direction resulting in a maximal decrease of the MSE criterion. Other more sophisticated numerical optimization techniques may be thought of, and are discussed below.

However, one of the remaining problems concerns the existence of local minima of the MSE criterion, to which the gradient type search techniques will converge. A possible

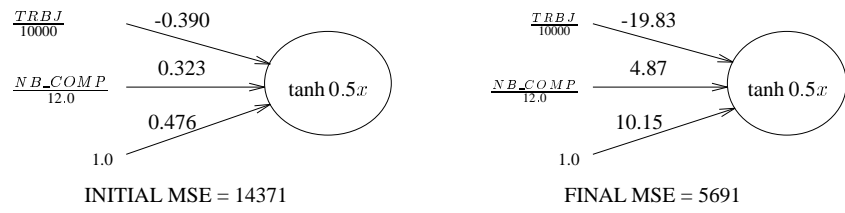


Figure 5.2 Soft threshold unit for the linear combination of *TRBJ* and *NB_COMP*

solution consists of repeating the search procedure for various randomized initial weight vectors; another suggestion has been to apply heuristic global optimization techniques such as the simulated annealing method or the genetic algorithms discussed in §3.5.3.

Another problem concerns the minimum MSE criterion itself, which does not necessarily lead to a minimum number of misclassification errors, neither in the learning set, nor - a fortiori - in an independent test set.

Illustration

To illustrate the above minimum MSE method, let us turn back to our example of the power system transient stability assessment problem of §3.4.

We consider again the two-dimensional attribute space (*TRBJ*, *NB_COMP*) and search for an optimal hyperplane. As is suggested in Figs. 3.12 and 4.1, the Stable and Unstable classes are far from being linearly separable in this attribute space. Thus the basic perceptron learning algorithm would probably not be appropriate and we propose to use the minimum MSE criterion, together with a soft threshold unit, using an hyperbolic tangent activation function

$$f(x) = \tanh(\beta x) = \frac{\exp\{\beta x\} - \exp\{-\beta x\}}{\exp\{\beta x\} + \exp\{-\beta x\}}. \quad (5.9)$$

The input attribute values have been normalized by dividing them by their maximum value ($\max(\textit{TRBJ}) = 10000$ and $\max(\textit{NB_COMP}) = 12$). The weight vectors were initialized to random values chosen in the interval $[-0.5 \dots 0.5]$.

The parameter β - although redundant - provides a convenient way to control the initial working range of the activation function. We have used a rather low value of $\beta = 0.5$, so as to start in the linear part of the activation function. Using a very high value of β on the other hand would result in saturating the activation function and slow down, or even prevent the convergence to a good solution. The same learning set as previously, composed of all the 12497 states of the data base, has been used.

In order to minimize the MSE, we have used the batch *steepest descent* algorithm. The iterative process is stopped as soon as a local minimum is detected or when a certain

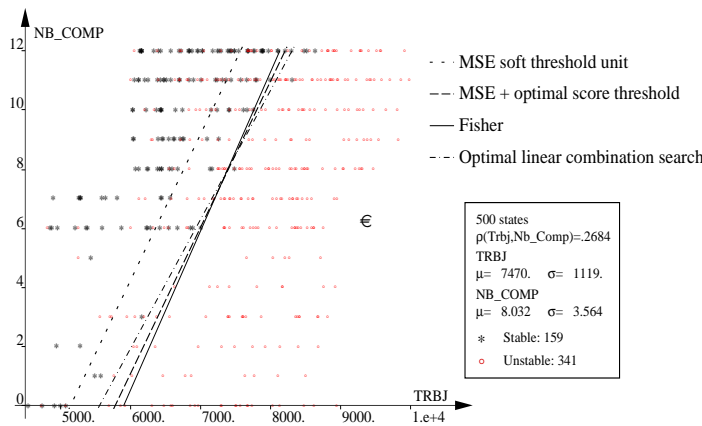


Figure 5.3 Comparison of various linear combinations

maximal number of iterations has been reached. The main advantage of this procedure is that it does automatically adjust the learning rate to the problem specifics and thus no manual tuning is required.

After 858 iterations, corresponding to a total CPU time of 12850 seconds, the above procedure converged to a local minimum of the MSE. The corresponding initial and final weight values are shown in Fig. 5.2. One can see that the MSE has been reduced from the initial value of 14371, corresponding to a random classification of the learning states, to a value of 5691, corresponding to a weight vector

$$w_0 = 10.15; w_{TRBJ} = -19.83; w_{NB_COMP} = 4.87.$$

Taking into account the attributes normalization, this corresponds to the linear combination partition

$$TRBJ - 205 * NB_COMP < 5120MW,$$

which is depicted in Fig. 5.3 along with the previously found optimal score linear combination of Fig. 3.12 and Fisher’s linear discriminant of Fig. 4.1.

The score of this test, obtained by formula 3.19, is equal to 0.3046 which is significantly smaller than the optimal score of 0.3646 and the score of 0.3440 of Fisher’s linear discriminant. This is essentially due to the fact that the minimum MSE criterion and the score criterion correspond to a different compromise. For example, using the above linear combination $TRBJ - 205 * NB_COMP$ and the optimal threshold search so as to maximize the score measure, yields a threshold of 5753 MW, and a nearly optimal score of 0.3558. This test is also shown in Fig. 5.3.

This simple but real example allows us to make some interesting practical observations.

First of all, the iterative gradient descent procedures are very slow and in practice often

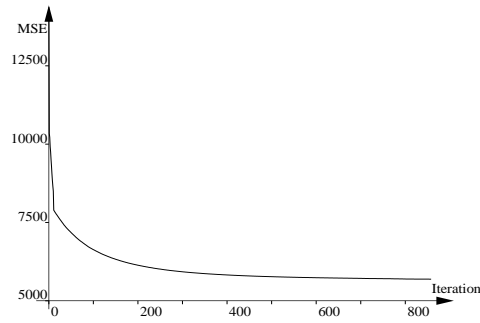


Figure 5.4 Variation of MSE during steepest descent iterations

require a large number of iterations to converge.

Thus, even in the case of a very simple low dimensional problem, with a small number of parameters, the computing time required to determine optimal weight values may become prohibitive, as soon as the number of learning states becomes important. This is illustrated in Fig. 5.4 showing the slow convergence of the MSE, during the 858 steepest descent iterations.

Below, in the context of the multi-layer perceptrons, we will consider more efficient numerical techniques allowing us to significantly speed up the gradient descent search. Anticipating this description, we have used the so called “Broyden-Fletcher-Goldfarb-Shannon” quasi-Newton approach to minimize the above MSE. This yielded a dramatic improvement in computational efficiency, since the convergence was obtained within 13 iterations and a CPU time of 323 seconds. The solution was even slightly improved (MSE = 5685) and corresponds to the weights

$$w_0 = 10.92; w_{TRBJ} = -21.25; w_{NB_COMP} = 5.18,$$

corresponding to the test

$$TRBJ - 203 * NB_COMP < 5138MW.$$

In spite of this important improvement, we will see that the multi-layer perceptron learning process is in practice several orders of magnitude slower than the TDIDT procedure, while providing most often only a rather small, if any, improvement in error rate.

The second observation concerns the effect of using different optimality criteria which may lead in practice to different results. We have already illustrated this several times. Table 5.2 summarizes the set of results obtained in the context of this simple two-dimensional example, using various techniques and criteria to determine the linear combination and optimal threshold, and assessed on the basis of two different optimality

Table 5.2 *Effect of criteria and algorithms on CPU time and quality assessment*

Learning criterion		Algorithm	Nb. Iter.	CPU sec.	Test*		Evaluation	
Direction	Threshold				λ	λ'	Score	$R^{LS}\%$
mx $R^{LS}\%$	mx $R^{LS}\%$	Optimal search	20	110	179	5469	0.3263	83.88
mx score	mx score	Optimal search	20	110	227	5560	0.3646	79.76
Fisher	Fisher	Direct	1	12	186	5465	0.3291	83.63
Fisher	mx score	Direct + search	2	20	186	5903	0.3440	79.07
MSE	MSE	Steepest Desc.	858	12850	205	5120	0.3046	83.54
-	-	BFGS	13	323	203	5138	0.3066	83.61
-	mx score	St. D. + search	859	12857	205	5753	0.3558	79.28

* $TRBJ - \lambda * NB_COMP < \lambda'$

criteria : the score according to eqn. (3.19) and the reliability R^{LS} in terms of the percentage of correctly classified states among the 12497 learning states.

For each trial, Table 5.2 indicates in addition to the criteria and algorithms used to determine the optimal linear combination direction and threshold, the number of iterations, the amount of CPU time required, the values of the linear combination parameters, the corresponding score and percentage of correct classifications obtained by using the corresponding test to predict the stability of the learning states.

The advantage of the optimal linear combination search is its flexibility with respect to the type of optimality criterion, and its good efficiency. However, the generalization of the nested optimization loop would hardly be feasible for more than, say, three dimensions.

From the CPU time point of view, the most attractive technique is the direct computation of Fisher's linear discriminant, which however does not in general provide an optimal linear combination with respect to either evaluation function. In the present case, however, the pure Fisher's discriminant is very close to optimal in terms of the reliability estimate R^{LS} .

On the other hand, using the perceptron like gradient descent technique is interesting due to the generality of the method, both in terms of numbers of attributes used in the linear combination and in terms of the actual neuron activation function and error criterion used, as we will see. This will be discussed more in detail below, after we have introduced the more general multi-layer architectures.

5.2.2 Multiple layer feed-forward networks

As we have mentioned above, the need for more complex multiple layer structures of networks was felt as soon as the limitations of the perceptron were established. The

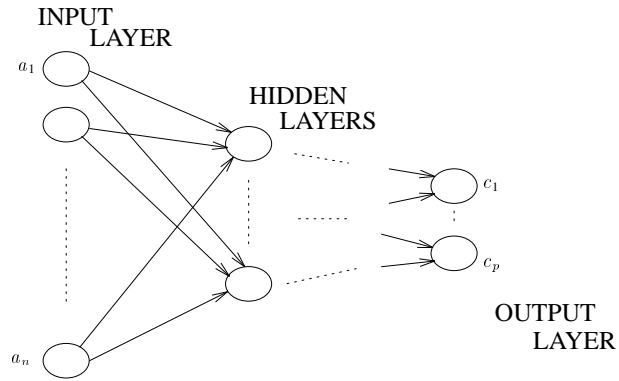


Figure 5.5 *Feed-forward multi-layer perceptron*

usual multi-layer perceptron structure is represented in Fig. 5.5. In addition to the input and output layers, a number of - generally at most 2 - hidden layers allow us to provide arbitrary complex function mapping capabilities.

Because of its historical importance we will describe in some detail the basic back-propagation algorithm, although in practice we did not actually use this method in our simulations, due to its low computational efficiency. However, the back-propagation algorithm uses the basic chain-rule algorithm for the gradient computation, which is also the heart of the more sophisticated techniques discussed in §5.2.4.

Back-propagation algorithm

The basic idea of the algorithm is to compute the derivatives of the error function in a layer by layer fashion, starting with weights feeding the output layer and ending with the weights feeding the first hidden layer of neurons.

Let us consider the general feed-forward structure suggested in Fig. 5.6, where the neurons are sequentially ordered from 1 to K . In this structure a neuron j receives a net input n_j

$$n_j(o) \triangleq \sum_{i=1, j-1} w_{i,j} x_i(o), \quad (5.10)$$

where $w_{i,j}$ denotes the weight of the connection from neuron i to neuron j , and $x_i(o)$ the activation (or output) of neuron i , for object o . Further, each neuron has a differentiable activation (or transfer) function $f_j(\cdot)$ and its output state x_j is computed by

$$x_j(o) \triangleq f_j(n_j(o)). \quad (5.11)$$

Although the classical multi-layer perceptron is a particular case of this structure, where some of the weights are constrained to be equal to zero, it is simpler to explain the

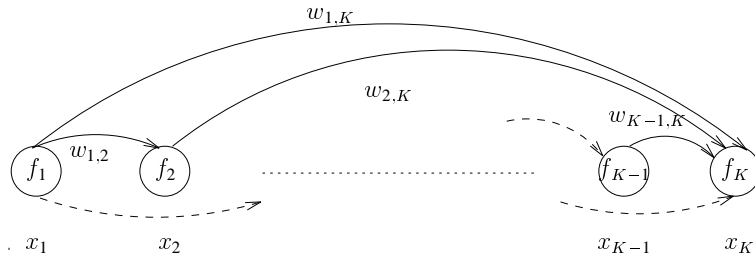


Figure 5.6 General feed-forward network

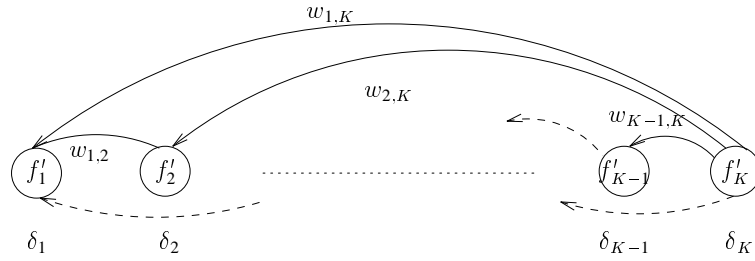


Figure 5.7 Back-propagation of errors

back-propagation algorithm on the basis of the above general, fully connected structure.

Further, we may consider the following general type of “error” function

$$ERR(w_{i,j}, LS) \triangleq \sum_{o \in LS} h(\mathbf{x}(o), \mathbf{y}(o)), \tag{5.12}$$

where $h(\cdot, \cdot)$ denotes a differentiable function of the neuron activation vector $\mathbf{x} = (x_1, \dots, x_K)$ and of the desired output vector $\mathbf{y} = (y_1, \dots, y_r)$.

The derivatives of the error function with respect to the network weights $w_{i,j}$ are then computed by the following formula

$$\frac{\partial ERR(w_{i,j}, LS)}{\partial w_{i,j}} = \sum_{o \in LS} \sum_{k=1, K} \frac{\partial h(\mathbf{x}(o), \mathbf{y}(o))}{\partial x_k} \frac{\partial x_k}{\partial w_{i,j}}. \tag{5.13}$$

On the other hand, the essence of the back-propagation algorithm which is suggested graphically in Fig. 5.7, consists of computing the partial derivatives

$$\frac{\partial x_k}{\partial w_{i,j}}, \tag{5.14}$$

by propagating them back from the high order to the low order neurons.

More precisely, these derivatives are obtained by the following backward recursion relations

$$\frac{\partial x_k}{\partial w_{i,j}} = x_i \delta_j, \quad (5.15)$$

where

$$\delta_j = 0, \quad \forall j > k; \quad (5.16)$$

$$\delta_j = f'_j, \quad \forall j = k; \quad (5.17)$$

$$\delta_j = f'_j \sum_{p=j+1,k} w_{j,p} \delta_p, \quad \forall j < k. \quad (5.18)$$

This is quite easy to prove.

First of all, the relations 5.16 express simply the fact that the network has a feed-forward structure, which implies that the state of neuron k is independent of the weights of connections to neurons of higher order.

Second, eqn. 5.17 is obtained by direct differentiation of 5.10 and 5.11.

And finally, the recursion relation (5.18) is obtained by applying the chain rule of differentiation in the following way, as suggested in Fig. 5.8

$$\frac{\partial x_k}{\partial w_{i,j}} = \frac{\partial x_k}{\partial x_j} \frac{\partial x_j}{\partial w_{i,j}}, \quad (5.19)$$

We can substitute in this equation the following identity

$$\frac{\partial x_j}{\partial w_{i,j}} = x_i f'_j, \quad (5.20)$$

which follows directly from the base case eqn. (5.17). Making explicit the dependence of x_k on x_j and $w_{j,p} \quad \forall p = j+1, \dots, k$, we note that x_k may be written as a function $g(w_{j,j+1}x_j, w_{j,j+2}x_j, \dots, w_{j,k}x_j)$. Thus it is clear that eqn. (5.15) applied to $w_{j,p}$,

$$\frac{\partial x_k}{\partial w_{j,p}} = x_j \delta_p, \quad \forall p = j+1, \dots, k, \quad (5.21)$$

which is supposed to hold by induction hypothesis, implies also that

$$\frac{\partial x_k}{\partial x_j} = \sum_{p=j+1,k} w_{j,p} \delta_p, \quad \forall j < k, \quad (5.22)$$

Q.E.D. \square

In a classical *multi-layer* feed-forward network, the first $n+1$ neurons would correspond to the extended input attribute vector \mathbf{a}' . Thus, their activation would be fixed, for a given object o presented to the network, independently of any weight values by

$$x_j(o) = a_j(o), \quad \forall j = 1, n, \quad \text{and} \quad w_{i,j} = 0, \quad \forall i < j, \quad (5.23)$$

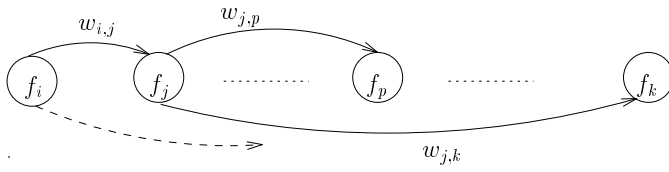


Figure 5.8 Explanation of the chain rule differentiation

and

$$x_{n+1}(o) = 1 \text{ and } w_{i,n+1} = 0, \forall o. \tag{5.24}$$

On the other hand, the last r output values would correspond to the output information of the network,

$$x_{k-r+j} = r_j, \forall j = 1, \dots, r \text{ and } w_{j,i} = 0, \forall i > j. \tag{5.25}$$

Finally, the layers are defined as groups of consecutive neurons receiving information from neurons in the preceding layer and feeding their information to neurons in subsequent layer. Denoting by L_l a layer number l , and L the total number of layers

$$L_l = \{j, j + 1, \dots, j + n_L - 1\}, l = 1, \dots, L \tag{5.26}$$

this corresponds to a set of connectivity constraints

$$w_{i,j} = 0, \forall i, j | i \in L_l, j \notin L_{l+1} \tag{5.27}$$

In the case of the multi-layer structure, the error function would explicitly take into account only neuron activations of the last (output) layer. For example, the standard MSE error function is defined by

$$h(\mathbf{x}(o), \mathbf{y}(o)) = \sum_{i=1,r} |\mathbf{x}_{k-r+i}(o) - y_i(o)|^2. \tag{5.28}$$

The overall derivative of the error function is then obtained by sweeping through the learning set, and computing for each object the activation vector $\mathbf{x}(o)$ in the feed-forward fashion, and using the back-propagation algorithm to compute the derivatives with respect to each weight in a backward fashion, cumulating the terms corresponding to the components of the activation vector which are explicitly used in the error function, proportionally to the corresponding partial derivative $\frac{\partial h}{\partial x_i}$. All the computations being linear, this may be done in a single pass for all output neurons, and for a given activation corresponding to a given object.

While the above recursion is a simple chain rule derivation, the interesting point is that the corresponding error back-propagation algorithm is local and uses the same network structure as the original feed-forward network. Another notable fact is the surprising computational efficiency of this algorithm, since *all* the derivatives are obtained with

the order of w operations, where w is the total number of weights of the network, which is also the computational complexity the network output function computation.

Finally, the method may be used either in an incremental learning scheme, adapting the weights after each presentation of an input attribute vector, or in the batch approach cumulating derivatives over the full learning set before adapting the weights.

5.2.3 Other objective functions

In our presentation of the back-propagation algorithm, we have insisted on its generality, showing that it is able to handle any kind of feed-forward network structures and may be adapted to general activation and objective functions. Below, in §5.2.4 we will discuss some alternative schemes for exploiting these derivatives in order to optimize the objective function in an efficient way, while in §5.2.5 we will briefly comment on some usual approaches of defining the network architecture in terms of its topology and activation functions. In this section, we comment on some frequently used network optimization criteria.

Regularization

Most of the objective functions which have been used in practice derive directly from the standard minimum MSE criterion. They take the following general form

$$MSE(w_{i,j}, LS) = \sum_{o \in LS} \sum_{i=1,r} |\mathbf{x}_{k-r+i}(o) - y_i(o)|^2 + G(\|w_{i,j}\|^2), \quad (5.29)$$

where $G(\|w_{i,j}\|^2)$ denotes a generic “regularization” term, which aims at accounting for the “smoothness” of the input/output mapping. The purpose of the regularization term is to avoid high frequency components in the input/output mapping so as to reduce overfitting problems. In many circumstances, using this kind of approach may improve the generalization capabilities of the network with respect to unseen objects, particularly when the number of parameters becomes large with respect to the size N of the learning set.

Entropy based criteria

Various other types of fitting criteria have been derived from the logarithmic entropy function. These are interesting alternatives in the case where the output information corresponds to conditional class-probabilities [RI91]. In this case, we assume that the output neurons correspond to the classes, and, ideally, the output vector would be equal to the vector of conditional class probabilities $\mathbf{p}(a)$ corresponding to the input attribute vector.

For example, the total residual entropy of the LS classification given the network weights may be defined by

$$N * H_{C|w_{i,j}}(LS) \triangleq - \sum_{o \in LS} \log P(c(o)|w_{i,j}). \quad (5.30)$$

Here $P(c(o)|w_{i,j})$ denotes the activation of the output neuron corresponding to the class $c(o)$ for each object, which is interpreted as the conditional probability of the object's class predicted by the neural network model.

On the basis of the analogy of this criterion with the entropy criterion of §3.4 used to evaluate decision trees, we may suggest the following artificial neural network *quality* measure

$$Q(ANN, LS) \triangleq N * I_C^{ANN} - \beta * C(ANN), \quad (5.31)$$

where $I_C^{ANN} = H_C(LS) - H_{C|w_{i,j}}(LS)$ denotes the mean information provided by the ANN, and $C(ANN)$ its complexity, e.g. its number of weights.

Various theoretical *minimum encoding length* or *maximum a posteriori probability* interpretations may be derived for this criterion [WE 94b]. From the *practical* viewpoint, using the same approach to evaluate decision trees and neural networks may allow us to compare them on the basis of a learning set by taking explicitly into account their complexity. This in turn may offer interesting possibilities of combining these approaches as suggested in chapter 6.

5.2.4 Efficient network optimization algorithms

The most obvious and simple way of using the back-propagation algorithm to optimize the neural network fit to the learning set, is to use the fixed step gradient descent algorithm, which is classically referred to as *the* error back-propagation algorithm [HE 91]. Unfortunately, this approach, already very slow in convergence in the single layer perceptron case, is even much slower in the case of multiple non-linear layers. In practice, the computing times become rapidly prohibitive as soon as the number of weights and learning states increase.

In the literature, a very large number of alternative algorithms have been proposed to speed up the convergence. The earliest methods, which consisted basically of adding a heuristic “momentum” term to the gradient, present the advantage of preserving the locality of the weight update rule of the back-propagation algorithm, which is their main attractive feature. Unfortunately, these ad hoc methods require, in general, a tedious manual tuning of their parameters, which for large scale problems may become very time consuming.

More recently, a certain number of researchers have proposed the use of some of the classical unconstrained optimization algorithms available from the optimization

literature [WA 87, PA 87, FO 92]. The very important improvement in efficiency obtained with respect to the standard steepest descent algorithm and the fact that no user defined parameters must be tuned has led us to use this type of approach.

Since it is not our purpose to discuss the broad topic of non-linear function optimization, we will briefly describe the particular method which we have been using in most of our simulations in the context of power system security problems. This is the “Broyden-Fletcher-Goldfarb-Shannon” (BFGS) quasi-Newton algorithm, already introduced in our example of §5.2.1.

Basic iterative optimization scheme

The basic scheme of the iterative optimization methods consists of defining at each step of the process a search direction \mathbf{s} in the weight space, and searching for the minimum of the error function in this direction. This is a scalar optimization problem

$$\min_{\lambda} ERR(\mathbf{w} + \lambda \mathbf{s}), \quad (5.32)$$

where \mathbf{w} denotes the weight vector.

The steepest descent method consists of moving in the direction opposite to the gradient

$$\mathbf{s} = -\nabla_{w_{i,j}} ERR(\mathbf{w}), \quad (5.33)$$

which leads to a zigzag optimization path, which converges slowly in some circumstances.

Quasi-Newton optimization

A better approach, at least nearby the solution, would be provided by a Newton-like method consisting of computing the search direction by

$$\mathbf{s} = -(\nabla^2 ERR(\mathbf{w}))^{-1} \nabla ERR(\mathbf{w}). \quad (5.34)$$

This approach may unfortunately be inefficient due to the high cost of computing the $\frac{w(w+1)}{2}$ terms of the inverse Hessian matrix $(\nabla^2 ERR(\mathbf{w}))^{-1}$.

Thus, the basic idea of the quasi-Newton family of methods consists of building up iteratively an approximation of the inverse of the Hessian matrix from repeated computations of the gradient.

More precisely, the BFGS variant which we have used, is based on the following update scheme at step k [FO 92]

$$\mathbf{s}^k = -H^k \nabla ERR(\mathbf{w})^k, \quad (5.35)$$

and

$$H^{k+1} = H^k + \left(1 + \frac{\delta_{\nabla}^T H^k \delta_{\nabla}}{\delta^T \delta_{\nabla}}\right) \frac{\delta \delta^T}{\delta^T \delta_{\nabla}} - \left(\frac{\delta \delta_{\nabla}^T H^k + H^k \delta_{\nabla} \delta^T}{\delta^T \delta_{\nabla}}\right), \quad (5.36)$$

where δ denotes the change of the weight vector at step k as determined by the optimal search in the direction s^k and δ_{∇} denotes the change in the gradient direction from step k to step $k+1$. The method starts with an initial guess H^0 of the Hessian matrix which is generally taken as the identity matrix.

As we have observed in practice, the use of this method allows us to considerably reduce the computational burden of the neural network learning, without requiring any manual tuning of parameters.

While these quasi-Newton methods are the most prevailing efficient techniques used in the context of feed-forward network learning, together with the conjugate gradient methods, they still remain iterative in essence. In particular for real life, medium to large scale problems, they may still require a large number of rather lengthy iterations, without guaranteeing global optimization.

Illustrative example

To fix ideas, we have applied the above technique to our example problem of transient stability described in §3.4. The neural network structure uses a single hidden layer, composed of 20 neurons which receive their input from 68 input neurons corresponding to the 67 pre-whitened candidate attributes and a constant input set to 1. Finally, the output layer is composed of 2 output neurons, one for the stable class and one for the unstable class.

The hyperbolic tangent activation function was used for each hidden and output layer neuron, and all in all this - apparently simple - network structure corresponds to $68 \times 20 + 20 \times 2 = 1400$ adaptable weights. We have used the first 10000 learning states of the data base as a learning set, so as to provide results comparable with the decision tree built for the same problem, described in §3.4.4.

The BFGS algorithm was used to learn the network weights which allowed us to reduce the MSE initially of 21040 to a final value of 2134, within a total number of 532 iterations corresponding to a CPU time of 140 hours. To reduce overfitting, a regularization term equal to $\sum_{i,j} w_{i,j}^2$ was included in the optimality criterion.

The resulting network was tested on the test set composed of the 2497 remaining states, resulting in an overall test set error rate of 2.44%, which compares rather favorably with the 4.21% obtained with the DT of Fig. 3.16. Figure 5.9 illustrates the variation during the iterative process of both the MSE determined in the learning set and the classification error rate computed in the independent test set.

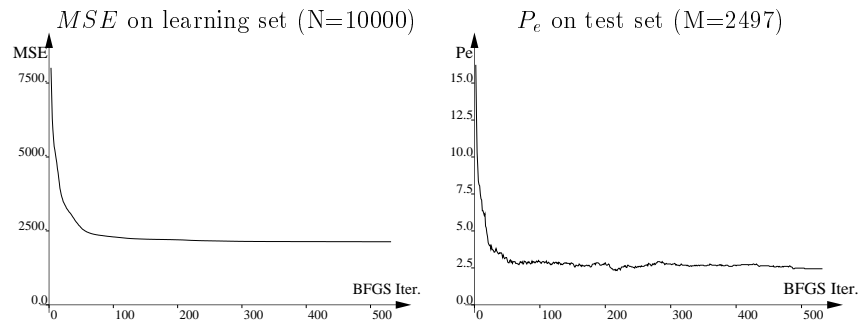


Figure 5.9 Convergence of the BFGS algorithm for the transient stability example

These curves illustrate quite well the practical difficulty of deciding an appropriate stopping rule for the iterative optimization algorithms. In this particular example we could have decided to stop the algorithm somewhere between 100 and 200 iterations, which would have allowed us to reduce the CPU time to about 60 hours, which is still very slow.

To evaluate the practical advantage of using a second order quasi-Newton approach together with a regularization term in the optimality criterion, we have repeated the above simulation using the basic steepest descent procedure together with a MSE cost function without penalization of weights; this is often considered as the “standard” back-propagation method. This computation did not converge perfectly and was thus stopped after 2000 iterations, corresponding to a CPU time of 443 hours. The final value of the MSE was 752, which is significantly lower than the value obtained above. However, the corresponding test set error rate was 3.12%, which is slightly higher than the 2.44% obtained above. The variation of the MSE and error rate during the successive iterations are shown in Fig. 5.10. It is interesting to observe that while the test set error rate stops decreasing after 650 iterations, the MSE continues to decrease steadily during the 2000 iterations.

We have also applied the BFGS algorithm to the same non-regularized MSE error criterion. It converged after 309 iterations (about 80 hours) to a MSE value of 144 and a test set error rate of 3.92%. Thus, in the present example using a regularization term actually allowed us to reduce the error rate from 3.92% to 2.44%, and the error rate of 3.12% obtained by the gradient descent algorithm was due to chance, because we stopped the algorithm “prematurely”.

The slowness of the neural network optimization algorithms, even in the case of intrinsically efficient quasi-Newton methods, makes practical experimentation with this method hardly feasible for real sized problems, even with the most efficient presently available computing hardware. In particular, the trial and error method suggested in the next section for determining an appropriate network architecture is possible only

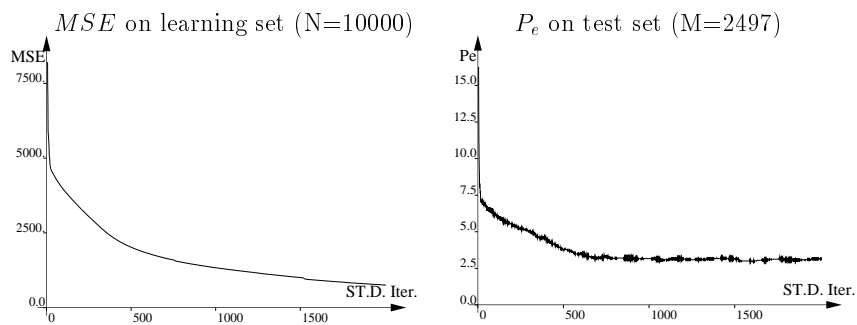


Figure 5.10 *Convergence of the steepest descent algorithm*

with large amounts of available computing power and small to medium problem sizes.

On the other hand, the flexibility of the neural network model allows it to fit many non-linear classification boundaries. In particular, as we have observed in the present example, it is often possible to reduce the error rate with respect to competing methods, such as the decision trees.

Finally, with the multi-layer perceptron and currently available methods, there is no support to help the user to interpret the resulting set of weights. This is particularly problematic in the context of high-dimensional input spaces, where often only a reduced number of attributes are actually useful for discrimination.

Thus, at the current stage of method development, we may consider the multi-layer perceptron as a flexible and generally accurate, but very slow and mostly black-box approach. In particular, the danger in this black-box nature comes from the fact that the multi-layer perceptron may exploit - without notifying it - abnormal correlations existing among some input variables and the output classification in order to maximize the fit. For example, such pathological correlations may be unduly introduced during the building of the learning (and test) samples, and this may lead to dangerous extrapolations.

Another problem concerns overfitting and generalization in regions of low probability. During learning, the neural network parameters are modified so as to reduce the MSE mainly in the regions of high density in the learning set, and this leads often to sacrificing accuracy in regions of lower density. This may sometimes lead to non-sense extrapolations, particularly when the output information varies in an important fashion in the denser regions.

For example we illustrate in Fig. 5.11 a typical problem which may be encountered in practice with the multi-layer perceptron. The three-dimensional graphs show the

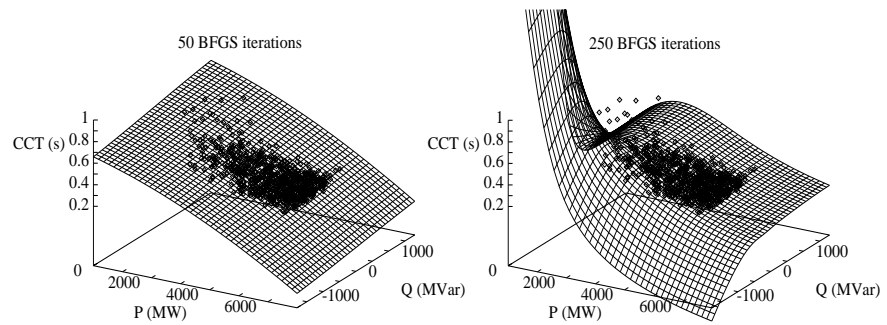


Figure 5.11 *Abnormal extrapolations due to overfitting*

relationship between the critical clearing time¹ of a three-phase short-circuit nearby an important power plant of the EDF system (see §13.3) and the total active and reactive generation of the power plant.

The continuous surfaces show the mapping obtained via an MLP approximation of this stability margin after respectively 50 and 250 BFGS iterations. Notice that locally within the CCT interval of $[0.150s \dots 0.350s]$ its value may be approximated quite precisely using only the two attributes concerning the state of the nearest power plant. However, for higher CCT values other effects related to the state of other nearby power plants may also influence significantly the value of this stability margin.

Since the corresponding state variables have not been used as input attributes in the present example, these effects may be considered here as noise. The difference between the continuous mapping of the CCT via the MLP and the actual CCT values determined by numerical simulation are shown by the cloud of points which represent the actual values (P, Q, CCT) of a sample of 800 learning states. This shows clearly that the mapping corresponding to 250 BFGS iterations tries to approximate more closely the values observed in the learning set in the region $[0.150 \dots 0.500]$ where the majority of the latter states lie. Unfortunately this is achieved by sacrificing the fitting to the few points farther away from the center of the cloud, thereby providing a pathological behavior.

Notice that the above kind of problem may not be detected by monitoring statistical features like MSE (even in a test set) or classification error rates. Only a more in depth analysis of the relationship modelled by the neural network would enable the detection of such abnormal behavior. While this was rather easy in the three-dimensional case of the above illustrative example, it is hardly possible in the context of large scale power system security analysis problems.

¹The critical clearing time (CCT) of a fault is the maximum time duration it may take the protection system to clear the fault without causing an irrevocable loss of synchronism.

5.2.5 Network architecture and data pre-processing

From the theoretical point of view, several results exist showing that provided non-linear activation functions are used, and if the topology of the network is sufficiently complex, most practically interesting input/output mappings, if not all, may be *represented* by the multi-layer perceptron with arbitrary good precision.

In practice, seldom more than two hidden layers are considered. In all our practical simulations we have even found that a single hidden layer, with a rather small number of neurons, seems to be sufficiently powerful, although we have not made many trials due to the time taken by such simulations.

For a single hidden layer perceptron the total number of weights w is equal to $(n + r + 1) * h$, where n , r , h denote respectively the number of input attributes, output variables and hidden neurons. An often used rule of thumb consists of choosing h so as to obtain a number of weights w equal to the number of learning samples divided by a constant factor, say five to ten, so as to ensure a high enough redundancy in the learning set and reduce overfitting.

Data pre-processing mainly consists of scaling the input attributes, so as to avoid saturating the non-linear activation functions during the initial iterations of the back-propagation process. Such a saturation would lead to a flat MSE behaviour and the possible freezing of the network weights to their initial values. In the context of classification problems, we have generally used the -1/+1 output encoding, using one output neuron per class. In the context of regression problems, for example when trying to obtain a security margin as output we have observed that the proper scaling of the output information not only improves the speed of convergence but also the quality of the solution.

Another interesting possibility consists of using the hybrid approach discussed in chapter 6, to determine the appropriate input attributes and the structure of the multi-layer perceptron on the basis of a decision tree previously built and converted.

Finally, various other techniques have been proposed in the literature to determine the appropriate structure of a network which we did not use, either because they were not implemented in the back-propagation software used for our simulations (e.g. the optimal brain-damage technique) or because they would have led us into prohibitive computing times, without promising practical benefit (e.g. the iterative network growth). We refer the interested reader to the reference [HE 91] for a description of current research on network growing and pruning.

Anyway, we believe that the projection pursuit technique discussed in §4.3.2 provides a more attractive solution to this problem. Further, in the context of two power system security problems, this method has already obtained significantly better results than the multi-layer perceptron [TA 94].

5.2.6 Interpretations of neural network models

In this section we discuss briefly various interpretations of the multi-layer perceptron which have been proposed. The difficulty of interpreting the meaning of a feed-forward neural network as a function of its weights is a well agreed weakness. As we have mentioned above, the general feed-forward neural networks provide therefore essentially a black-box model of a problem. As formulated by Towell and Shavlik [TO 93]

... this is a significant shortcoming, for without the ability to produce understandable decisions, it is hard to be confident in the reliability of networks that address real-world problems.

Probably, there is a fundamental dilemma between conceptual simplicity on the one hand, which enables us to interpret and understand a model, and representation power on the other hand, which provides flexibility and accuracy in general.

The high representation power of the feed-forward neural network models is responsible for their success in terms of accuracy. For the same reason, it is quite easy to represent any kind of more or less simple, restrictive class of models by neural networks. We will give some symbolic and geometric examples of this below. Unfortunately, this overly general nature prevents us from providing the reverse mapping, translating a neural network into a simpler, easily interpretable model, without making restrictive assumptions and often unacceptable approximations and loosing the benefit of the MLP model.

The analysis and interpretation of neural network models is an active research area, but no approaches which would at the same time be general and satisfactory have yet been proposed. The main reason for our scepticism on the eventual development of such methods is the observation that in general the number of parameters of a neural network is much larger than its number of input variables, frequently one or two orders of magnitude. Thus, trying to understand a problem by interpreting the corresponding neural network model, in terms of its weights, may turn out to be much more difficult than trying to understand the original problem directly.

One possible approach, which we will advocate in the next chapter, consists of using a hybrid methodology, maintaining at the same time an easily understood model (e.g. a decision tree) and its alter ego in terms of a more accurate but black-box model (e.g. a feed-forward neural network). The latter may provide improved accuracy by *slightly* deviating from the former model, without however jeopardizing their overall consistency in interpretation.

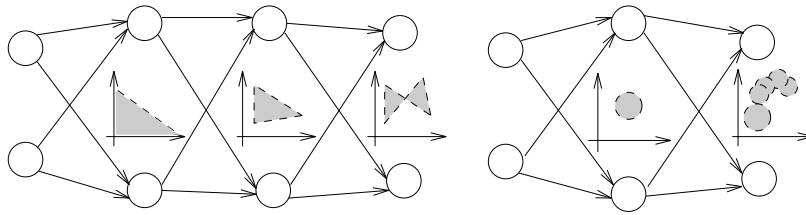


Figure 5.12 The “hyperplane-box-region” model and the “prototype” model

Neural network implementation of frequently used models

To illustrate the generality of the feed-forward neural network model, we provide some classical interpretations, suggesting that this is indeed a rather powerful and general model.

Generally speaking, the first hidden layer may be considered as a feature extraction layer, and the subsequent layers are viewed as providing logical combinations of features.

The first such model, illustrated in the left part of Fig. 5.12, corresponds to the use of sigmoid type activation functions. In this setting, the weights arriving at a given neuron of the first layer define a hyperplane “à la perceptron”. The activation of such a state will be high or low according to the semi-space in which the input pattern lies. For a set of h such hidden neurons, the activations provide a bit-pattern indicating the membership of the current attribute vector in each of the corresponding semi-spaces. A neuron in a second hidden layer may then combine this information so as to test whether the state belongs to the intersection of some of these semi-spaces. Thus h' neurons in a second hidden layer allow us to define a set of h' convex boxes, and finally the third output layer may associate the union of some of the convex regions of the preceding layer with each output neuron. This allows us to define output classes of arbitrary shape. In addition, taking into account the fact that the activation functions may vary smoothly from -1 to +1, this type of network will actually allow us to define regions as fuzzy sets.

Another, closely related model shown in the right part of Fig. 5.12, uses a single hidden layer with kernel type (e.g. Gaussian) activation functions, together with a simple linear output layer. In this interpretation, the weights from the input neurons to a hidden neuron may be considered as defining the location of a prototype in the (augmented) pattern space, and the activation of the neuron will be high only if the input attribute vector is sufficiently close to this prototype. The weights to the output layer combine the proximity information so as to approximate the desired output. For example classes may be defined as unions of proximity regions surrounding the prototypes, and regression functions may be considered as the superposition of kernel

functions.

Another, similar representation uses sine or cosine activation functions, and the neural network model may be interpreted as a kind of Fourier analysis technique. A further generalization of this idea leads to the functional link network proposed by Pao [PA 89b], which is based on an extended attribute space defined by an a priori given set of linearly independent functions of the input attributes, which are used in a single layer perceptron fashion. This is the neural network version of the generalized linear discriminants discussed in [DE 82].

5.3 KOHONEN FEATURE MAPS

We now turn for a brief while back to the realm of unsupervised learning, and consider one of the neural network based approaches to this problem, namely the feature maps developed by Kohonen [KO 90]. Our aim is not to discuss the neural network approaches to unsupervised learning in general, and there are many other interesting such approaches for feature extraction, clustering or data compression which would be interesting to consider [PA 89b, ZU 90, HE 91].

There are three main reasons why we have chosen to describe the Kohonen feature mapping approach. First, it is a promising method for data analysis, due to its graphical interpretation possibilities, and could be particularly useful in the context of power system security assessment, where in depth data analysis is of paramount importance. Second, this method is essentially *complementary* to the classical statistical techniques of unsupervised learning, presented earlier. Finally, some interesting applications of the Kohonen's feature map to power system security problems have been proposed in the literature, and our brief description of the technique should provide the basic notions required for understanding our later discussion of these applications.

5.3.1 Unsupervised learning

The self organizing feature map (SOM) developed by Kohonen belongs to the category of competitive learning models which aim at representing a data set by a smaller number of representative prototypes. There are many possible practical motivations for this kind of approach. For example in the context of information communication, this may provide an efficient way of encoding information. In the context of data analysis it may provide a small representative subset of states.

In comparison to the other similar methods, e.g. the clustering algorithms discussed in §4.4, the main originality of the SOM is that it allows us to organize the learned prototypes in a geometric fashion, for example on a uni- or a two-dimensional regular

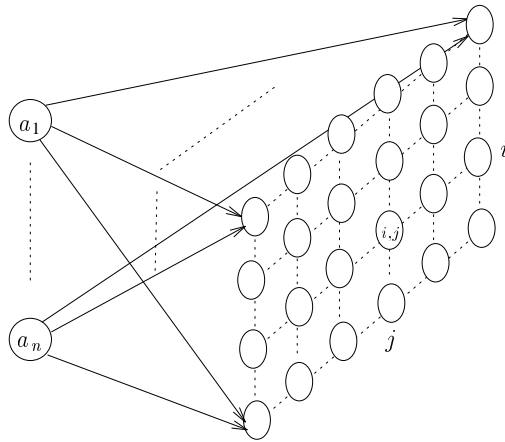


Figure 5.13 Two-dimensional Kohonen feature map

grid or map.

In the sequel we will particularize our presentation to the two-dimensional case, which is the most usual one, for graphical interpretation reasons. The interested reader may refer to the paper by Kohonen [KO 90] for a general description and an in depth presentation of the biological motivations underlying the two-dimensional SOM.

To fix ideas we have represented in Fig. 5.13 a hypothetical two-dimensional 4×6 feature map. Each neuron i, j corresponds to a prototype in the attribute space, say $\mathbf{a}^{i,j}$. The connection weights from the input layer to the map correspond to the attribute values of the corresponding prototype. Further, in addition to an a priori defined distance $\delta(\mathbf{a}^{i,j}, \mathbf{a}^{k,l})$ in the attribute space, the relative location of these prototypes on the feature map defines a *topological* distance.

In this model, the output corresponding to an object o is defined as the nearest prototype in the attribute space, i.e. \mathbf{a}^{i^*,j^*} such that

$$\delta(\mathbf{a}(o), \mathbf{a}^{i^*,j^*}) \leq \delta(\mathbf{a}(o), \mathbf{a}^{i,j}), \quad \forall i, j. \quad (5.37)$$

What is expected from the learning algorithm is to define the prototype vectors so as to minimize the quantization error, e.g. in the MSE sense (as in the statistical clustering algorithms of §4.4), and in addition to define the positions of these prototypes on the feature map, so as to preserve the topological properties of the original attribute space. More precisely, we expect prototypes which are close in the original attribute space to be located close on the map.

Notice that this kind of objective is not very different from multi-dimensional scaling, which aims at finding a configuration of points (e.g. the prototypes) in a low-

Table 5.3 *Kohonen self-organizing map learning algorithm*

-
1. Consider the objects of the learning set in a cyclic or random sequence.
 2. Let o be the current object, $\mathbf{a}(o)$ its attribute vector, and \mathbf{a}^{i^*,j^*} its closest current prototype.
 3. Adjust the prototype attribute vectors according to the following correction rule

$$\left(\mathbf{a}^{i,j}\right)^{new} = \left(\mathbf{a}^{i,j}\right)^{old} + \eta \Lambda(i - i^*, j - j^*) \left(\mathbf{a}(o) - \left(\mathbf{a}^{i,j}\right)^{old}\right). \quad (5.38)$$

dimensional space such that the distance among the points in this low-dimensional space corresponds to the distance among prototypes in the original attribute space [DU 73].

Kohonen's algorithm

The elementary learning algorithm is an iterative method considering the learning set objects successively and updating the weight vectors at each step, so as to reinforce the proximity of the object and its currently closest prototypes. This is indicated in Table 5.3 in the particular case of a two-dimensional feature map.

The parameter η denotes the learning rate of the algorithm, and the function $\Lambda(\cdot, \cdot)$ is a neighborhood function, i.e. a decreasing function when the distance on the feature map increases. A frequent choice is to use the Gaussian kernel

$$\Lambda(x, y) = \exp\left\{\frac{-(x^2 + y^2)}{2\sigma^2}\right\}. \quad (5.39)$$

Both the learning rate η and the width parameter σ are in practice gradually decreased during successive learning iterations. Thus, initially corrections are made so as to move a large part of the prototypes at each iteration considerably closer towards each learning object. At the later iterations, only the nearest neighbor prototype is moved and only a small correction is made at each step.

Unfortunately, the theoretical analysis of this learning algorithm has not yet been carried out very far, and among the many questions which may be raised only a few have been answered and only in the simple one dimensional case.

Intuitively, we may feel that the above algorithm will tend to minimize a quadratic quantization error in the learning set. Of course, at best a local minimum of this

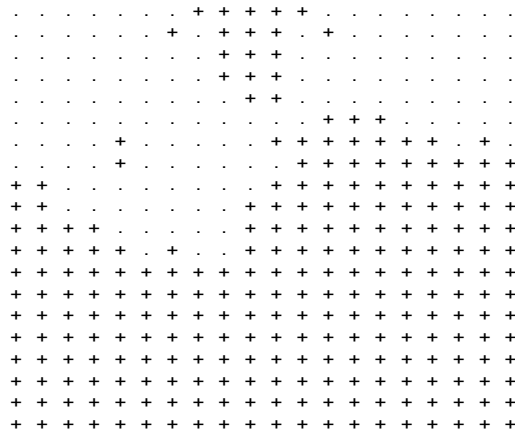


Figure 5.14 Kohonen map for the voltage security example. Adapted from [TA 94]

quantization error may be reached. Further, the meaning of this criterion depends, of course, on the scaling of the input attributes, in the practical case of a learning set of finite size.

On the other hand, in the case of a one-dimensional attribute space, it is possible to show that asymptotically the prototypes are regularly spaced on the feature map with an attribute density proportional to $p(a)^{2/3}$ where $p(a)$ denotes the probability density in the original attribute space. So, the Kohonen feature map tends to place the prototypes by undersampling high probability regions and oversampling low probability ones [HE 91].

5.3.2 Possible uses

The SOM is often used for graphical data inspection and visualization.

For example, a typical application consists of building a two dimensional feature map and displaying graphical information on this map, showing class labels or attribute values in terms of the i, j coordinates. This can also be used for monitoring the position of objects on the map [NI 91, MO 91].

Illustration 1. Similarities among power system states.

To fix ideas, we have represented in Fig. 5.14 a feature map which has been constructed for the academic voltage security example of §10.2, which was studied in the context of the Statlog project. A random sample of 1250 *just after disturbance* (JAD) states was generated, and each state is characterized by 28 attribute values, corresponding to the power flows, voltages and reactive power reserve.

The Kohonen map of Fig. 5.14 was determined without using information about the classification (critical vs non-critical) of the power system states. After convergence, the labels indicated in Fig. 5.14 were calibrated by determining the nearest neighbors in the learning set of each prototype vector and by associating to the latter the majority class among its nearest neighbors. In Fig. 5.14 “+” represents a prototype corresponding to a majority of critical situations, and “.” a prototype corresponding to a majority of non-critical situations.

The clustering apparent in Fig. 5.14 shows, for example, that there may be two distinct types of non-critical states [TA 94]. Monitoring the position on the map of the real-time power system state could provide a means to display security information to an operator. Using the latter map as a nearest neighbor classifier yields a test set error rate of 5.6%, determined in an independent test set composed of 1250 other states, generated in a similar fashion to the learning states. This is however a rather large error rate, since for the same problem the decision trees obtained a test set error rate of 3.8% and the multi-layer perceptrons yielded error rates of 1.7%.

Illustration 2. Similarities among physical parameters.

Finally, anticipating on the presentation of the voltage security study on the EDF system in §14.4, we provide an illustration of an interesting possibility of using the SOM for analysing physical correlations among variables.

To fix ideas, let us consider the problem of defining a set of representative attributes to characterize voltage behavior of a large scale power system in the JAD state, which is considered in the context of emergency state detection of voltage critical situations. For this problem, physical considerations suggest that the low-voltage side voltage magnitudes at the EHV/HV transformers may provide a very good picture of the severity of the disturbance and at the same time will reflect the amount of load which would be restored due to the automatic action of the transformer taps. Thus, these variables are liable to provide good indicators to detect critical situations.

However, even in a restricted region of a large scale power system, such as the one studied in §14.4, there may exist a rather large number of such transformers and correspondingly a large number of HV voltage candidate attributes.

Thus, there is a need to compress this information into a smaller number of *equivalent* voltages, in short there is a need to identify the *voltage coherent* regions in the power system. Once these regions are identified we may define equivalent voltages through the aggregation of elementary voltages in each region.

This is a typical attribute clustering problem, which we may try to solve with the Kohonen feature map. In our example, we start with an initial set of 39 HV voltage attributes. Each attribute is characterized by the value it assumes for a random sample of JAD states. For each attribute the same sample of states is used corresponding, in

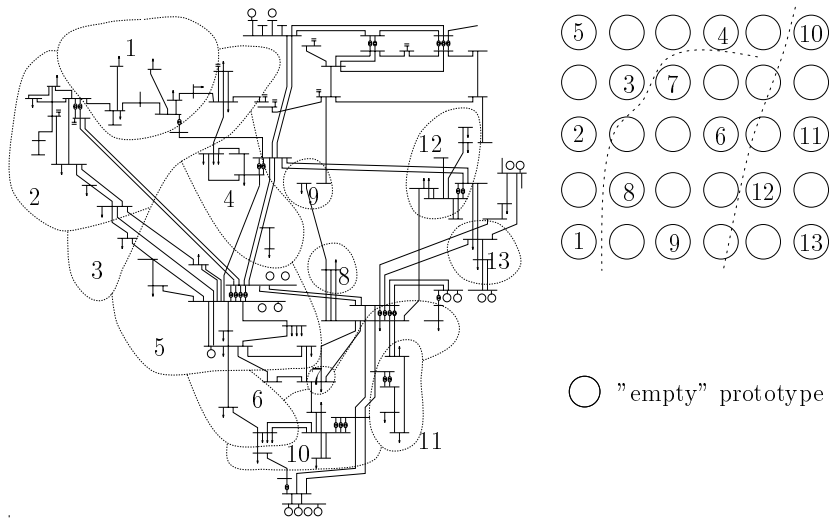


Figure 5.15 Voltage coherency SOM

the case of our illustration, to a given disturbance and 100 randomly generated pre-fault operating states.

Thus the “learning set” is composed of 39 vectors of 100 components. These vectors are pre-whitened and the Euclidean distance used by the self-organizing learning algorithm becomes equivalent to the correlation coefficient. In other words, this algorithm will try to identify regions of strongly correlated voltages. To this end, we specify a 5×6 feature map which is randomly initialized, and adapted on the basis of the above learning set. After convergence, each cell corresponds to a new vector of 100 components. The map is calibrated by identifying for each one of the 39 vectors corresponding to the 39 HV voltages its nearest neighbor on the map, i.e. the prototype to which it is most strongly correlated.

The obtained clustering is represented in the right part of Fig. 5.15. The non-empty cells correspond to the actual 13 prototypes determined by the algorithm. Each prototype corresponds to a set of HV voltages of which it is the nearest neighbor among all prototypes defined on the SOM. The empty prototypes are those which are the nearest neighbor of no HV voltage at all. In the left part of Fig. 5.15 the regions corresponding to the non-empty prototypes have been represented on the one-line diagram of the EDF system.

It is interesting to notice that the location on the SOM of the prototypes corresponding to the voltage coherent regions may be compared with the adjacency of these regions on the one-line diagram. For example regions No. 10, 11, 12, 13 which are located rather far away from the voltage weak region are also grouped together and away from

the other prototypes on the feature map. On the other hand, the intermediate regions No. 6, 7, 8, 9 are also located in an intermediate position on the feature map. Finally, the regions No. 1, 2, 3, 4, 5, which are at the hart of the physical voltage security problem, are located in the left part of both the one-line diagram and the feature map. This illustrates rather well the capability of the Kohonen SOM to preserve topological relationships among prototypes.

The above analysis, although based on a very small sample of 100 states, reflects physical information confirmed by operators' knowledge and is also consistent with other statistical analyses based on the hierarchical clustering algorithm presented in 4.4.2. With respect to this latter method, the Kohonen algorithm has several potential advantages.

First of all it is in principle able to determine automatically the appropriate number of prototypes. In our example, this led to 13 voltage coherent regions, although the maximum possible number of prototypes in the used feature map was 30.

In addition this method provides an indication of the topological relationship among prototypes, in terms of the distance on the feature map. We feel that this may be particularly useful in the context of the determination of coherent regions, where the resulting organization of prototypes may be compared with the electrical distances in the power system.

In comparison to sensitivity based coherency approaches, the present technique is much more flexible and potentially much more powerful. Indeed, the sensitivity techniques are essentially providing a case by case analysis, which is determined for a given power system topology and operating state. The present approach, however, provides a systematic analysis which is based on a statistical sample which may be either very specific or very diverse, depending on the type of analysis sought.

5.3.3 Supervised learning

Many parameters must be tuned in practice before obtaining good results with the above algorithm in terms of a low quantization error. This concerns first of all the choice of an appropriate map topology and neighborhood function, and a distance definition in the original attribute space. This latter is often based on an Euclidean type of distance based on some previously extracted features, e.g. a subset of pre-whitened attributes. The other choices concern parameters of the algorithm such as the success criterion and rules to define the learning rate and window, and the initial location of the prototypes. Often, several learning sessions are run in parallel on the same data set, and the most successful result is chosen as the final SOM, on the basis of the corresponding quantization error criterion.

If used correctly, the above technique may allows us to design a set of prototypes

which provide a good approximation of the information contained in a learning set, as described by a set of attributes. This may directly be used for classification purposes, or similarly for regression, by calibrating the prototypes on the basis of the learning set. For example, for each prototype we may count the relative frequency of learning states of each class of which the prototype is the nearest neighbor among all prototypes on the map. These may then be used so as to associate a conditional class probability vector and a corresponding majority class.

The above is the most elementary and simplest way of exploiting a SOM for prediction. One may however argue that this will not lead necessarily to a good behavior in terms of classification reliability, since the class information is attached a posteriori but has not been used during the construction of the map. Indeed, in practice this method turns out to provide very deceiving results in terms of classification accuracy. For example, in the Statlog study the results obtained scored worst among all methods which have been tried [TA 94].

A better idea would consist of using classification information during adaptive training, so as to take into account this information to control the location of the prototypes. Applying this idea yields the so-called “learning vector quantization” (LVQ) family of methods proposed by Kohonen [KO 90], which modify the reinforcement rule of Table 5.3 so as to improve the classification reliability. We will not describe these methods in detail, but the basic idea consists of attaching a priori a class label to each prototype, and changing the sign of the $\Delta\alpha^{i,j}$ correction term for those prototypes which do not correspond to the same class as the current object.

5.4 CONCLUDING REMARKS

There is a very large number of neural network techniques for supervised learning, both for classification and regression type of problems, as well as for unsupervised data analysis and clustering.

In our description we have merely presented the two techniques which have received most of the attention of researchers in the context of power system security applications, by trying to give an honest look at these techniques, guided by our own practical experience and the in depth study made in the context of the Statlog project. For the interested reader, we strongly recommend reading the final report of the latter project [TA 94], which gives a dispassionate account of the state of the art in classification methods.

The fact that we have chosen to describe both the multi-layer perceptron and the Kohonen self-organizing map may be interpreted as a definite conviction of the future usefulness of these methods in the context of power system security assessment problems. However, this does not imply that other methods which we have not described,

could not be interesting. Our purpose was mainly to provide the reader with a taste of the practical advantages and difficulties of either method, which are complementary in nature with other methods described in the earlier chapters.

Whatever their attractiveness, we believe that at the current stage of technology, the main difficulty with these methods is their lack of interpretability features, in particular in comparison to the machine learning methods. We have discussed this in several places and have shown that it might prevent the methods from being used in the context of real-world applications.

If we compare the two types of neural networks discussed in this chapter, we observe first of all that the multi-layer perceptron techniques are very powerful in terms of accuracy as well as being easy to apply.

In particular, with the more efficient second order quasi-Newton optimization methods, no prior parameter tuning is required and learning times may be reduced so that the application to problems of realistic size becomes feasible. These latter methods lead however to more complex software implementation, and still suffer from high computing requirements; an improvement of two orders of magnitudes would be required to allow response times to become small enough for interactive experimentation, within the context of real-world power system security problem sizes.

The reverse situation holds for the Kohonen network, which has a rather fast and straightforward learning algorithm but where it is the user's responsibility to adapt parameters so as to obtain interesting results. This method certainly requires some more expertise to get the best out of it.

Resuming our discussion about the appropriateness of distinguishing between "statistical" and "neural" approaches to learning, we may observe that the probabilistic framework used in the classical statistical methods is an important tool for the study of neural network approaches. This is also reflected by the significant fraction of the more recent theoretical work on neural networks, which deals with probabilistic modelling and statistical analysis [LE 90c, BU 91, RI 91, RI 93]. At the same time, modern statistical methods (e.g. the projection pursuit techniques [FR 81, HW 93]) are obviously closely related to the connectionist models.

On the other hand, from the implementation point of view, the high parallelism of the connectionist models is equally present in many, if not all, of the classical statistical methods (nearest neighbor, kernel density estimation, projection pursuit, . . .).

Thus, our classification into statistical and neural approaches is only for convenience of presentation, and we don't believe that from the viewpoint of applying methods of either of these categories to power system security problems there would be a fundamental distinction. More precisely, we believe that the differences among the individual methods are more significant than the differences among the classes of

approaches. The difference often lies more in the way these methods have been applied in the past than in the algorithms. The neural network approaches have generally been applied in a more or less black-box fashion whereas the statistical techniques use a modelling approach, in order to identify and validate simplifying assumptions about the problem structure, such as independence and normality. Consequently, neural network techniques have mostly been applied as a stand-alone tool, while the statistical techniques usually rely more strongly on a priori analysis of problem features and on choosing appropriate data transformations for input and output representations.

6

Hybrid approaches

6.1 INTRODUCTION

In the preceding three chapters we introduced a certain number of supervised and unsupervised learning techniques, each one of which has its functionalities and also its range of problems where a near optimal behavior may be expected. On the other hand, many practical problems may require a combination of these methods for their solution and from the methodological viewpoint, cross-fertilization among approaches may lead to better, essentially hybrid strategies.

In the recent years, a growing number of hybrid methods have been published combining aspects from machine learning with statistical and neural network approaches. In this chapter we will briefly describe some possible combinations of the decision tree induction technique, which fills the basic requirement in the context of many power system problems of interpretability and efficiency, and some other techniques which may offer some possibilities to enhance this approach in order to extract more information from the available data bases. In the process we will also provide some references to other research work in the context of hybrid learning techniques.

From our practical point of view the aim of these hybrid approaches is mainly to improve the accuracy of the security classification obtained by a decision tree, and in particular to reduce as much as possible the risk of not detecting insecure situations without increasing too much the false alarm rate.

6.2 MACHINE LEARNING AND NEURAL NETWORKS

6.2.1 Introduction

There are several approaches to combine the idea of iterative tree growing algorithms with the flexibility of general feed-forward neural networks.

The first type of approach uses a greedy network growing algorithm, which is strongly inspired by the techniques used to build the trees. This yields a class of tree-structured neural network training algorithms which are more or less closely related to the TDIDT framework, and which aim essentially at fitting the network complexity to the available data and to reduce computation times during the learning stage [SA 91c, CI 92]. The projection pursuit algorithm discussed in §4.3.2 appears also clearly as a greedy type of algorithm although the resulting network structure is not organized in a tree fashion.

Another approach consists of constructing decision trees using more complex surfaces in the attributes space than single attribute (threshold) tests in order to split at a tree node. Each of these surfaces is then implemented by a neural network model. The earliest such methods merely used perceptrons or linear discriminants in order to determine an appropriate linear combination [BR 84, UT 88]. The basic idea is to enhance the decision trees to be able to identify some cross-correlations among *several* attributes and the goal classification. This was also the motivation behind the search for linear combination attributes described in §3.4.3. As noted earlier, while these enhancements may significantly improve the accuracy of the decision trees, it is also true that they may hinder the interpretation of the tree's information.

Thus, in the above approach there are still some developments required in order to be able to assess, at the tree growing stage, whether the increased test complexity yields indeed a significant improvement in accuracy and if not, to rely on the simpler standard kind of node splits. In particular the training algorithms should be able to find a compromise between trees using a too high number of too simple tests and those using a too small number of too complex tests. This would in turn require us to develop a measure of *test complexity* which should be combined with the classical measures of *structure complexity* used in the quality measures (e.g. in eqn. (3.14)). This possibility is further discussed in [WE 94b].

Finally, the last technique, which is also the most simple one to implement, consists of using a two-stage process. In the first stage, a decision tree is derived in order to compress information contained in the data base. This allows us in particular to determine the attributes which have significant correlation with the target classification. In the second stage this reduced set of variables is used as input attributes to the neural network model, which is further adapted on the basis of the learning set, using an available standard back-propagation software package.

This strategy has the main advantage of simplifying considerably the resulting neural network structures and thereby reducing dramatically training times. Several variants may be imagined to derive an appropriate neural network architecture from a pre-constructed decision tree [SE 90, AR 92]. Below we will discuss briefly the hybrid techniques, further described in [WE 93a, WE 94a]; we will provide some practical applications of this particular approach.

6.2.2 A hybrid decision tree - artificial neural network approach for power system security assessment

The hybrid Decision Tree - Artificial Neural Network (DT-ANN) approach aims at combining the advantages of the two approaches while circumventing their weaknesses. DTs are used to yield a first shot, transparent and interpretable model of the relationship between variables representing the states of a power system and its security. The powerful non-linear mapping capacities of multilayer perceptrons are then exploited to augment the discrete classification of the tree with a continuous security margin type of information. This richer information may be used in various ways; in particular, it may contribute to making better decisions during the on-line use of the method.

Such a hybrid approach is schematically shown in Fig. 6.1. Decision trees are first built using a data base composed of preclassified power system states; they identify the relevant test attributes for the security problem of concern, and express, in a hierarchical fashion, their influence on security. Second, this information is reformulated as a four-layer feed-forward multilayer perceptron. Third, the MLP weights are tuned on the basis of the learning set augmented with the security *margin* type of information to enhance classification reliability and transform the *discrete* classification information of the tree into a *continuous* security margin.

Among the possible ways to reformulate a DT as an equivalent neural network, we have tentatively used the one proposed in [SE 90]. It consists of the following four-layer structure [WE 93a].

1. **The INPUT Layer (IL)** contains one neuron per attribute selected and tested by the DT. Their activation levels correspond to the attribute values of the presented state.
2. **The TEST layer (TL)** contains one neuron per DT test node. Each TL neuron is linked to the IL neuron corresponding to the tested attribute.
3. **The ANDing layer (AL)** contains one neuron per DT terminal node. Each AL neuron is connected to the TL neurons corresponding to the test nodes located on the path from the top node towards the terminal node. Its activation level is high only if the state is directed to the corresponding terminal node of the DT.
4. **The ORing layer (OL)** contains one neuron per DT class, connected to the AL

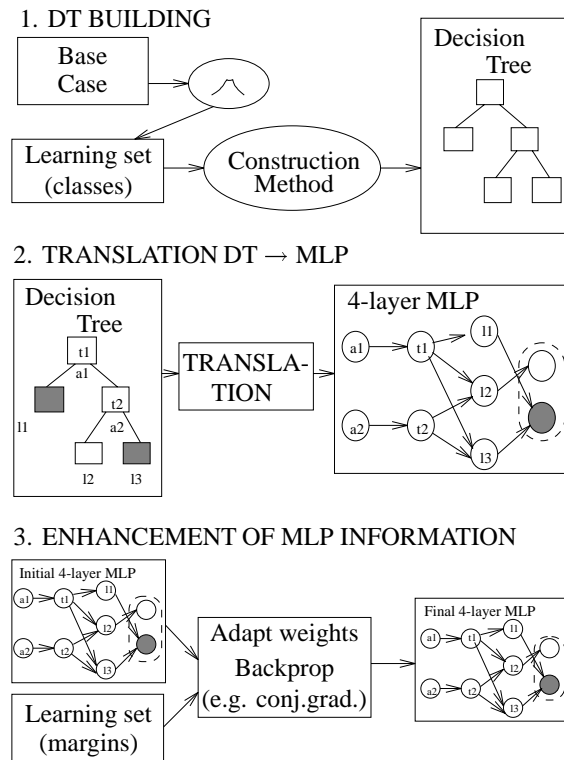


Figure 6.1 Hybrid DT-ANN approach

neurons corresponding to the DT terminal nodes where the class is the majority class. Its activation is high, if at least one of the AL neurons is active.

In order to replicate exactly the classification of the DT, sharp activation functions must be used, to make the transition from -1 to 1 sufficiently sharp, when a state crosses the security boundary defined by the DT.

If the network is used to approximate a continuous security margin, rather than to merely classify, some modifications are required. First, the output layer would be replaced by a single output neuron, fully connected to all neurons of the AL. In addition, since the weights as given by the DT translation are not necessarily appropriate, it relies on learning to choose them correctly. To obtain a smooth, easily adaptable input/output mapping a rather smooth transition function is used.

However, in order to obtain meaningful results, and in particular to avoid overfitting problems, it is important to take care about the normalization and truncation of the margin before the back-propagation algorithm is used to adapt the weights of the ANN.

This is because the attributes used to formulate the decision tree may not be sufficiently informative to determine the margin when the latter is much smaller or much larger than the threshold used to define the security boundary. Thus this kind of approximate margin information will essentially be valid only *locally* around the security boundary.

6.3 MACHINE LEARNING AND DISTANCE COMPUTATIONS

The multi-layer feed-forward perceptron may be seen as an implicit way of defining a distance in the attribute space. In the above hybrid approach this distance is used to replicate at the output of the neural network a distance to the security boundary defined by a decision tree in the attribute subspace corresponding to its test attributes.

As was shown, the weights may be further adapted so as to fit the corresponding metric to a predefined security margin, in the *vicinity* of the security boundary. This is based on the conjecture that the attributes which allow us to predict the security *class* with a sufficiently high reliability should also contain sufficient information to predict the *value* of the security margin, nearby the security boundary. This conjecture was verified for different types of security margins in many simulations on simple test systems and also on some real large-scale systems.

The advantage of using the implicit metric of the multi-layer perceptron is that the back-propagation algorithm provides an effective and at the same time very flexible - though time consuming - method to adapt this metric to the problem specifics, on the basis of information contained in a learning set. Below we will discuss the possible advantage of using classical distance computations in the attributes space defined by a decision tree. In addition to providing the distance to the classification boundary of a tree, this kind of distance may also be used to compute the similarity between states on the basis of their location in the attribute space, which may for instance be used in a nearest neighbor kind of classifier.

6.3.1 Margin regression

A conceptually quite similar idea to the above hybrid DT-ANN approach was first proposed in [WE 88]. This approach is based on the definition of a distance in the attribute space, in terms of weighted attributes used in a decision tree. Thus the distance is of the following form

$$\Delta(o_1, o_2) \triangleq \sqrt[k]{\sum_{i=1, \dots, n} w_i |a_i(o_1) - a_i(o_2)|^k}, \quad (6.1)$$

and the weights w_i and order k are adjusted on the basis of the learning set to correlate the latter distance as strongly as possible with a predefined security margin. In particular,

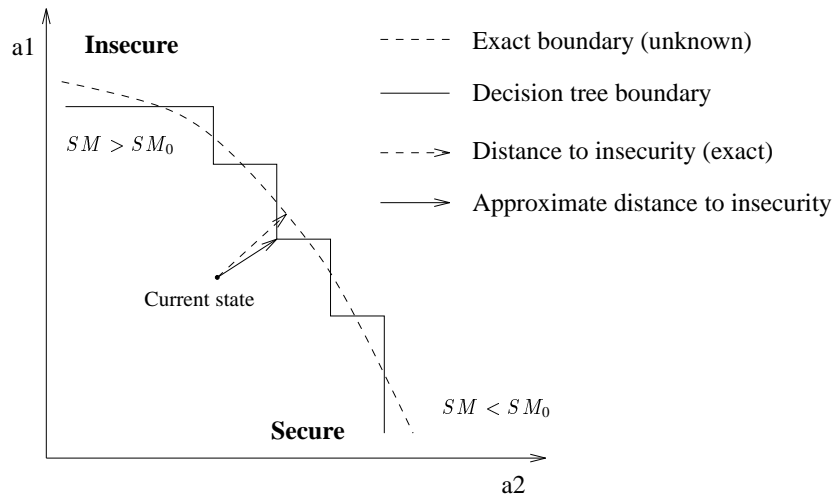


Figure 6.2 Illustration of distance computations in the attribute space

$k = 1$ yields the Manhattan (or city-block) distance which would correspond to a linear approximation of the security margin in terms of the attribute values, near the security boundary.

This idea is illustrated in Fig. 6.2, where the security boundary is supposed to be defined with respect to a security margin SM and a given threshold SM_0 .

The distance from a location on the secure side to the insecure region, is approximated by the distance to the region covered by the terminal nodes of a decision tree, where a majority of unstable learning states are recorded. The higher this distance the more secure the state; thus monitoring the variation of this kind of distance will allow one to identify whether the system drift moves its operating point closer to insecurity or not. Similarly, the distance from a state on the insecure side to the secure region allows one to assess its degree of insecurity and may provide a quick indication of how to modify its operating point (its attribute values) so as to move towards the secure region. Since a decision tree decomposes the security region into a union of hyperboxes defined by simple constraints, the computation of the distance is almost trivial.

As indicated above, one of the major problems is the appropriate choice of weights to combine the different attribute values in the distance, which may correspond to different physical quantities such as powers, voltages and even topological indicators. The approach taken in ref. [WE 88] was to consider that the weights would be defined either a priori on the basis of pragmatic considerations, or they should be adapted on the basis of the sensitivity of the “benchmark” security margin with respect to the attributes used. In particular, in this reference we proposed to use an iterative numerical

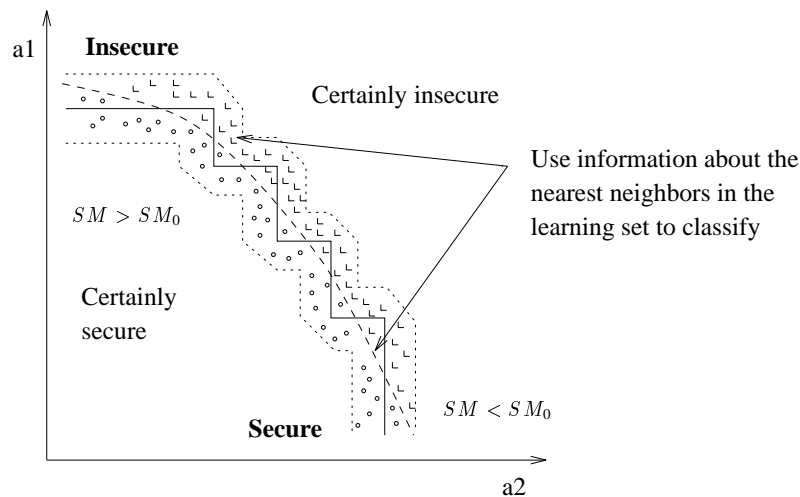


Figure 6.3 Uses of distance computations in the attribute space

technique to adapt the weights on the basis of the precomputed margins of the learning states, so as to maximize the correlation coefficient of the distance and the margin.

This approach yielded reasonably satisfactory results for several transient stability problems; nevertheless further research would be needed in order to develop a systematic and robust optimization technique.

6.3.2 Nearest neighbor

Once an appropriate distance has been defined in the attribute space, one may think of various interesting possibilities. For example, Fig. 6.3 suggests a straightforward way to define a buffer region across the approximate security boundary where more refined information should be used to classify a state. In particular, one may determine an appropriate distance threshold and define the buffer region as the part of the attribute space where the distance to the boundary is smaller than this threshold. This is illustrated in Fig. 6.3, where a Manhattan distance was used hypothetically. Such an approach may allow us to identify those states for which there is a high risk of misclassification.

Further, within the above buffer region around the security boundary, we may use the nearest neighbor classifier and exploit the same distance to identify the specific learning state belonging to this region which is most similar to the current state.

Of course, as soon as a reasonable distance has been defined, many other possibilities

may also be thought of. For example, suppose that a state is considered to be probably insecure, either because it falls into the insecure region where the decision tree classification is deemed to be reliable, or because some of its nearest neighbors in the attribute space are insecure. Then the distance may be used to find a set of reference situations in the data base which are secure, and which are as similar as possible to the current state. These reference situations may then be shown to the operator as a set of alternative controls, and the one satisfying feasibility and economy constraints may be chosen as a new operating point. This opens a broad range of possibilities, in particular to provide very quickly preventive or corrective actions which may then be applied and leave further time for more sophisticated techniques to determine a better new state.

6.4 DISCUSSION

Comparing the above two approaches we note that their main difference lies in the fact that the DT-ANN hybrid approach provides a more systematic, and at the current stage of research, more effective means to adapt the weights in order to fit automatically the output to the desired security margin. A certain number of experimental results show that this technique is quite effective. However, the obtained information is useful only locally, nearby the security boundary and cannot provide distances among individual states.

On the other hand, the explicit distance computations in the attribute space may be easier to interpret and closer to the human way of thinking. They therefore offer a promising research direction to exploit more systematically the information contained in the generated data bases. At the present stage, several results have already been obtained on various power system problems. For example it was found possible to determine the coefficients of the distance so as to provide a good approximation of the security margin, and to use this distance to identify states too close to the security boundary to be classified reliably with a decision tree [WE 90a]. On the other hand, using the attributes identified by a decision tree allowed us to improve systematically the quality of the nearest neighbor classifier, and in some circumstances to reach and even exceed the performances of the decision trees, while the standard $K - NN$ method using all candidate attributes got very poor results.

At the present stage of the research, the main difficulty is the lack of systematic and robust techniques to determine the appropriate weights of the distance. Such a technique could possibly be based on some of the more recent heuristic optimization methods such as those described in §3.5.3; it could also take advantage of the information quantity provided by the various test attributes of a decision tree; admittedly, this is a prerequisite to the systematic use of these methods in the context of security assessment problems.

7

Comparing supervised learning methods

In this chapter we attempt to give a synthetic overview of salient characteristics of the *supervised* learning methods presented so far. Our purpose is not to suggest that one particular kind of method would be more appropriate than others. Rather, we start from the premise that almost every method may be useful within some restricted context, and summarize the respective strengths and limitations of the various methods so as to highlight their complementary possibilities.

To simplify, we will only consider the main more or less “stand-alone” techniques, leaving aside the auxiliary tools, such as genetic algorithms and linear models. On the other hand, the discussion and comparison of unsupervised learning methods is rather uneasy, in particular due to their empirical character.

We will first consider important practical criteria - computational and functional - which should be taken into account when comparing supervised learning methods, and briefly comment on the proper evaluation methodologies of these criteria. We will accordingly distinguish three important classes of supervised learning methods : rule based, smooth function mapping based and memory based. Further, we will indicate interesting algorithms from each category, before summarizing their main characteristics.

Finally, we will briefly review some important comparative studies and in particular the Statlog project, from which several results have been quoted in the preceding chapters.

7.1 CRITERIA

Among various criteria for comparing methods for supervised learning, we will consider computational aspects and functional description in terms of the types and quality of the security information which may be derived from a method. We will also point out some aspects related to the evaluation methodologies which should be used in practice.

7.1.1 Computational criteria

These criteria concern the computing time and memory requirements of the methods during the off-line *learning* phase and the on-line use of a method for *prediction* of unseen cases. Of course, for a given method depending on the algorithms and the software implementation, there may be various compromises among these two aspects. In particular, the appropriate use of parallel computation might change considerably the relative positioning of the different methods. Also, the computational requirements depend, in general, strongly on the problem size. For example, at the learning stage the product $n \times N$ of the number of attributes by the number of learning states may be used as the problem size, while at the prediction stage the complexity of the learned model would be useful, which depends implicitly on the above two numbers.

We have already illustrated in various examples that the methods which are slow during the learning phase (e.g. the multi-layer perceptron) may be very fast in the prediction stage. On the other hand, the nearest neighbor type of methods are in general quite fast during the learning phase, but, compared to other methods, they are really slow and require large amounts of memory during the prediction stage.

In the sequel we will give some indications about the relative computational performances of the different methods, in the context of an assumed realistic problem size for power system security assessment, corresponding to a product $n \times N \in [10^5 \dots 10^6]$ and a model complexity adapted to this problem size.

7.1.2 Functional criteria

Under the category of functional criteria, we group all the non-computational criteria, concerning the type and quality of information provided by the methods, both at the learning and at the prediction stage. In particular, these criteria include *accuracy*.

Accuracy will of course strongly depend on practical problem features, such as the type of security information sought, the number, type and distributions of attributes, and last but not least the learning set size. In the context of security assessment we have found that the relative accuracies of various methods depend on the physical problem (e.g. preventive transient stability assessment vs. voltage security emergency

state detection), and of course on the types of candidate attributes used. In particular, some problems are rather local and are therefore easily handled by the decision tree methods, while others tend to be more diffuse, calling for the combination of a higher number of elementary attributes, which is easier to do with an approach like multi-layer perceptrons or the projection pursuit technique.

Some methods are able to provide important data analysis and explanatory information at the *learning* stage, allowing us to identify the important attributes and the physical relationships among them and the output information, thereby providing a good general summary of the data base information. Other methods are able to identify the closest reference case at the *prediction* stage, and may thus provide case by case justifications for their predictions.

Finally, some other methods are unable to provide any explanatory information at the learning or prediction stage, but provide the possibility of modelling numerical output information as a smooth function of its input attributes. This kind of method may be particularly useful to approximate security *margins*, and provide *sensitivity* calculations of the predicted margin with respect to input attribute values.

7.1.3 Evaluation methodologies

Often, comparisons among methods have led to rather useless results due to a lack of rigor in the evaluation methodology. Below we will give some very straightforward but important tips to help in making a honest comparative assessment of methods, both from the accuracy and the computational points of view. Our discussion focuses on power system security problems, but most of the considerations remain true in general.

Simulated data sets

We first discuss the use of simulated data sets. Indeed, in the context of power system security assessment generally the data sets are obtained by *generating* a random distribution of states for a power system model and applying various calculations to obtain the attribute values and security characterization.

This is further discussed extensively in the following chapters, but it is important to notice that with *simulated* data sets, correlations are sometimes unduly - and unexpectedly - introduced among some variables due to particular modelling simplifications. Some examples of these kind of correlations will be illustrated later, in the chapters reporting on practical results. For the time being, let us consider a simple “imaginary” example.

Let us suppose that we are considering security assessment of a power system, and that we have generated a data base obtained from various load levels. We assume,

that the states are generated by keeping a constant geographical load distribution and power factors and that the generation pattern is adjusted to the load level via an optimal power flow module, or any other deterministic procedure used to simulate operation strategies. Thus, if the topology and voltage set-points are not varying independently of the load level, the operating points lie on a one-dimensional subspace of the attribute space. Moreover, it would not be surprising if the security margin was decreasing for increasing load levels. However, using power flows and/or voltages as attributes may apparently render the discrimination more difficult, at least for some methods (e.g. like the nearest neighbor rules) which are sensitive to redundancy and normalization. On the other hand, other methods could recover the one-dimensional load level information from the given attributes, by approximating the inverse mapping. These latter methods would then appear to be significantly superior to the former methods.

Of course, in real life the load distribution may vary as well as its power factor, but more importantly the security criteria would be used to assess situations for which the generation distribution would not correspond to the above deterministic rule. Thus it would not be a good idea to subjugate completely the generation pattern to the load level in the training and test sets.

We now temporarily close the discussion of the consequences of using simulated data sets until chapter 11.

Accuracy concerns

The assessment of accuracy is certainly the primary concern in the context of supervised learning, and even more in the context of its application to power system security problems. In general, we would like to be able to obtain a criterion neither overoptimistic, which might lead to non-detections of dangerous situations, nor overpessimistic, which would lead to overconservative control policies, and corresponding economic costs.

Of course, we know that *perfect* criteria are an illusion, particularly in the context of learning approaches. Thus, it is of paramount importance to be able to assess the accuracy or reliability in practical situations. The first requirement should be to use a sufficiently large test set composed of independent states. By sufficiently large, we mean about 1000 test states or more, so as to reduce the standard deviation of test set error rates to less than 1% (according to eqn. (2.47)). Ideally the test states should also be independent of the sampling assumptions made to generate the learning set. For example, they should include states derived from data recorded in the field, modified randomly so as to create various secure and insecure data sets. If it is not possible to obtain data from the field, as is unfortunately the case at the research stage, an appropriate approach consists of using the same sampling procedure as for the learning states.

In practice the complete data base is merely divided into a randomly chosen test set (say

of 1000 states), put aside and used *only* for the evaluation of accuracy. The remaining states may then be used as learning sets and cross-validation sets to select appropriate classifiers or regression models. Notice that it is important to use the *same* test set to compare various methods. Note also, that there is no valid excuse for using too small test sets, since in the context of power system problems there is in principle no difficulty in generating a large enough data base when starting a research project. Once the software has been developed for the random sampling and the computation of attribute values and security information, it is merely a question of CPU time. Since the learning set will at least contain several hundreds of states, generating a large enough test set will at most multiply the CPU time required for the data base generation by a factor of 2.

A second aspect in evaluating security assessment methodologies consists of distinguishing among various categories of errors. For example, if security margins are available, which is often the case, at least three categories of errors should be defined : *normal* errors (i.e. consisting of small deviations in terms of the margin), *dangerous* errors (i.e. highly optimistic diagnostics), *false alarms* (i.e. pessimistic diagnostics).

Finally, it is important to realize that the learned models may depend quite strongly on the random nature of their learning set. In particular, in addition to the uncertainty of the test set error estimates due to finite test set size, there is an additional chance factor due to the finite size of the learning set. For example, in our experiments in transient stability and voltage security, we have found that this may be responsible for relative variations of more than 10% in the test error rates. While using very large learning sets could allow us to reduce this randomness, in practical large-scale system security assessment environments, computational resources available for the generation of the data base generally constrain its size (see the discussion in the next section). Thus, while it may be *theoretically* interesting to study the asymptotic behavior of a method, from the practical point of view there is little interest in simulations considering learning set sizes larger than say 500 times the number of independent attributes used to characterize the power system states. Within this bound, it may be interesting to construct learning curves (or surfaces) with various methods in order to assess the effect of the learning set size and the attributes on the resulting accuracy.

Finally, an important bias in comparative studies may be due to the highly variable degree of expertise of the authors in the different methods they try to compare. Often, researchers compare their own favorite algorithm, for which they are presumably expert, with a set of “competing” methods, which they discover while doing the comparative study. For this reason, the compared algorithms often represent the state of the art only for the favorite method, and under such conditions highly biased conclusions may be reached.

The very large diversity of methods makes it difficult to obtain honest comparisons, and this is the main reason why this kind of comparison has started only recently, in

particular with power system security problems. Within this context, we have provided our data sets to the research teams involved in the Statlog ESPRIT project (see below, §7.3.1), which offers the guarantee of an unbiased assessment, as much as possible.

Computational performances

In addition to accuracy, computational performances are also very important, and should be assessed in order to evaluate the relative ease of experimenting with a method using various sets of parameters.

Clearly however, most of the software packages used at an early stage of a research project are quite suboptimal in terms of computational efficiency and it is often possible that an order of magnitude of speed improvement may be obtained.

Another aspect which may render the assessment of computational performances difficult, is related to manual tuning which is required with many heuristic methods and which may influence quite strongly the resulting performances. Often the best (and also the least) one can do is to acknowledge the fact that there is such a tuning stage, and to indicate the amount of time it took in practice to adjust the parameters to the particular problem at hand.

In addition, at the present time, computer architectures are changing rather quickly and the relative speed of the various methods' implementations may strongly depend on the computer architecture, such as fast floating point units or size of high speed cache memory, and compiler facilities like parallelization and other optimizations.

Finally, while in many methods (e.g. the decision trees or the multi-layer perceptrons) the constraining computational requirement is related to the learning stage, with other methods (e.g. nearest neighbor, kernel density estimation) the prediction stage may be much more constraining in practice. So, both aspects must be assessed carefully.

7.2 SURVEY OF METHODS

Below we provide a summary of the main characteristics of the different supervised learning methods selected for further consideration. Of course, our judgement cannot be free from subjectiveness and is limited in scope to power system security problems. However, factual foundations of our assessment are given in the chapters of Part 3, relating to applications of various methods to a variety of security problems of real-life and academic systems. Our presentation is also influenced by the Statlog project, which appears to be in good agreement with our own results obtained independently.

7.2.1 Three classes of methods

Before providing a synthetic description, we will classify the supervised learning methods into three categories, according to the possible uses that they may provide in the context of security assessment.

Rule based

To this class belong methods, like the decision tree and rule induction methods, which are able to provide the model they have learned in the form of explicit, more or less global rules, expressing in an easily understandable fashion the information they have extracted from a learning set. To each rule corresponds a set of conditions on the attribute values, which correspond to an elementary region of the attribute space.

This precludes us, in practice, to represent information about continuously varying security margins in a continuous model. Rather, it is necessary to *discretize* information : security margins must be quantized into a small number of security *classes* and models are expressed as discretizing the attribute space into a rather small number of regions of “constant” security.

The price of discretizing is loss of information together with a certain degree of approximation. These methods may however be very competitive with more complex techniques, provided that the complexity of the problem is not too high, and in particular that it is possible to provide a reasonable approximation of the security classes with a small (say less than 100) number of regions.

Of course, it is possible to derive continuous models from the box type description, for example by using distance computations [WE 88] or interpolation techniques [CA 87] or using the hybrid techniques discussed in the preceding chapter.

Smooth function mapping based

This class of models, such as the projection pursuit technique or the multi-layer perceptron, are based on the regression approach to supervised learning. They are able to approximate security margins by a continuous input/output mapping, thereby offering possibilities such as sensitivity analysis and control.

On the other side of the coin, we find mainly the absence of understandability of the resulting models, in particular in the case of high-dimensional input spaces.

Of course, using a reduced set of attributes and projecting the multidimensional model on this reduced space, may allow us to have a closer look at its input/output relationship, and provide some interpretation. This would however call for another method, e.g. of the preceding category, to suggest interesting combinations of variables to look at.

Memory based

In contrast to the two preceding approaches, which translate the initial learning set into a synthetic model which is self-sufficient for later prediction tasks, the memory based methods require the explicit storage of the learning states and exploit these for prediction by identifying in a case by case fashion the most relevant encountered states. This class contains the instance based learning methods from machine learning and the statistical nearest neighbor and kernel density methods.

In power system applications, the main advantage we may anticipate for such methods is that they would allow some refined local reasoning capabilities, and provide justifications to the operator in the form of validated reference cases. Additional human expertise might then be used in order to question the validity of the extrapolations. Reject options may thus be implemented on the basis of the differences observed between the current situation and its nearest neighbor in the data base. For example, if a very unusual topology is encountered, which was not represented in the learning set and if the nearest neighbor state has a very different topology, then either a conservative bound may be derived on the security margins or the state may be rejected as impossible to analyze by analogy with the learning set cases, the latter being too different. Another interesting possibility is for the validation of control actions. If a state is not sufficiently stable, then we may search in the stable subset of the data base for the nearest neighbor, in terms of control distance.

While strong in local reasoning, the nearest neighbor approach is unable to provide directly the required global information and a simple iterative approach could become cumbersome due to the computational costs of searching large data bases.

Of course, either of the two preceding approaches may provide the required global information to render the search of large data bases more efficient. In particular, we have mentioned in the preceding chapter that a hybrid DT-NN approach may use the partition provided by a DT in order to directly guide the search towards the right region of the attribute space. This may lead to improvements in terms of computational speed of one or two orders of magnitude.

Figure 7.1 gives a pictorial representation of the classes of learning methods, and their associated characteristics, which are assessed more precisely in the next section in the context of power system security.

7.2.2 Synthetic comparison

Here it is important to insist on the fact that the evaluation may significantly change from one problem to another. We report on our own experience, in the context of power system security problems, which is however well confirmed by results obtained

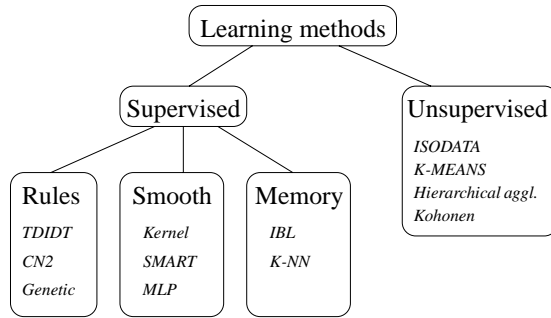


Figure 7.1 Different classes of learning methods

Table 7.1 Synthetic characterization of supervised learning methods (see text for explanation)

Method	Accuracy					Computational (SUN Sparc2)		Functionalities					
	VSESD		PTSA			Tot	Learn (sec)	Predict (sec)	I	M	L	R	S
Rule	0	0	+	-	0	0	$10^3 - 10^4$	$10^{-4} - 10^{-3}$	Y	P	N	P	P
Smooth	++	++	+	+	+	+	$10^5 - 10^6$	$10^{-3} - 10^{-2}$	N	Y	N	Y	Y
Memory	+	--	--	0	--	-	$10^2 - 10^3$	$10^{-1} - 10^0$	Y	Y	Y	Y	N

by other researchers and in particular in the Statlog project.

Table 7.1 provides a summary of main features of the methods from the three above classes. For the accuracy assessment we reproduce rankings for five different problems. The first two problems correspond to voltage security emergency state detection (VSESD) described in §§14.2, 14.4, and the three other problems correspond to examples discussed in §§13.3, 13.4 on preventive wise transient stability assessment (PTSA).

For each problem, several trials have been made for each class, corresponding to different parameters and methods. For the rule based techniques we give results corresponding to various tree induction methods. For the smooth function approximation techniques results are obtained either from the projection pursuit technique SMART or from the multi-layer perceptron. Finally, for the memory based methods we have used the $K - NN$ method, adjusting the value of K to obtain optimal results and using various lists of candidate attributes appropriately pre-whitened; in particular, those selected by the decision tree building procedure provided, in general, significantly better results than the initially proposed attributes.

In Table 7.1, the class of methods obtaining the best result is marked + (or ++ if this result is significantly better than the others), the one obtaining the worst result is marked - (or --) and the one obtaining intermediate results is marked (-, 0, +) as appropriate. The last column provides the mean accuracy ranking of the method. The next two columns indicate computational requirements in terms of an interval corresponding to

computing times in seconds required for learning a model and for using it for making one prediction. These numbers, while purely indicative, are scaled in seconds CPU time on a 28 MIPS SUN SPARC2 workstation and correspond to a problem size corresponding to our example transient stability problem of §3.4.1.

Finally, for each kind of method we have indicated its functional possibilities, in terms of interpretability (I), margin computations (M), locality of reasoning (L), reject options (R) and sensitivity computations (S). We use the following abbreviations : Y to denote a functionality which is definitely there, N to indicate its absence, and P to distinguish those cases where a functionality may be possible via some adaptations.

7.3 RESEARCH PROJECTS

We take the opportunity to discuss (very briefly) some of the research projects which have aimed at comparing various learning methods. It appears that many of the published studies happen either to be of a rather limited scope or to suffer from some of the pitfalls we mentioned earlier.

In the context of power system security assessment, no valuable comparative studies have been published so far, involving state of the art methods from all three classes of machine learning, statistical and neural network approaches. This is mainly because up to recently research was still at the level of preliminary investigations, considering mostly simulations on academic test systems of small size. We are convinced that the unbiased assessment of the methods requires tests on real systems, in particular of large-scale dimension.

This is justified by the fact that the learning problems become really difficult only if the security problem considered is sufficiently complex, corresponding to variable topologies and large-scale effects. We will discuss this in more depth in the next few chapters. Here we will merely point out the sound comparative study of the Statlog project and give some further references to some of the best known comparative studies available in the literature.

7.3.1 Description of the Statlog project

Goals

The main goal of the Statlog project was to break down the divisions among different disciplines of machine learning, statistics and neural networks, which hindered a systematic high quality comparative review of learning methods.

The project concentrates on supervised learning methods for classification problems,

and the first goal was to provide a critical performance assessment of presently available methods and indicate the nature and scope of further developments required by some particular methods to meet the expectations of industrial users.

Methods

More than 20 different methods have been compared, including the standard and modern statistical techniques, various decision tree and rule learning methods and various neural network based approaches.

Each method was run by a research team appropriately selected so as to offer a high level of expertise in the particular technique considered.

Problems

More than 20 different large-scale problems have been considered, concerning bank credit assignment, image recognition, image segmentation, medical diagnosis, power system security assessment and various other problems.

Most of the data sets are real, i.e. non-simulated data sets.

Conclusions

We strongly recommend the reading of the book corresponding to the final report of the project [TA 94]. In the chapters of part 3 of this thesis, concerning practical applications of power system security problems, we will reproduce and discuss in detail the results obtained for the two corresponding problems.

7.3.2 Other studies

Besides the Statlog project, we mention the study of ref. [AT 90], since it is often quoted and is the only recent work, in addition to our own work reported in [WE 93a], which compares different methods for power system security assessment. This study compared multi-layer perceptrons and decision trees on three problems among which one is a small power system security problem. The authors of [AT 90] conclude that results obtained by both methods are impressive, although their multi-layer perceptrons are slightly better in terms of accuracy. This is a neat comparison, but unfortunately it does not report on any computational aspects, neither does it consider a real or realistic power system problem.

In addition to the above, several more or less serious comparisons have been published comparing decision trees with neural networks [SH 91, MO 89, FI 89].

Finally, the authors of reference [WE 89c] compare a large set of methods, including various statistical techniques, machine learning and neural networks.

Part II

POWER SYSTEM SECURITY PROBLEMS

8

Physical problems

In this second part we will concentrate on essential issues of the application of the learning techniques to power system security assessment. In the last part we will illustrate practical applications mainly in the context of transient stability and voltage security.

8.1 APPLICATIONS OF LEARNING TECHNIQUES

The general principle of the (machine) learning approach to security assessment is synthesized in Fig. 8.1. For a given security problem we may distinguish three steps : (i) data base generation; (ii) statistical analysis and automatic synthesis of security criteria (trees, neural nets, . . .) along with their validation; (iii) use of the criteria to assess security of new incoming situations. The dotted feedback lines in Fig. 8.1 show the iterative nature of the process.

The physical problem statement is considered in this chapter.

The data base generation calls for a random sampling approach and requires in practice the development of an effective tool, which must be tuned to the power system and security problem at hand. This is further discussed in chapter 11.

The statistical analysis step and design of security criteria calls for the proper application of the techniques described in the chapters of the first part. We will illustrate their use later in chapters 13 and 14.

Finally, the way the criteria could be exploited in various planning, operational planning and operation environments will be discussed below in chapters 9 and 10.

In practice, the particular outlook of this learning approach will vary with the physical phenomena considered, the way they are tackled, the particular environment, and the

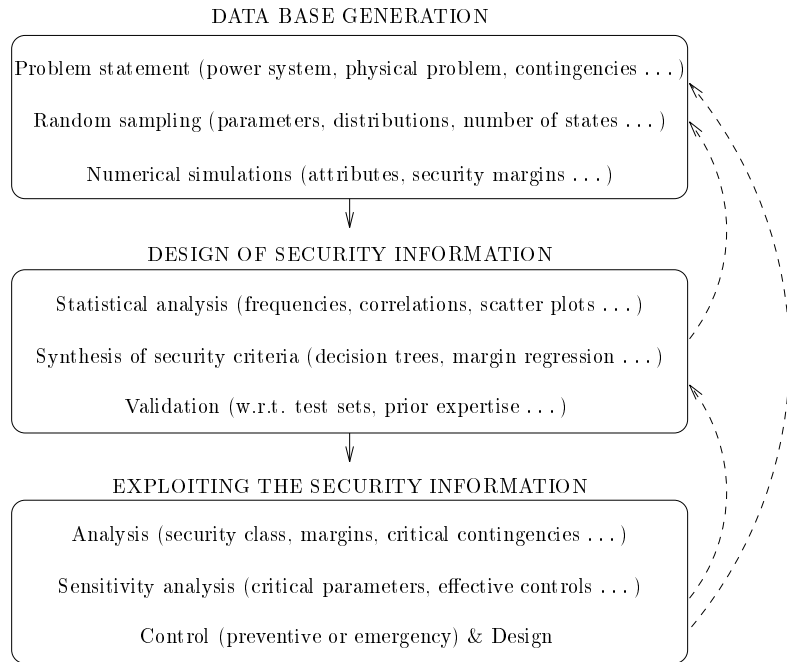


Figure 8.1 *Learning approach to power system security assessment*

practical use which is projected. This is discussed in the next section.

Notice that in many applications the first two steps of Fig. 8.1 are performed off-line, generally in the study environments where classically the security limits are established by experts, while the last step is performed on-line in the control room environment, where operators exploit the security information to run their system.

Generally, in terms of the computer based learning methods four types of possible security diagnostics will be distinguished.

Unknown. The current situation is too different from those which have been considered in the random sampling approach when building the data base for the learning of the security criteria. Thus, there is no possibility of extrapolating the available information. The best is to run a numerical simulation (or an approximate, faster calculation).

Ambiguous. The current situation falls in the domain of validity of the data base, but is too close to the security boundary and we are unable to decide whether it is secure or insecure.

Definitely secure. The current situation falls in the domain of validity of the data base,

and has a very high probability of being secure. Hopefully this is the case in the majority of situations.

Definitely insecure. The current situation falls in the domain of validity of the data base, and has a very high probability of being insecure. The operator must quickly determine preventive control actions or prepare emergency controls to be activated in case the contingency actually happens.

8.2 PHYSICAL PHENOMENA

The considered physical phenomena (e.g. short-term vs mid-term transients) may influence very strongly the way off-line security studies are organized. Also, depending on the structure of the system (radial vs meshed, isolated vs strongly coupled . . .) the way security problems are approached may change significantly.

The time scales corresponding to the considered dynamics influence also strongly operation strategies defining how much control may be done in the context of emergencies and how much should be done in advance, in a preventive approach. For example, slowly developing voltage collapse emergencies may leave enough time for corrective control, while very rapidly developing system wide disturbances, as is the case with transient instabilities, can hardly be corrected in real time with present day technology. They must thus be circumscribed in a preventive security assessment approach, to avoid instabilities with respect to the most probable disturbances, and with appropriate *pre-designed* defense plans to minimize the consequences of instabilities. Below we discuss further these two problems.

8.2.1 Transient (angle) stability

In the following we give a brief discussion of some basics of transient stability assessment. We refer the interested reader to the book [PA 93] and the references it provides for a more in depth discussion of various important topics in transient stability, and an account of research trends in the context of fast transient stability assessment methods.

Basic formulation

Transient stability concerns the dynamic behavior of a power system during the first - say 10 - seconds following major disturbances, such as a three-phase short-circuit on the extra-high voltage (EHV) grid, followed by one or several line and/or generator trippings.

The system is said to be transiently stable with respect to *a particular disturbance*

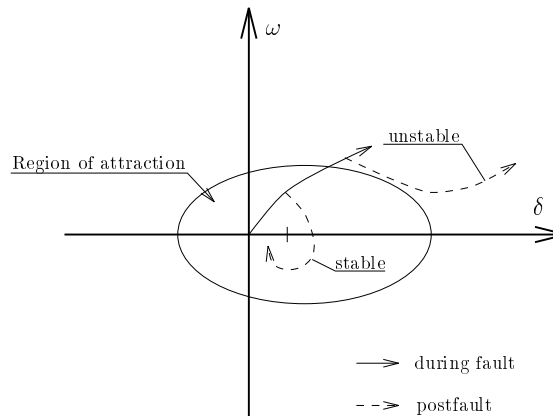


Figure 8.2 *Transient stability behavior : stable vs unstable*

if its dynamic performance during the first seconds following the latter occurrence is “acceptable”. Criteria of acceptable transient behavior depend on the particular utility. For example in some European utilities only the first swing stability is considered explicitly and the system is termed “stable” if no pole slips among any two synchronous generators are observed during the first few (2 or 3) seconds. On the other hand, in most North-American utilities a longer time period of about 10 seconds is considered and the criteria take explicitly into account the stabilization (damping) of EHV voltage and frequency oscillations within acceptable ranges.

Whatever the precise technical criteria, during the period of time considered in transient stability studies the relevant question is mainly whether the system will be able to reach a short term electromechanical equilibrium state in the postfault configuration or not. This will be the case if the system entering its postfault configuration is in the *region of attraction* of an acceptable postfault equilibrium state. This state space stability concept is illustrated in Fig. 8.2 which shows a hypothetical two-dimensional dynamic state space, which, while extremely simplified, captures the essentials of most real-life transient stability problems.

The trajectories in Fig. 8.2 show the important periods of time considered in transient stability studies.

The pre-fault state is the equilibrium in which the system sits at the moment of occurrence of the disturbance, which is a normal synchronous operating condition.

The during fault time period is the very short duration ($\approx 100ms$) starting with the inception of the initiating fault (e.g. a short-circuit) and leading to subsequent protective switching operations (e.g. line tripping, followed by unsuccessful reclosure and retripping). During this period the generators start departing from their

synchronous operation, those being closer to the fault location accelerating more strongly than the others, in general.

The postfault period, will in case of stable behavior result in the system settling down to its new equilibrium state, or in case of instability in possible loss of synchronism and subsequent tripping of some of the generators. In practice, it may take only a few seconds before the system irrevocably loses its synchronism, which leaves only a very short time period available for the possible detection and correction of developing instabilities.

Transient stability is a strongly non-linear problem, and in particular highly fault dependent. The *critical clearing time* (CCT) of a fault is a conventional security margin used to quantify the transient stability with respect to a disturbance. It is the maximum time duration it may take to clear the fault without causing the irrevocable loss of synchronism. If the CCT is larger than the actual fault clearing time the system is actually stable, otherwise it is unstable. Another security margin used in transient stability studies is the so-called energy margin, determined in the context of the direct Lyapunov-like methods [WI 70].

Plant mode vs area mode instabilities

In the context of transient stability studies one may distinguish between two different modes of electromechanical transients.

The first kind of behavior is illustrated in our transient stability studies on the EDF system in §13.3; it concerns *plant* mode instabilities where a power plant located closest to the fault location is endangered by losing synchronism with respect to the remaining system. In this case the limiting parameters concern mainly the active and reactive generation of the considered power plant.

The second kind of situation is the so-called *area* mode instability where a complete subsystem, including several power plants, loses synchronism with respect to the remaining system to which it is loosely connected. Although these latter situations may be more complicated to analyze in practice, it is interesting to notice that a very large majority correspond also to a “two-machine” problem, where one group of machines is in danger of losing synchronism with respect to the *remaining* machines. The limiting parameters in this kind of situation are often the power flows through weak interface tie lines between the two areas.

Whatever the kind of instability, the group of generators losing synchronism is denoted as the *critical cluster* and in practice it turns out that one may study most (if not all) multi-machine situations by considering only the relative motion of the critical cluster with respect to the remaining machines. Thus, stability assessment amounts to identifying the critical cluster and building a two-machine equivalent and from there a

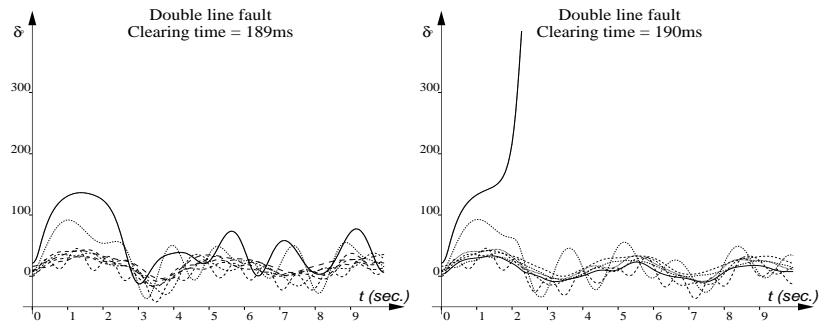


Figure 8.3 Typical marginally stable and unstable swing curves

one machine infinite bus equivalent, which may be studied by the well known equal-area criterion [XU 92]. This is quite an important outcome, since it simplifies greatly the transient stability assessment problem by focusing on the most important physical effects. In particular, the knowledge of the critical cluster may be exploited in order to suggest effective preventive and emergency control actions [OH 86, XU 93c].

Available approaches to transient stability assessment

There are two classes of approaches to transient stability assessment. They both rely on an analytical simulation model of the system appropriately exploited by system theory approaches.

The first is the conventional *time-domain step-by-step simulation* (SBS) technique, which is used in most utilities for off-line studies. The method consists basically of exploiting a mathematical model of the power system dynamics during the considered time span, and a numerical simulation package in order to simulate the during and post fault transients. This yields the so-called swing curves describing the dynamic behavior of the relative motion of the mechanical angles of the machines, the observation of which allows in principle to identify instabilities.

This is illustrated in Fig. 8.3 for a double line fault for the system considered in §13.3. The left part of Fig. 8.3 shows the swing curves of a subset of 8 machines, for a marginally stable situation, corresponding to a clearing time of 189ms. Assuming a clearing time of 190 ms would yield the unstable behavior depicted at the right part of Fig. 8.3.

In terms of accuracy, the SBS method is certainly the benchmark and is used to evaluate the accuracy of other methods. However, in terms of useful security information the technique provides only a very crude YES/NO type of information, and cumbersome repetitive computations are required in order to obtain security margins and sensitivities of these margins with respect to operating parameters assessed in the prefault situation.

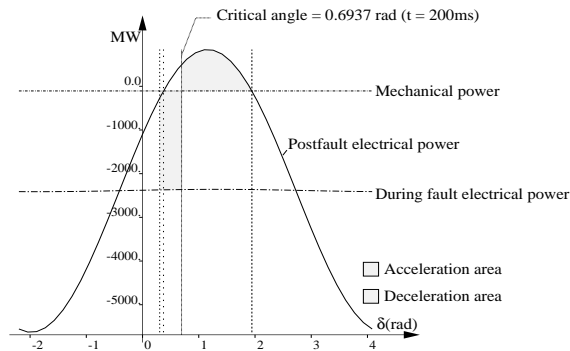


Figure 8.4 Equal-area criterion applied to the critical machines of Fig.8.3

The second class of methods are the so-called *direct Lyapunov* approaches (already mentioned in §1.3.1) which aim essentially at avoiding the lengthy simulation of the postfault transients. The basic principle of these methods consists of characterizing the stability of the postfault equilibrium by an energy function, which is a positive definite scalar function defined in the state space region surrounding the postfault stable equilibrium point, and by approximating the relevant part of the stability region separatrix by a constant energy surface corresponding to a maximal admissible value of the energy function. The assessment of the security may then be done by determining the value of the energy function when the system *enters* its postfault configuration and comparing the latter energy with an appropriate threshold value corresponding to the maximal admissible value of stable states.

In principle, these kind of methods are able to significantly reduce computation time while providing in a one-shot procedure the value of a security margin and sensitivities of the latter with respect to some important parameters [PA 93]. Their main difficulties are related to their simplifying assumptions concerning dynamic modelling of the system. This may require in practice ad hoc adaptations of the method to power system specifics and may lead to tedious validation studies.

As a particular case of direct methods, we mention the *extended equal-area criterion* (EEAC) which is described in [XU 88], and which has been used in some of our preliminary studies [WE 87a, WE 90a]. It is based on the conjecture that transient stability problems may be explained in a satisfactory way by a two-machine aggregated model, further reduced to a one-machine infinite-bus (OMIB) equivalent; an approach for identifying automatically and efficiently the machines belonging to the critical cluster complements the method.

Figure 8.4 shows a graphical representation of the equal-area criterion corresponding to the swing curves represented in Fig. 8.3. The main curve of sinusoidal shape represents the electrical power in the postfault configuration as a function of the mechanical angle δ

of the equivalent OMIB system. The intersections between the upper straight horizontal line representing the equivalent mechanical power and the previous curve define the stable and unstable equilibria of the postfault equivalent OMIB system. The lower, almost flat sinusoidal curve represents the electrical output power of the OMIB system in the during fault period and the difference between the latter and the mechanical power is proportional to the acceleration in the during fault period, the integration of which with respect to δ is proportional to the kinetic energy received in the during fault period by the OMIB system. This is the acceleration area which is depicted in Fig. 8.4.

The fault clearing consists of switching “instantaneously” from the during fault to the postfault electrical power characteristic, which results in a deceleration. It may be shown that if the deceleration area as is shown on the picture is larger than the acceleration area then the system will remain stable, otherwise it will lose synchronism. Thus the difference between these two areas defines the stability energy margin.

Further, the critical clearing angle corresponding to equal acceleration and deceleration areas may be computed easily and therefrom the critical clearing time. In the present case, the critical angle is of 0.6937 radians which corresponds to a critical clearing time of 200ms, which we may compare with the interval of [189 . . . 190] found via the SBS procedure. Notice that this is fairly precise even though the system machines are far from being divided in two coherent groups, as is shown in Fig. 8.3.

In the context of the research on the EDF system reported in §13.3, this method has proven to be an extremely robust and efficient tool for the study of the simplified model [XU 92]. Further, recent research shows promise in adapting the method to cope with the main relevant modelling effects such as fast-valving and voltage regulators.

8.2.2 Voltage security

A majority of recent large-scale system breakdowns have been the consequence of instabilities characterized by sudden voltage collapse phenomena.

The main reason for this are the improvements of protection devices as well as generators speed and voltage regulators and SVCs, which have increased the transient stability limits of power flows, allowing more power to be transferred over longer distances. The reactive compensation problems resulting from higher active power flows and consequently higher reactive losses have led to making the appropriate control of EHV voltage problematic in extreme situations, leading to voltage instabilities which have caused large blackouts.

This has been a major incentive to research in the context of voltage security. The topic being rather recent, there are still many open questions in particular concerning the definition of widely accepted models and corresponding stability criteria. We refer the interested reader to the references [IE 90, NO 91] for a recent overview of the concepts

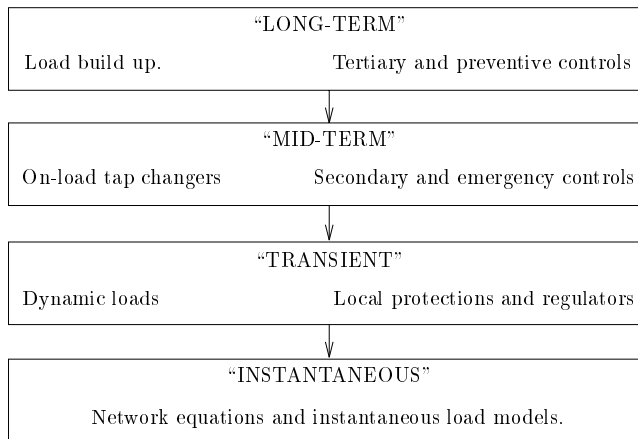


Figure 8.5 Time scales for voltage stability simulations. Adapted from [VA 93b]

and industry experience in this field.

Basic formulation

Voltage security may be defined (loosely) as the ability of a system to maintain its capability of controlling its EHV voltage while submitted to various disturbances, in particular with respect to outages and rapid load build up.

Thus, while transient angle stability is by definition only concerned with a single very short time frame, in the context of voltage security the physical phenomena may be divided into *various* time scales, depending on the physical causes driving the process of voltage collapse.

Figure 8.5 adapted from reference [VA 93b] indicates the four basic time scales which may be involved in the context of voltage stability studies.

The instantaneous network equations consider the quasi steady state equilibrium reached after electromagnetic transients have died out. This leads to a set of algebraic “load flow” equations.

The transient behavior concerns a typical time scale of the first 10 to 20 seconds following a fault. In addition to a risk of angle instability, the system may also be endangered during this period from the voltage collapse point of view, in particular by fast load dynamics, tending to restore very quickly the active power demand after an outage, or by under-voltage induction motors stalling leading to a fast increase in reactive load.

The mid-term voltage instabilities concern the phenomena driven by slower controls

acting in a period of a few minutes following an outage, such as the load restoration process due to the automatic on-load tap changers and the over-excitation limiters of generators.

The long-term behavior concerns the ability of the power system to follow the anticipated increase in demand and takes into account various “tertiary” controls acting in the same time frame of say some tens of minutes.

Short-term vs mid-term vs long-term instabilities

The problem of voltage security is basically related to the existence of a maximum amount of (active and/or reactive) power which may be transferred through the transmission network from the remote generation sites to the load.

Figure 8.6 shows the well-known PV curve illustrating the maximal load transfer capability of an EHV system to a load region by the voltage characteristic of a particular 225kV bus in this region. Whatever the precise meaning of the physical quantities, this kind of curve describes the voltage security problem correctly, at least qualitatively. Due to the non-linearity of reactive transmission losses and due to the upper limits of reactive power generation capability curves of generators and compensating devices, there exists a maximum amount of power which may be delivered to any group of EHV buses.

The difference between the current load level and the maximum value is the load-power margin. In a large-scale power system this quantity may be computed using various hypotheses of the augmentation of individual loads and their correlations. One particular approach consists of computing the margin assuming that the active and reactive load levels are following a direction defined by the real-time observed trend [LE 90a]; another approach consists of computing the direction so as to maximize a given criterion [VA 91a]. In any case the computed margin may be used to assess the vulnerability of the base case power system state with respect to the long term load trend. It may however also be used as a security index to rank contingencies, by computing the value of the load-power margin in the post-contingency situations.

The “long-term” voltage security assessment problem is mainly concerned with the evaluation of load power margins as a function of expected changes in the system within the considered time window. If there are no planned or unforeseen outages, a normal load build up would consist of moving along the PV curve of Fig. 8.6 from A to B, and the task of the operator would be to bring sufficiently soon additional local generation into operation to avoid collapsing at this point. It is important to notice that in practical power systems the point of collapse may be reached with normal values of voltages.

Since we are mainly interested in the security assessment with respect to major equip-

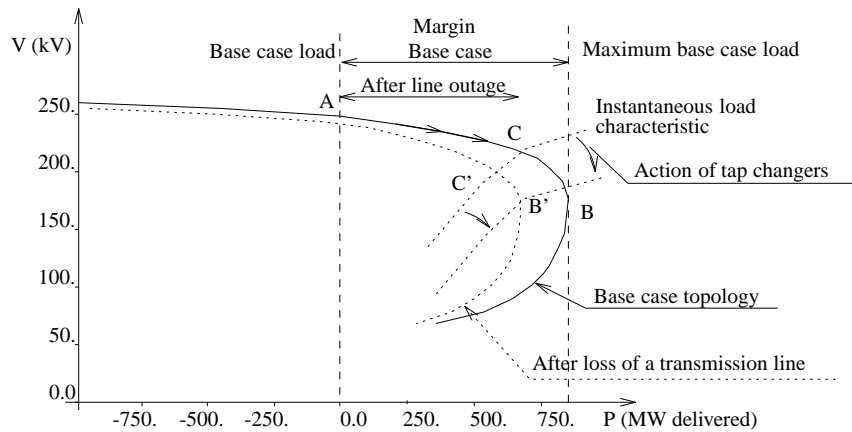


Figure 8.6 Typical EHV PV transmission characteristic

ment outages, we will explain in some more detail the physical phenomena which are of interest in this context.

Let us consider Fig. 8.6 and suppose that at the intermediate point C an important disturbance leads to an EHV line outage. Physically, the system will move along the “instantaneous” load characteristic according to its transient dynamics, from point C to point C'. Due to the sensitivity of load to voltage this results in a drop of load as well as voltage in the HV and MV subsystems. Consequently automatic on-load tap changers (OLTCs) try to increase their transformer ratios in order to restore nominal secondary voltage, which will in turn tend to restore the pre-disturbance load level.

In terms of the PV curve of Fig. 8.6 this consists of shifting the instantaneous load characteristic towards point B'. Figure 8.7 shows the time variation of the voltage nearby the consumers. During normal operation, the tap-changers and the various EHV voltage control loops maintain nominal voltage. Following a major outage the MV voltage level drops consequent to the drop in EHV voltage. In the subsequent stage however, the transformer ratios are changed automatically so as to restore the nominal voltage.

Orders of magnitudes of time scales are as follows

A to B. Typically we would expect a fast load build up to take of the order of half an hour to reach a critical situation. It is the operator's responsibility to monitor the margins in the normal situation and with respect to possible disturbances so as to take appropriate decisions in due time if the margin becomes too small.

C to C'. This transition includes the protective switching, electromechanical transients and action of the first overexcitation limiters, and could typically take between 10

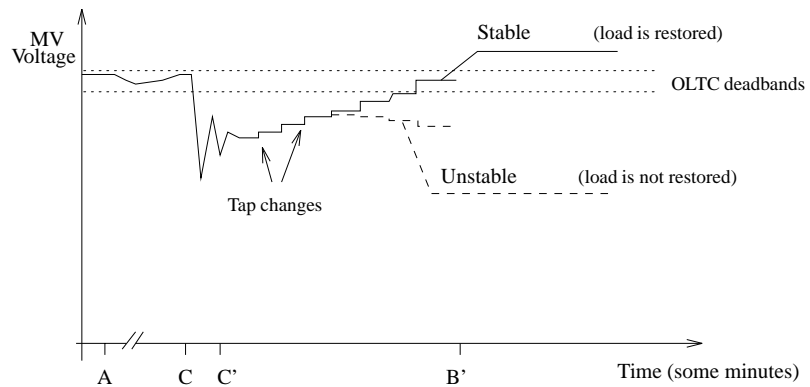


Figure 8.7 Typical evolution of consumer voltages

and 20 seconds. A transient voltage instability could correspond either to the non-existence of point C' or to its location beyond the nose of the curve.

C' to B'. This transition is mainly driven by the automatic OLTCs and may take in practice some minutes. Notice that if the point C corresponds to a pre-disturbance load demand larger than the post-disturbance maximal loadability, this transition will eventually lead to voltage instability. There may be sufficient time to anticipate such a critical evolution and apply appropriate emergency control actions, such as blocking the tap changers, shed some load and/or start up some fast units for local reactive power support.

Conventional approaches for voltage security assessment

Because of the very broad time frame covered by voltage stability related phenomena, it is not surprising that a rather broad range of approaches and tools have been proposed in the literature and are used in practice. Since it is out of the scope of this thesis to discuss all these methods, we will merely describe briefly the three approaches which have been used in the context of the simulations reported in chapter 14. It turns out that these are quite representative of the spectrum of voltage stability analysis tools.

The first method is the purely static *load flow* computation. The aim of this tool was to assess for a given situation whether the corresponding demand level was nearby (possibly beyond) the nose of the PV curve. Closely located multiple load flow solutions may be considered as a good indicator of loss of voltage control and risk of voltage collapse. When this situation arises, conventional load flow computations either have difficulties in converging or may converge towards a highly sensitive solution. In the investigations reported in [WE91c] we have used a load flow software together with the sensitivity computation of total reactive generation with respect to incremental bus

load changes as a voltage security analysis tool [CA 84].

The second tool, used in the context of the simulations on a 7-bus system reported in [VA 91b], consists of a full *dynamic simulation* via an appropriate numerical integration technique, which takes into account the modelling of electromechanical transients as well as the voltage regulators, in addition to the OLTC dynamics. This kind of tool allows us in principle to simulate all the relevant phenomena, and might thus be considered as a benchmark tool.

Finally, the last kind of tool used in the more recent investigations on the EDF system models only the equilibrium equations of the transient dynamics and thus consists of *successive transient equilibrium calculations* following the tap changer and secondary voltage control driven discrete dynamics. This is an intermediate approach between the two former ones; while neglecting the short-term transients it is able to consider the sequential variation of the load and controls which are relevant in this time frame, while being computationally efficient enough to model large-scale systems with a high level of detail at the subtransmission level [VA 93b].

8.3 PROBLEM FORMULATION

In this section we screen important aspects of a range of security problems which may be considered in off-line security studies, and for which it might be appropriate to define one or more learning problems, in terms of : (i) a *universe* of possible power system situations, (ii) security *classes* or *margins*, and (iii) attributes used as predictive input information. In doing so we will not distinguish among the above discussed physical nature of the problem : similar security problems may be defined with respect to either transient or voltage stability, and even with respect to both, at least in principle.

The complexity of the security assessment task of a large-scale power system requires us to decompose it into simpler subproblems, corresponding to the investigation of the influence of a restricted set of parameters on the security in a restricted sense, considering contingencies one by one, or looking at the phenomena observed in a particular region of the overall system. When the security information thus collected is supposed to be exploited in a future operation or study environment it is important to circumscribe the class of situations to which this information may be safely extrapolated. This is particularly true if the process of deriving the security information is more or less automatic, as is the case in the computer based learning frameworks presented in this thesis.

Recall that in a practical situation the security problem definition is generally strongly dependent on the planning and operation practices of a power system as well as the underlying physical problem. Thus, the following discussion merely provides a weak general framework, while the actual solutions are mostly ad hoc, and rely very heavily

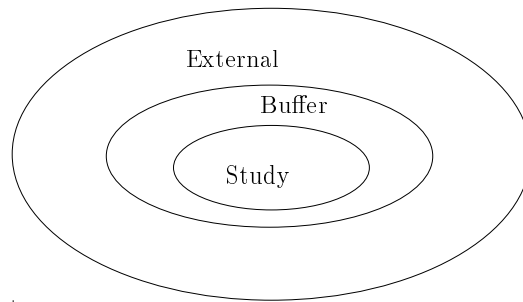


Figure 8.8 *Three level decomposition for security studies*

on the existing expertise of utility engineers.

8.3.1 Prefault power system configurations

In general, we consider the security assessment of a restricted part of the power system under study. For example, in the transient stability problem introduced in §3.4.1 only the aspects concerning the power flow limits of the James' Bay corridor have been considered. In the voltage security studies described in §14.4, the concern is a subregion, weak in terms of voltage security. Similarly, in the study carried out on the EDF system of §13.3, we consider the very specific subproblem of transient stability constraints of an important power plant.

Such a decomposition may rely either on prior physical information about the considered security problem, or less ideally on practical administrative boundaries existing in a power system. In any case, they will lead to at least three levels of representation in the context of a security study, as is represented in Fig. 8.8.

The study region includes all components which are suspected to have a *first order effect* on the security level, with respect to the problem of concern. This will include the elements which may be outaged in contingencies as well as all components whose influence on security is deemed important to assess, and in particular those which may be used as control means to enhance security. When generating a data base all relevant combinations of the corresponding component states should be screened, quite independently of the practical probability of their occurrence in real life.

The buffer region includes the class of components whose status may *influence marginally* the security level in the study region, while it is not desired to use these as control or predictive variables. The corresponding states should be sampled, independently of the study region, so as to represent correctly various possible situations which may happen in reality.

The external region contains the rest of the system, the precise state of which is supposed to be *irrelevant* for the considered security problem. In general the external system state is inherited from the base case and kept essentially unchanged. It may however be a good idea to use two or three different base cases so as to make robust the study results.

It is important to notice that the above decomposition is independent of the mathematical simulation model used to represent devices in the security simulations. For example, although this is certainly not optimal, it is a common practice in the industry to use within a given study context the same level of (rather detailed) modelling for the whole system of the considered utility while representing in a simple (if not simplistic) way the systems of neighboring utilities.

Topologies

The topology of a power system is defined by the transmission lines and EHV transformers in operation as well as the configuration of substation busbars in terms of electrical nodes. These aspects may have a very strong effect on the security level and a particular difficulty is related to the high number of possible topologies which may result from the various combinations of elementary states.

A particular security study might focus on a specific (constant) topology, but in general, in the context of large-scale EHV systems the future topology is not perfectly known at the time when the security study is carried out, and several possible topologies must be considered.

Generally, there is a small number of relevant substations which may operate in either one or two nodes. If these may have a primary effect on the security then they should be sampled independently. Frequently, there exist operation guidelines suggesting which substation configurations should be chosen under particular conditions.

Considering the availability of transmission lines and transformers, a convenient way to define possible topologies consists of considering modifications with respect to a base case, in terms of a set of lines and transformers in operation which may become out of operation or vice versa. Then, various levels of topology variations may be considered, with an increasing number of differences with respect to the base case, and all possible combinations are enumerated independently, excluding the irrelevant ones.

Finally, when generating a data base all the possible topologies are a priori sampled and the data base size should be large enough to screen well enough each important class.

Load / generation / power flow patterns

From the EHV point of view it is often a good approximation to assume that the elementary bus loads in a given region are strongly correlated. In addition, it is generally convenient to assume that the power factor is constant independently of load level at each EHV bus. Thus, the most simple approach would consist of defining an interval of possible regional load levels, sampling the latter interval and distributing the corresponding active and reactive load on individual buses proportionally to the base case values.

However, if the active or reactive load *distribution* may have a non negligible effect on the security level, as may be the case in the context of voltage security, it may be better to combine the above uniform distribution with various random variations of individual loads and power factors.

The generation pattern is in practice strongly coupled to the load level and transmission system topology, due to the operation planning practices. However, in the context of security assessment studies where the purpose is precisely to derive the required information to define these strategies, it is very important to screen various generation patterns, *independently* of the load level and topology. This will include varying the number of available units in the power plants of the study and buffer region as well as their level of active and reactive generations. The external system may be used in order to supply missing active and reactive power.

In the preceding approach, the power flows are a consequence of the independent choice of topology, load and generation patterns. This may lead to unrealistic or inappropriate distributions of power flows. An alternative approach could be to choose the power flows independently and to define load and generation patterns so as to comply with these flows. This kind of strategy may be used in the context of transient stability studies of radial systems such as the one described in §3.4.1.

8.3.2 Classes of contingencies

In addition to the range of power system configurations, an important parameter is the kind of contingencies with respect to which security is evaluated. This depends again very strongly on utility specific practices and on the physical characteristics of the considered power system.

Initially, most of the pattern recognition and machine learning studies concerned preventive security assessment considering one contingency at the same time. This was mainly motivated by the highly non-linear characteristics of most security problems with respect to large disturbances, which makes the security region strongly contingency dependent. By allowing us to exploit more efficiently the local nature of security

constraints, this contingency by contingency approach yields indeed simpler problems, for which it is often easier to derive accurate contingency specific security information.

On the other hand, our recent multi-contingency investigations in the context of both transient and voltage stability have shown the interest in carrying out systematic studies, screening in parallel a broad panel of contingencies for all relevant power system configurations. This allows us for instance to systematically compare the relative strengths of contingencies and to identify critical contingencies and classes of similar contingencies. It may also provide feedback information to improve the protection system for example by reducing clearing times for the most critical faults.

In general, a security study should screen all the relevant contingencies corresponding to a given problem. Until now, most of the security studies in the context of operation planning used a manual approach selecting interesting scenarios combining the choice of interesting power system configurations and the selection of the potentially most dangerous contingencies, on the ground of prior knowledge and intuition of the expert. Although in some particular situations it may be easy to identify the most critical contingencies, given enough available computing power it may be preferable to simulate systematically every contingency for every considered power system configuration, since the only price to pay for this richer information is in terms of CPU time.

8.3.3 Learning problems

In the context of power system security assessment, a learning problem is defined by a set of possible power system states which are classified as secure and insecure with respect to various possible contingencies or described by security margins.

In addition, it is interesting to distinguish between preventive and emergency state security assessment.

Attributes

In the *preventive security assessment*, the considered power system states are normal prefault situations independent of the contingency. The security is assessed with respect to a list of hypothetical contingencies. The attributes are variables which thus essentially characterize the prefault system state, and which are likely to provide security criteria. It is important to distinguish among various kinds of attributes, such as *controllable*, *directly observable*, and complex *ad hoc* attributes. The type of attributes chosen in practice depends on the particular compromise between interpretability / accuracy / robustness which is sought. For a given security problem it may be interesting to consider various such compromises and derive the corresponding criteria. Notice that the time constraint is not very restrictive in the context of preventive security assessment, as far as the computation of complex attributes is concerned.

In the *emergency-wise security assessment*, the considered power system states correspond actually to post-contingency situations. For example, in the context of transient stability these states would typically be snapshots of dynamic - non-equilibrium - states during a period of 100 to 200 ms after fault clearing. Within this context timing uncertainties may become an important factor rendering the available measurements less useful. In the context of mid-term voltage stability, on the other hand, the considered situations would correspond typically to the pseudo-equilibrium states reached after the short-term transients have died out. In this case uncertainties concerning the load model may make the interpretation of measurements ambiguous. The attributes used to characterize the power system states may be of three types : (i) system measurements collected in real-time, possibly filtered by a fast state estimation triggered upon the fault occurrence, if sufficient time is available as for example in the case of mid-term voltage security assessment; (ii) information obtained from the protection system allowing to identify the disturbance; (iii) stored precomputed information obtained from the pre-fault system state. Notice that, depending on the time frame only a subset of these attributes may be actually available in real-time, for a given practical power system; but in the future we can imagine faster information systems and better real-time processing capabilities, allowing to use more and more sophisticated information. These considerations will be clarified further in the later chapters by means of some practical case studies in the context of both preventive and emergency wise security assessment.

Security information

To define a learning problem we need to choose a particular encoding of security information in the form of either discrete classes or continuous security margins. Many different ways of encoding this information may be thought of more or less adapted to various learning techniques.

As concerning security margins, we already mentioned that it is possible to define continuous margins which allow us to quantify the degree of security, in most security problems. If a classification model is sought, e.g. a decision tree, then the classes can be defined with respect to one or more thresholds on the security margin. For example, in the context of transient stability assessment a conventional margin is the critical clearing time, and a state could be classified as secure if its critical clearing time is larger than the upper bound of the actual clearing time. In the context of voltage security assessment, we may use the load power margin as an appropriate indicator of the distance to insecurity, and we would consider a system as secure with respect to a particular disturbance, if this state evolves to an acceptable mid-term equilibrium in the post disturbance configuration and if the latter state has a large enough load power margin to allow some plausible safe increase in the load level during the minutes following the incident.

As concerning the number of contingencies tackled simultaneously, we already men-

tioned some pros and cons of the single- vs the multi-contingency approach. If we consider the single-contingency approach, we must formulate a number of elementary learning problems covering the pre-established list of relevant contingencies.

If we consider multi-contingency criteria there are various possible approaches. In the context of emergency control, we would aim at building contingency independent criteria, i.e. criteria which would be able to predict the future evolution of the system for a wide class of emergency situations resulting from a combination of a wide class of prefault states subjected to various possible disturbances. This could even be a practical requirement if it is not possible to identify the actual contingency in real-time. In the context of preventive security assessment, on the other hand, there are at least two possible options : contingency dependent multi-contingency criteria, and global contingency independent criteria. In the first case, we seek to assess the security of scenarios combining a prefault state and an hypothetical contingency, and these scenarios would be characterized by attributes providing both information on the prefault state and the contingency [AK 93]. In the second case, we would look for worst case security assessment with respect to a class of contingencies, considering the security level of a state as the security with respect to the most constraining contingency. The latter kind of criteria could be very useful in order to assess the global degree of security of the study region and to provide control means so as to achieve security simultaneously with respect to all possible contingencies.

9

Practical contexts

9.1 INTRODUCTION

In the preceding chapter, Fig. 8.1 has synthesized in three main steps the principle of the general automatic learning approach from simulations.

The data base generation and the automatic construction of security criteria are the two tasks which require large amounts of computational resources and where the expertise of security specialists is required to analyze and validate the resulting criteria. Notice that in the near future we expect to reduce the response time of these kind of studies to some hours, by exploiting increasing speed of CPUs and trivial parallelism. While this will allow us to prepare the security criteria closer to real-time and thus take into account a better knowledge of the actual situation, it is clear that the successful derivation of security criteria will rely on the validation by the engineers responsible for security studies and thus cannot be completely automated, nor brought fully into the on-line environment.

However, once the data base has been generated and the security criteria have therefrom been derived and validated, they may be easily, and with a wide variety of possibilities, exploited in the on-line operation context. Moreover, the adaptation of the parameters of the security criteria to a major shift in the operating conditions could be done quite automatically in real-time, provided enough computing power is available.

Below, we discuss the main tasks accomplished in the off-line security study environment of planning and operation planning and in the context of on-line security assessment and real-time monitoring and control. In each particular context, we will indicate the general tasks and suggest possible uses of the automatic learning based framework to improve the quality of the security information and make better decisions. In the next chapter, we will come up with some specific applications, taking into account feasibility and practical relevance.

Notice that in addition to providing a methodology able to produce useful security information by making it easier to run multitudinous simulations and exploit the results, the computer based information acquisition supposes - or at least encourages - a uniform coding of information, and this makes it easier to communicate information among different persons. Further, the use of a same methodology in both planning and operation planning studies will produce a synergy between these two environments thereby improving system performance and economy.

9.2 OFF-LINE STUDIES

One of the main tasks of off-line security studies is to figure out the main weaknesses of a class of future power system configurations, so as to take appropriate decisions to improve the reliability and security of the system at reasonable costs.

The main difference between planning and operational planning is the much higher level of uncertainties in the former case.

9.2.1 Planning

In the context of planning studies, hypothetical situations are considered several years in advance, and parameters of the future equipments are often unknown and must be postulated. For example, important first order parameters like machine transient reactances may show errors of up to 20%, and line reactances are often erroneous by several percent, and of course the load prediction is far from being reliable.

Within these error bounds the planner has to justify investment decisions of a very high financial and technical impact. Probably, the economic costs of future insecurities are rather difficult to evaluate at this step, and even if they are systematically taken into account in reliability studies [DO 86], the importance of security is often underestimated. At least, it is not the feeling of many operational planning engineers that security concerns have received the due consideration in planning studies and it is often true that design decisions have not taken into account security criteria early enough.

An important concern in power system security is that technological, environmental and economical pressure may impose changes in the system design and operating strategies, which in turn may drastically change the limiting phenomena. It is well known, for instance, that in western North America, the limiting factor of the transmission system, which used to be angular transient stability, has become in recent years voltage security. This is due to the successful countermeasures taken to cope with transient stability, in particular the faster (and more clever) protection and powerful (in the very short-term) voltage support devices (excitations and SVCs) as well as fast valving and other

emergency control schemes. Nobody can be certain about the future outlook of power systems, but for sure, further important technological improvements (e.g. FACTS, high temperature superconducting devices, . . .) and socioeconomic changes (e.g. open transmission systems) will continue to strongly influence their structure.

The planning environment is the first place where such a drift from one security problem to another may be detected, provided that extensive and systematic security studies are carried out.

At the same time, the planning environment is the most open one to experimentation of new methodologies. Traditionally, Monte Carlo type simulations have been used to assess reliability; as we discuss in chapter 11, these are closely related to the computer based learning techniques described in this thesis. Exploiting these techniques appropriately relies on the three following coexisting factors [CA 93b].

Models. Appropriate models are needed to study the various, short-term, mid-term and long-term dynamic and static aspects which are important for system security. Much progress has been made in this field during the last 20 years, and we may expect to be able to maintain the adequacy of models when major technological changes will be incurred in the future. Maybe some progress would be needed in handling the unobservable parts of a power system (e.g. load characteristics, interconnections, . . .) by modelling the effect of existing uncertainties in security studies. A technique able to do this is suggested in §12.1.3.

Effective simulation tools. This aspect encompasses algorithms and their mapping on existing hardware. While imperfect in various aspects, we may consider that the existing numerical methods offer a sufficiently complete panel of methods appropriate for security assessment problems. Maybe the most desirable progress concerns the modularity and maintainability of the corresponding software packages, and the construction of appropriate user oriented environments built on the top of the simulation packages. This should allow us to easily combine various simulation modules in order to determine security margins, sensitivities and evaluate design or control options. Further, these top-level environments should be able to exploit, in a transparent way, the available distributed and heterogeneous computing environments.

Data management. Security studies involve large numbers of repetitive simulations, and as increased available computer power and more effective distributed computing environments become reality, these numbers will start growing very quickly. Thus, it becomes more and more important to develop efficient data management methodologies and tools. This concerns both the preparation of input information, helping to choose relevant cases and the management and analysis of output information. The computer based learning framework offers such a methodology. It allows to systematically screen relevant power system situations and disturbances and to

apply various simulation modules in order to obtain the corresponding security information. On the other hand, various complementary techniques are available for analyzing and exploiting the resulting information in order to help taking improved design decisions.

9.2.2 Operational planning

In contrast to planning, in operational planning the system components' characteristics are known with a much better precision and modelling uncertainties are mainly limited to the load characteristics and external systems which are difficult to identify. Now the engineer is responsible for the secure operation of his system, in a more deterministic setting than in the context of planning, while maintenance requirements and economy issues influence strongly the acceptable choices.

Operational planning studies aim also at adjusting parameters, e.g. settings and coordination of protections and preparing emergency control schemes.

Generally speaking, in operational planning security studies lead to the definition of operation guidelines which must be reliable to the greatest possible extent. In particular, this leads to the use of rather detailed modelling practices which so far have strongly limited the number of possible simulations made to define the security limits for the operator. Thus, present practice consists mainly of choosing a small number of relevant situations in a manual way to derive the operation guidelines, while introducing margins so as to avoid insecure operating states.

While the operational planners may be reluctant to use probabilistic techniques and to consider new methods in general, because of their heavy responsibilities, we believe that systematic screening techniques such as those proposed in this thesis, will be very useful in the future to exploit systematically the growing computing powers available. We also believe that many of the security analysis methodologies and tools used in planning, may be inherited in the operational planning environment, as soon as the available software and hardware become sufficiently powerful to use the same models in planning studies as are presently used in operational planning.

Consequently, the present gap between the two environments should shrink and security information could be transferred continuously from the planner to the operational planner; the latter would essentially refine the security limits obtained from preceding studies, given the additional information about system parameters and expected operating ranges. Further, by using common models and methodologies, it will be much easier to communicate among planners and operational planners; in particular better feedback from the latter may be expected leading also to better planning design decisions for future security.

Notice also that using the same data management environment would allow to share

information much more easily, for instance by remote data base access, either from the planners or from the operational planners side. Further, all this information may in turn become accessible as easily to the operator in training sessions and also directly in the control room.

An important aspect worth mentioning concerns the trends towards opening access to the transmission system. While it is too early to assess the exact impact of this on future power systems, it is clear that free access will tend to distribute some of the presently centralized decision processes to various external bidders, controlling on the basis of their own economic criteria the generation and load behavior. One of the possible important consequences will be that the operator in charge of the transmission system, which will tend to a sole "grid", will have to face much more uncertainties about short and medium term behavior of load *and* generation. This means that it might well become infeasible to continue using the deterministic criteria used so far, and probabilistic methodologies able to model and cope with uncertainties would be needed. This is another important motivation to develop techniques such as those described in this thesis.

9.2.3 Training

Since planning and operational planning make it possible to avoid critical situations with a very high probability, operators seldom experience such situations in real life. Nevertheless, they must be prepared to react correctly to such events, and the most effective preparation is via training simulators reproducing the various critical scenarios which may lead to major disturbances on the EHV system.

We are convinced that the large amounts of information about the security of a power system collected within the previous two study environments might be exploited very usefully in the context of operator training.

If a well organized security assessment framework is used in the future planning and operational planning studies, with a systematic way of storing and accessing information about elementary cases, then this information may be easily accessible from the training environment. Thus, scenarios which have been simulated previously may be analyzed by the operator together with the security criteria which have been derived on their basis and which provide the operating guidelines. For example, this information may be used as a catalog to choose security scenarios for the training simulator corresponding to predetermined security characteristics.

Further, in addition to exploiting the individual cases stored in a data base, synthetic explicit models such as decision trees may be shown to the operator via appropriate graphical visualization tools to explain security problems and teach counter-measures.

9.3 ON-LINE APPLICATIONS

In the context of on-line system operation, the task of an operator will be to follow the load buildup and monitor the security level of the system with respect to the most likely contingencies, so as to take provisional actions to ensure the security of the system in preventive mode, or to prepare corrective actions in the case of emergency. This task, while being most of the time routine, may become extremely tricky and overwhelming if the system enters an unusual state, e.g. due to an unusually fast load buildup or to some unforeseen outages.

9.3.1 Normal operation

In normal operation, the security criteria derived from the planning and operation planning environments may be used to help the operator appraise the current situation. For example, security margins may be displayed for various contingencies - as with conventional security assessment approaches - and the situations of the off-line generated data bases found to be most similar to the current state may be systematically tracked according to various similarity criteria defined off-line.

On the other hand, if a potentially dangerous contingency is identified, decision trees may be exploited to identify the most effective control means, and a secure state may be proposed to the operator by looking up the data base. This is of course a dreamed situation which may be reached in some distant future.

Let us notice that in the context of normal operation, economy is a very important aspect. Thus preventive control decisions should not be taken lightly. This implies in particular that the cost of overconservative security criteria is a determining factor of their acceptability. If tools are available in the control room to determine appropriate corrective emergency actions then it is possible to apply the preventive control only temporarily, so as to give some time to the latter tools to determine and arm the appropriate emergency control actions, on the basis of the present situation.

Finally, one very important condition for a method to be accepted in the on-line environment is that it must not increase the probability of erroneously declaring a state as secure with respect to current practices. Of course, no method can pretend to be perfect, but at least the probability of dangerous non-detections should be small enough. Within this constraint, the objective will be to reduce to the extent possible the probability of false alarms, in order to allow an as economic as possible operation.

9.3.2 Under emergencies

The distinguishing feature of an emergency with respect to the preceding situations is that this it is a post-contingency situation, where the short-term evolution is *deterministic* and can lead to a more or less important loss of integrity. Thus the question is not whether we should do something, but what should we do to minimize the loss of integrity. Time - and not economy - becomes the critical factor here, and the main strategy consists of taking quickly some simple palliatives, in order to save enough time to determine and implement further levels of more refined curative actions.

For example, if an important plant is in danger of losing synchronism we may sacrifice some of its generation by shedding one or two units. This may however initiate a process of dropping voltage, which may in turn be mitigated by blocking tap changers or tripping load, which will leave some additional time to get voltage and frequency support from some fast startup units, thus giving time for the operator to further reschedule the generation in a more economic fashion.

Since emergency control is the last chance to avoid moving to the in extremis state, it is important to define appropriate strategies for the early detection of emergencies after the occurrence of a disturbance. As we suggested above, some of these strategies may be derived from the preventive security assessment made during normal operation, but in addition, emergency state detection and control should be able to cope with unforeseen events, since often a dangerous situation results from a complex combination of contingencies for which it is impossible to make provision in normal mode operation.

Another particular aspect of emergency states is that they correspond to unusual - non equilibrium - states, where often real-time information and models may be erroneous.

9.4 COMPUTING ENVIRONMENTS

Before concluding this chapter, we will briefly discuss the impact of modern computing hardware and software environments on the computer based learning techniques applied to security assessment.

Observing the evolution in the last ten years we may identify some important factors.

Client / server. This is a uniform model of interactions between the producer and the consumer of a resource (CPU, data base . . .) which allows to build very complex computer systems on the basis of a simple generic model.

Distributed. It is clear that local and wide area networks have given another dimension to computer systems. With a very cheap local area network technology it is easy to upgrade progressively systems composed of more than hundred workstations,

which may share information and other resources in a very transparent way for the end-user.

Standards. Standardized operating systems (UNIX), communication protocols and software development and run-time environments (languages like C++, graphic libraries like X11 Motif. . .) make it easier to transfer complex applications from one platform to another, and allow easier communication and cooperation between various applications running on different systems.

Open. The preceding three items have increased considerably the flexibility of computer systems, both from the hardware and the software viewpoint, leading to the ideal concept of “open systems”.

Hardware progress. The higher success of the above “parallel distributed processing” approach, as compared to the “massively parallel processing” approach, is also due to the fact that the systems have been able to take advantage - in a very smooth fashion - of the permanent improvements in computing hardware. For example in a period of five years, processor speeds and memory have been multiplied by a factor of more than ten, without any increase in costs.

Due to the cost effectiveness of the distributed memory architecture, it is very likely that in the future this same basic architecture of systems will become the standard platform, used at the same time in the research divisions, in the study environments of planning and operational planning and in operation. Of course, the functional requirements in these various contexts are different; for instance control room applications and supervisory software will probably remain very different from those used in the off-line study environment [DY 93].

Nevertheless, the main point is that no architectural constraints will prevent an application currently used in off-line studies to be accessible in the control room, for example via transparent network computing. On the other hand, there will be virtually no technical barrier for exchanging data among various control centers and study environments. Further progress may be expected in the coming years in the following areas.

Processor speed. In the next three years we may expect processor speeds of standard UNIX workstations to be multiplied by a factor of ten, and maybe another speedup of the same order by the year 2000. At the same time the capacities of short-term and long-term memories will scale up proportionally.

Networking. While local area networks are presently extensively used to distribute applications among dedicated processors, high speed wide area communications become possible among various remotely located areas. This allows to exchange data more efficiently among different departments of a utility and thereby to increase the cooperation among them.

Software. For the building of CPU servers composed of a large number of UNIX machines accessible on the utility network, new coupling tools are being developed, allowing one to exploit heterogeneous distributed memory systems [GE 93b].

Notice that these changes will lead more and more to a uniformization of hardware and software environments, allowing integration in the control room of security assessment software presently used only in the off-line studies. At the same time, information may flow back from the on-line computers to the study environments allowing us to take easily system snapshots and to feed them into the security simulation software. The security studies thus obtained may be systematically analyzed to assess and correct current policies.

9.5 CONCLUDING REMARKS

In this chapter we have stressed the fact that the strong trend in computer architectures is likely to lead to similar computing environments in the three practical application fields of planning, operational planning and on-line operation. At the same time, the computing powers are expected to increase very rapidly in the near future, making possible the use in these three contexts of unified approaches, power system models and data representations.

On the other hand, to exploit efficiently these possibilities, new tools - mainly for data management and top level functions - must be studied and developed. The approach described in this thesis may meet these requirements well. It may be used in either of these application fields to run security studies more systematically; even more importantly, it will *encourage* the sharing of information among the different practical contexts and the use of common methodologies and models.

10

Typical applications

In the preceding chapters we introduced the underlying physical phenomena and the practical functional requirements of various security assessment problems. The aim was to provide a general view of possible applications of the learning approach to security assessment. In this chapter we will present two concrete examples which have been studied in the literature via the proposed methodology. Our purpose is to fix ideas by providing a deeper insight rather than an exhaustive enumeration of possible applications.

10.1 ON-LINE PREVENTIVE SECURITY ASSESSMENT

Nowadays, on-line preventive security assessment is being approached via two complementary methods; the first is the use of limit tables determined off-line; the second is based on on-line numerical computations using more or less simplified analytical models.

The first technique is basically a pattern recognition approach, where the patterns are determined in a tedious manual way. The computer based learning framework essentially provides a means to perform pattern recognition more systematically, thereby allowing us to exploit more effectively the available computing powers while mastering their rapid growth.

In the particular context of on-line transient stability, most of the utilities dealing with this problem presently rely on off-line predetermined limit tables. The other approaches to transient stability evaluation involve bulky computations and are not yet feasible on-line with present day control center facilities. On the other hand, because of the very short time span of transient stability problems, emergency control is limited to more or less local protection systems, also based on pre-established thresholds.

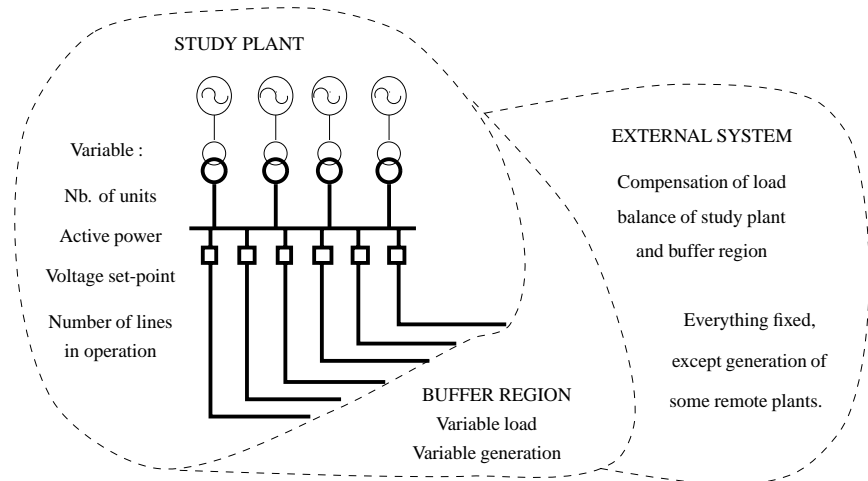


Figure 10.1 Preventive transient stability assessment of a power plant

10.1.1 Example problem statement

We consider a transient stability limited power plant, and aim at identifying the operating limits which should be respected in a normal situation in order to guarantee the ability of the power plant to maintain synchronism with respect to a set of “dimensioning” contingencies. This problem has been studied in the context of our collaboration with EDF; the results are reported in references [WE 90b, WE 91d, WE 91e, WE 93d, AK 93]; below we provide a simplified sketch of this study.

Figure 10.1 shows the power plant, composed of four nuclear units of 1300MW, which feed the remaining power system through four step-up transformers and six 400 kV lines. The study plant interfaces with the external system through a buffer region comprising the relevant 400 kV and 225 kV system whose status might influence the stability of the power plant.

For the stability of this power plant, the following kind of faults are considered as potentially constraining.

Single-line faults. A single-line fault is characterized by a three-phase short-circuit on a line which is cleared by a permanent tripping of that line. Different such faults correspond to different possible locations of the short-circuit on different possible lines. A priori, the most constraining assumption for the plant stability corresponds to a short-circuit on the end of the line connected to the plant’s substation.

Double-line faults. These are very severe contingencies corresponding to a simultaneous short-circuit on two parallel lines and resulting in the tripping of both lines

in the post-fault period.

Busbar faults. These faults correspond to a three-phase short-circuit on the busbar of the study plants' 400kV substation. They lead to the tripping of all lines and machines connected to the corresponding busbar section. Depending on the assumption of the distribution of lines and machines on different busbar sections, different busbar faults are possible in practice.

All in all the above yield a set of 17 different faults : 6 single-line faults (3 normal and 3 consisting of a slow reclosure on a faulted line after 20s), 6 double-line faults (3 normal and 3 consisting of a slow reclosure assumption), 5 busbar faults (2 normal, and 3 with the assumption of unavailable breakers).

10.1.2 Data base generation

To generate a data base representative of normal and extreme pre-fault situations we have screened the following range of parameters.

Study plant (internal) region.

Topology. Prefault outage of 0, 1, 2 or 3 lines out of the six 400kV outgoing "evacuation" lines of the power plant.

Unit commitment. Between 1 and 4 units in operation.

Active generation. Variable and non-uniform sharing of active power among units.

Voltage set-points. Uniform variation of voltage set-points of the power-plant units, so as to obtain a uniform distribution in the interval of [390...420]kV on the EHV side of their step up transformers.

Buffer region.

Load level. Regional load-level is variable, independent of the generation schedule.

Power plants. The operating state of the two closest power plants to the study plant are variable from the point of view of their active power, reactive power and number of units in operation.

Substation configuration. The number of electrical nodes within the three closest 400kV substations, forming the interface with the external system, are varied according to system statistics.

External system.

The external system is essentially kept unchanged with respect to the base case, which is a winter peak load assumption; only some remote power plants are used to compensate the active power balance of the study plant and the buffer region.

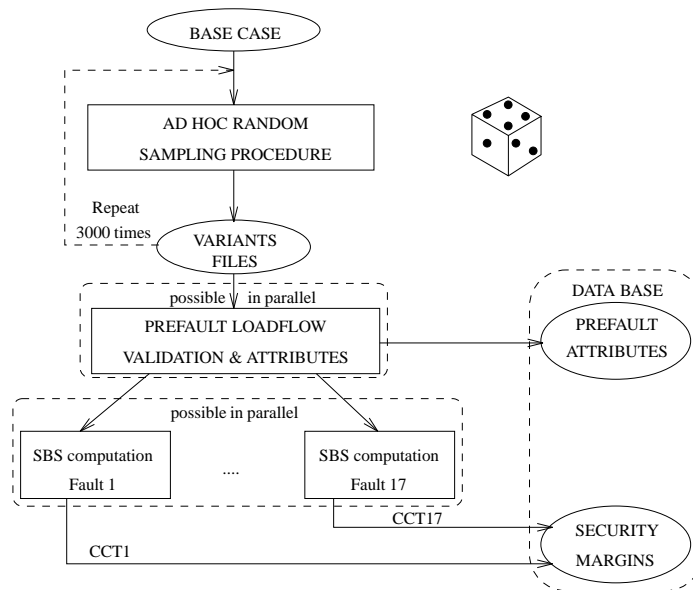


Figure 10.2 Automatic off-line construction of a data base

The definition of the above independent parameters has resulted from discussions with the power system engineers in charge of the stability studies. The buffer region has been defined so as to encompass the part of the EHV system liable to influence the stability of the study plant, and the corresponding effects have been varied in the random sampling procedure *independently* of the state of the power plant, so as to reflect *plausible* system operating states.

On the other hand, the parameters of the study plant itself, including the status of the six EHV outgoing lines, have been varied independently of each other, so as to screen *the full range* of possible operating states. Thus, the resulting security criteria cover a much larger range of plant conditions than those usually encountered in practice. For example, while usually nuclear generation sets are exploited at nominal active power, we have screened also the situation where one or two units operate at intermediate power.

Following these preliminary discussions, a random sampling procedure was developed to construct the data files corresponding to a sample of 3000 pre-fault states. For each state, a loadflow computation was performed, yielding a “sound” state and the attributes describing the plant and outgoing lines’ statuses were stored in the data base files, together with the 51,000 CCT values, obtained from the systematic stability simulations performed for the 17 above defined contingencies. This is schematically illustrated in Fig. 10.2. These computations involved a complete model of the EDF

EHV system, comprising about 500 nodes, 1000 branches and 60 machines. To speed them up, a simplified dynamic model was used for the machines and the step-by-step transient stability computations were performed in parallel on four 28MIPS workstations, yielding an overall response time of about ten days for the 51,000 CCT computations.

Using a realistic dynamic model for the generators would increase by a factor of about 30 the total computing time. Thus, to make computations feasible in practice, faster workstations and a higher degree of parallelism would be required, which will be technically possible in a very near future. Notice also that the number of faults which would be studied in reality for each operating state would probably be smaller. For instance, as we will see in §13.3, the single line faults are always less severe than the corresponding double line faults and would not need to be studied. Thus, using for example twenty 90 MIPS workstations these simulations could be done within about the same response time of ten days.

10.1.3 Security criteria learning

For each state a wide variety of attributes have been computed describing the state of the study plant and the power system elements inside the buffer region. They are essentially more or less sophisticated parameters of the prefault operating state, such as power injections and flows, voltages, topological indicators, number of machines and lines in operation, short-circuit powers Depending on the projected use of the security criteria only a subset of these attributes was used to derive security criteria in the form of decision trees.

In refs. [WE 90b, WE 91d, WE 91e, WE 93d, AK 93] and also in the summary provided in chapter 13.3, we further discuss the investigations carried out. They concern for instance the decision trees' reliability and complexity assessed in terms of the degree of sophistication of the used candidate attributes and also the various ways of exploiting multi-contingency information of the data base.

Below we will merely illustrate two particular examples.

Global security criterion

A *global* decision tree was derived to characterize the region of simultaneous stability with respect to a set of 14 faults.

A state is considered to be stable with respect to a particular fault if the CCT of this fault is larger than the actual clearing time (90ms for line faults and 155ms for busbar faults). Among the 3000 prefault states, 2300 were used as a learning set; among these 733 are unstable with respect to at least one out of the 14 contingencies.

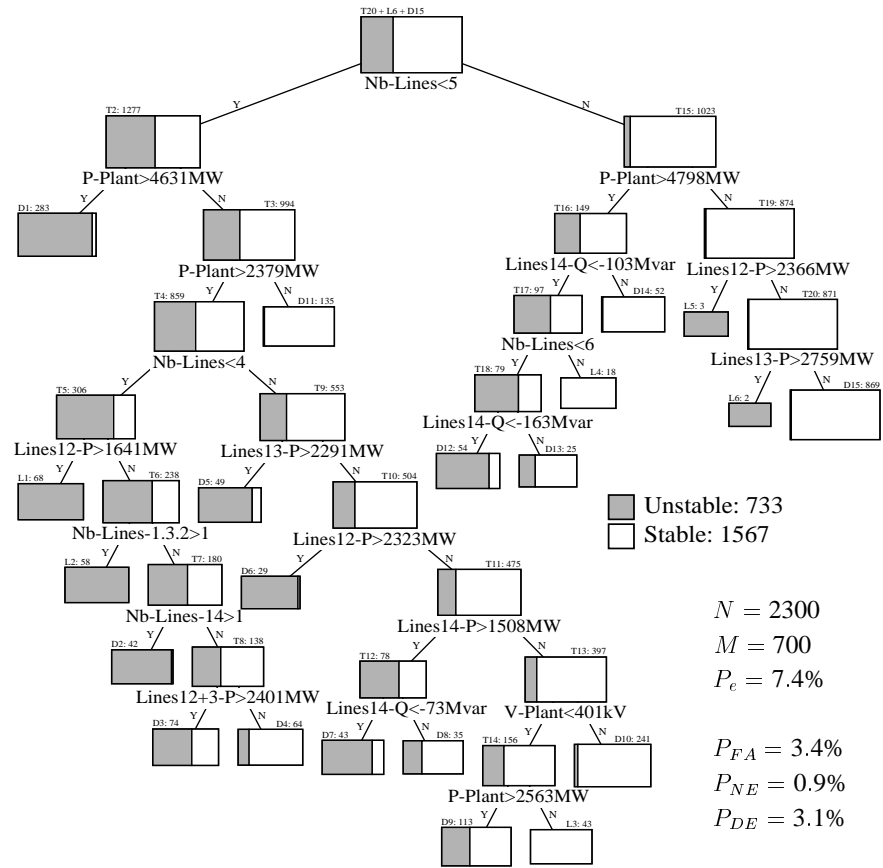


Figure 10.3 Global decision tree covering 14 contingencies. Adapted from [WE 93e]

The resulting decision tree is portrayed in Fig. 10.3. One can see that to characterize the stability of the study plant the tree building procedure has selected a subset of 10 attributes out of the 40 candidates. These are simple, more or less directly controllable pre-fault parameters like

Nb-Lines, the number of outgoing lines in operation.

P-Plant, the total active power of the machines in operation in the power plant.

Tr-P, Tr-Q, active or reactive power flows through various lines.

V-Plant, the EHV voltage in the plant substation.

Figure 10.3 also illustrates that the tree may assess in a single shot the stability of the

power plant and also suggest control means in order to move, whenever necessary, the operating state from the unstable region to the stable region.

Of course a decision tree provides a rather rough model and is subject to classification errors. As indicated in Fig. 10.3, 7.4% of the 700 independent test states are misclassified. More precisely,

3.4% correspond to false alarms where a state is declared unstable while it is actually stable;

0.9% correspond to “normal” errors, namely states which are marginally unstable¹ and which are classified stable by the tree;

3.1% correspond to dangerous diagnostics, namely states which are fairly unstable² and which are classified stable by the tree.

Single-contingency security criteria

In the case where a learning problem corresponds to a particular contingency, the security classes are defined with respect to one or more threshold values on the CCT of this contingency. In the most simple case which we consider here, the stable and unstable class are defined with respect to the actual clearing time of the CCT.

Building a decision tree for this more specific stability problem aims at exploiting more specific information concerning the plant operating state, such as for example the number of lines in operation in the post-fault configuration, i.e. the number of lines in operation in the pre-fault which are not tripped for the particular assumed fault.

Figure 10.4 illustrates a decision tree thus obtained for a particular double-line fault. A state is classified as unstable if the CCT of this fault is smaller than the actual clearing time supposed to be equal to 90ms. It is interesting to observe that the most discriminating attribute used at the root takes into account fault specific information : P/N_{bl} denotes the ratio of the total active power generated in the pre-fault state by the number of outgoing lines remaining in operation in the post-fault state. For this tree the test set error rate has reduced to 1.9%³. At the same time, there are only 3 dangerous non-detections, i.e. cases classified stable by the tree, while their CCT is actually smaller than 81ms.

Using the hybrid approach described in §6.2, we have derived a multilayer perceptron from the DT, where the 7 test attributes identified by the tree are the input variables, and where the value of the CCT is approximated in the interval $[70 \dots 110]$ ms around the classification threshold. The weights of the multilayer perceptron composed of 7

¹There exists at least one fault whose CCT is in the interval $[0.9\tau \dots \tau]$, where τ is the clearing time.

²There exists at least one fault whose CCT is in the interval $[0.0 \dots 0.9\tau]$.

³To build and test the tree only the states among the 3000 of the data base where at least one of the two faulted lines is in operation have been used; this yielded respectively 2132 “relevant” learning states and 643 “relevant” test states.

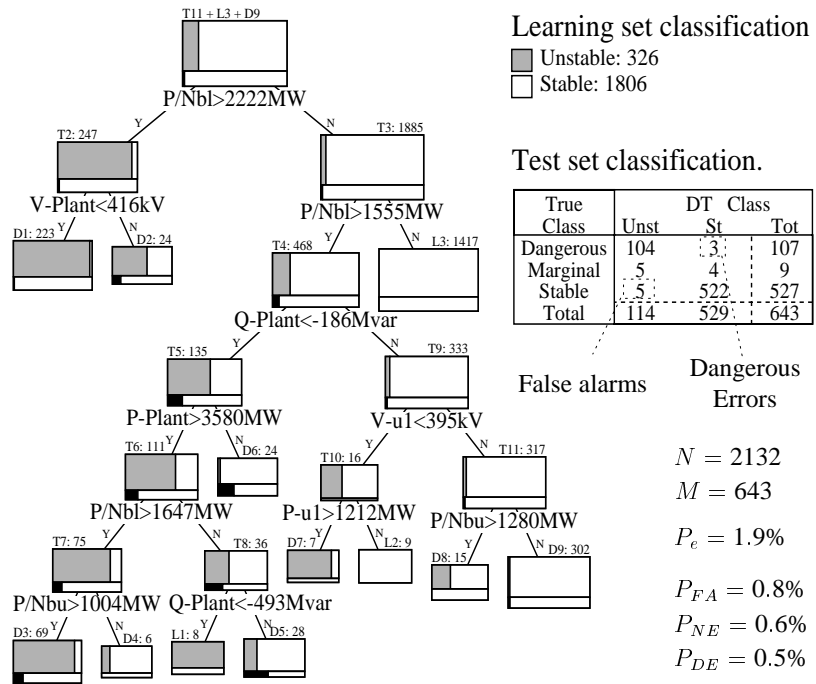


Figure 10.4 Single contingency decision tree for a double-line fault. Adapted from [WE 93e]

input neurons, a single hidden layer of 15 neurons and a single output neuron, were adapted on the basis of the known CCT values using the BFGS procedure. The input attributes have been prewhitened and the output stability margin was normalized as is suggested in Fig. 10.5. The procedure reached a local minimum of the regularized MSE criterion within 80 iterations, corresponding to a CPU time of 4440 seconds (to be compared with the CPU time of 390 seconds, necessary to build the decision tree). This allowed us to further reduce the test set error rate to 1.2%, and at the price of a false alarm rate of 3.4%, to eliminate all dangerous and normal non-detections!

10.1.4 Comments

The preceding example suggests several interesting aspects of decision trees built in the context of stability assessment, and more generally of preventive security assessment.

First of all, it is interesting to use standard operating parameters in order to build security criteria, and in particular decision trees. This makes it possible to analyze the attributes actually selected and their threshold values. For example, the decision trees

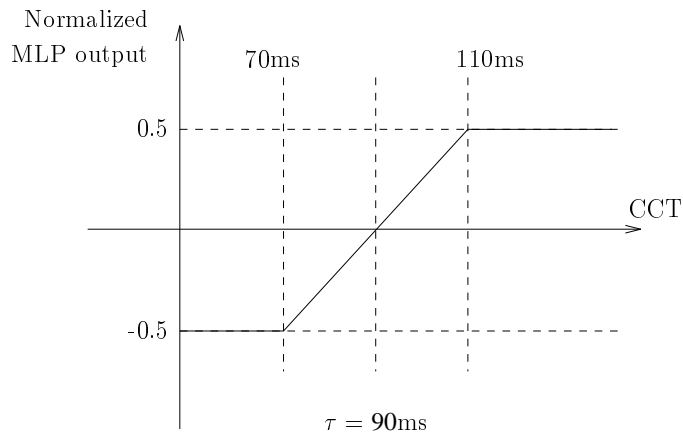


Figure 10.5 Output normalization for the hybrid MLP CCT approximation

of Figs. 10.3 and 10.4 reproduce well known relationships among standard operating parameters and stability.

Global and single contingency decision trees are complementary. The former provide straightforward conditions of simultaneous stability with respect to the set of faults for which they have been derived, and this kind of information is directly applicable for preventive control. The latter trees are liable to provide a more reliable security assessment, and may therefore be useful - possibly together with other more black box criteria - to identify all potentially dangerous situations.

The use of security margins as a complement of security classes has also several interesting outcomes. First of all, margins may be exploited to analyze more closely classification errors, since they allow us to differentiate among dangerous and normal errors. Further, they may be exploited in a regression model to provide a smooth approximation of the stability which may in turn be used to reduce the probability of non-detections of unstable states. In the present example this approach has been very effective in reducing the probability of non-detections.

10.2 EMERGENCY STATE DETECTION

Putting aside the case of thermal overload problems of static security, in emergency state detection main issues are the limitation of available real-time information about the system state due to the shorter time frames, and the fact that the system is in an abnormal, dynamically evolving situation.

While in preventive security assessment real-time information may be assumed to be the

output of a reliable state estimator validated by the operator, and there is enough time available to make more or less sophisticated network computations, in the emergency state the available information often reduces to a set of raw measurements, and the time period left for decision making becomes much shorter.

One of the main difficulties in emergency state approaches is that the power system is in a dynamic state at the moment of acquiring the attribute values, and it is necessary to make sure that the stability criterion derived is sufficiently robust with respect to random (uncontrolled) variations in the data acquisition time.

Another fundamental question is how to choose an appropriate time constant to refresh periodically the stability criteria to adapt them to changing power system conditions. The related feasibility question concerns the amount of computational power which must be invested in order to provide reliable enough criteria with a response time compatible with the frequency of updating the criteria. This may strongly depend on the particular power system under consideration and the degree of modelling sophistication required for simulations. Thus, there is still need for more in-depth investigations, in particular in the context of testing the feasibility of this approach for real large-scale systems.

Note that computer based learning approaches have already been proposed for real-time transient stability prediction in the emergency state [OS 91, RO 93]. Due to the very short time frame available (of say between 100 and 300 ms after the fault clearance), these approaches would be supposed to be fully automatic and closely related to adaptive system protection and adaptive out-of-step relaying [CE 93].

The authors of ref. [RO 93] discuss preliminary research on transient stability prediction on the basis of real-time phasor measurements, using decision trees. Here, we will consider the case of mid-term voltage instabilities and provide a discussion of the particular considerations of emergency state detection.

10.2.1 Example problem statement

We consider the EHV system depicted in Fig. 10.6, which was designed to reflect typical behavior of a voltage weak region of an EHV power system, importing variable amounts of reactive power through the interconnection lines.

The effects of the external and buffer region are modelled by the infinite bus at node 11, interconnected to the weak region by two rather long 380kV lines. The interface between the local EHV transmission system and the 90kV subtransmission network is represented by two buses 50 km apart; at one of these buses a local power plant is connected composed of three units of 113 MW (133 MVA). Reactive shunt compensation is connected to the 90kV buses, whose voltages are normally regulated via the OLTCs which equip the 380/90kV transformers. The MV distribution networks and the load

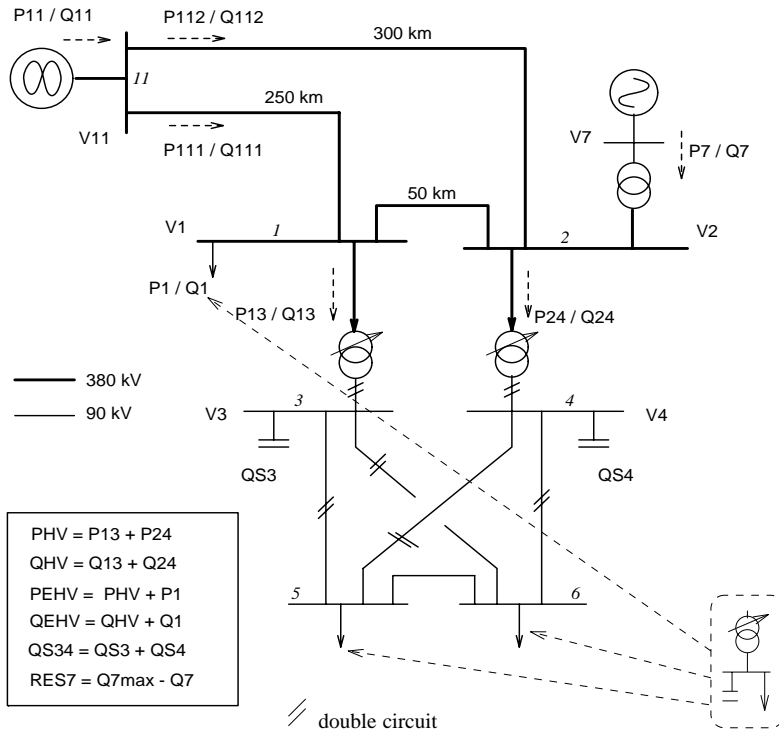


Figure 10.6 Voltage emergency state detection in a weak region. Adapted from [VA 91b]

are represented by an equivalent model at buses 1, 5 and 6, including an equivalent MV load, compensation and OLTC. The voltage regulator of the local generation sets is equipped with a maximum excitation limiter which tolerates a temporary overexcitation of about twice the permanent limit, during 40s.

Five possible disturbances have been considered

Line trippings. Loss of line 11-1 or 11-2.

Unit tripping. Loss of one or two units of local generation.

Combined. Loss of line 11-2 and two units of local generation.

The emergency state detection problem consists of predicting during the *just after disturbance state* (JAD), i.e. during the short-term equilibrium state that the system reaches after the electromechanical transients have died out, say about 20 seconds after the disturbance inception. Using a snapshot of system measurements, the prediction determines if the forthcoming OLTC load restoration process, together with the action of overexcitation limiters, will lead to voltage collapse or not.

10.2.2 Data base generation

A data base representative of the JAD states was obtained by generating, firstly, a sample of various pre-fault situations, and applying to each state the five disturbances to produce the five corresponding voltage stability scenarios. These have been simulated with a variable step short-term dynamic simulation program, which computed the attribute values and allowed us to classify the scenarios as either critical or noncritical. This is further described below.

Prefault states

A sample of 500 pre-fault operating points was generated randomly. Since no prior information was available for this synthetic system, uniform and independent prior distributions were used for the following input parameters of the loadflow [VA 91b].

External system. V_{11} was varied between 1.0 and 1.1 pu.

Local generation. P_7 was varied between 0 and 350MW and the minimal number of units was put into operation to yield this power. The reactive power of each unit was chosen uniformly between -20 and +64 MVar.

Load level. The total load was varied between 900 and 1350 MW, and distributed among loads at buses 1, 5 and 6.

Reactive compensation. The number of capacitor banks (of 50 MVar each) at buses 3 and 4 was varied between 2 and 6.

The 500 pre-fault states were generated by drawing randomly the above input variables and applying a loadflow computation; those states which were actually kept were only those in which this computation converged properly and for which the EHV voltages were within predefined bounds.

JAD states

To obtain the data base composed of 2500 JAD states, each of the 500 pre-fault states was combined with the 5 disturbances. For each of the corresponding 2500 scenarios, the disturbance was simulated starting from the pre-fault equilibrium using a standard numerical integration program. At time $t=20$ seconds, the attributes characterizing the JAD states were computed and saved into the attribute files. The simulation was continued up to five minutes and the scenario classified as noncritical if the voltages controlled by the 3 OLTCs were successfully brought back to their set-point values.

The overall data base generation procedure, whose aim was to provide a representative sample of possible JAD states, combining various pre-fault operating states and disturbances, is illustrated in Fig. 10.7.

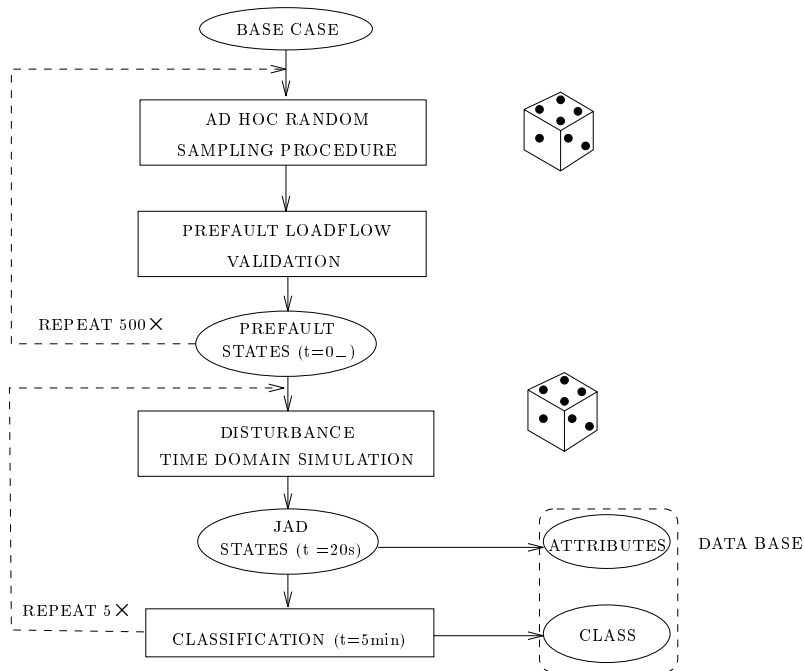


Figure 10.7 Construction of a data base of JAD states

The 28 candidate attributes used to characterize the JAD states are those indicated in Fig. 10.6. They represent essentially EHV quantities which may be available from the SCADA system in the JAD state.

10.2.3 Security criteria learning

To obtain a security criterion, the data base was randomly split into $N = 1250$ learning states and $M = 1250$ test states. A decision tree, built on the basis of the learning states and the 28 candidate attributes is represented in Fig. 10.8. This is essentially an emergency state detection criterion applicable in the JAD state, independently of the prefault state and the particular disturbance which are at the origin of the JAD state.

The decision tree is composed of 7 test nodes and 8 terminal nodes. Its top node corresponds to the complete LS , composed of 454 critical and 796 noncritical states. Out of the 28 candidate attributes only three have actually been selected to formulate the tree. In fact, two of these, V4 and Res7, carry 97% of the information of the decision tree. When used to classify the 1250 unseen test states, the decision tree realizes 96.24% correct recognitions. Thus, despite its simplicity, it is able to correctly represent voltage security behavior of the considered system. Among the 47 classification errors of the

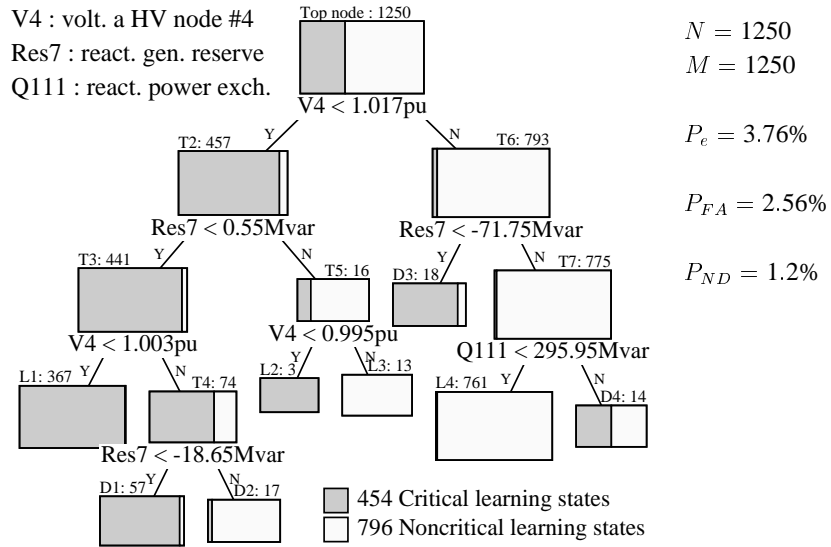


Figure 10.8 Emergency state detection tree. Adapted from [VA 91b]

tree there were 15 non-detections and 32 false alarms.

The geometric representation of the decision tree is given in Fig. 10.9, where its critical and noncritical security regions have been projected on V4 and Res7, together with the 2500 states of the data base. Each class appears as the union of hyperboxes corresponding to the terminal nodes of this class. In turn, each terminal node's hyperbox is defined as the intersection of the semiplanes defined by the tests at its parent nodes.

Further, the hybrid DT-ANN approach was applied. Since in this particular case no continuous security margin was available, a *classification* multilayer perceptron was derived from the decision tree. This is the two hidden layer perceptron represented in Fig. 10.10. Its initial weights have been derived by translating the decision tree and then adjusted so as to reduce the MSE, by using the BFGS optimization method. This allowed us to reduce the error rate from 3.76% to 2.96%, corresponding to 7 non-detections of critical states and 30 false alarms. Obviously, the hybrid DT-ANN approach has improved the reliability of the tree much more significantly in the previous case of §10.1.3, than in the present case. The reason may be found in the richer information provided by the CCT margin in the transient stability case of §10.1.3.

10.2.4 Comments

The approach used to build a representative data base for emergency control consists of applying a set of possible disturbances to a representative sample of prefault states.

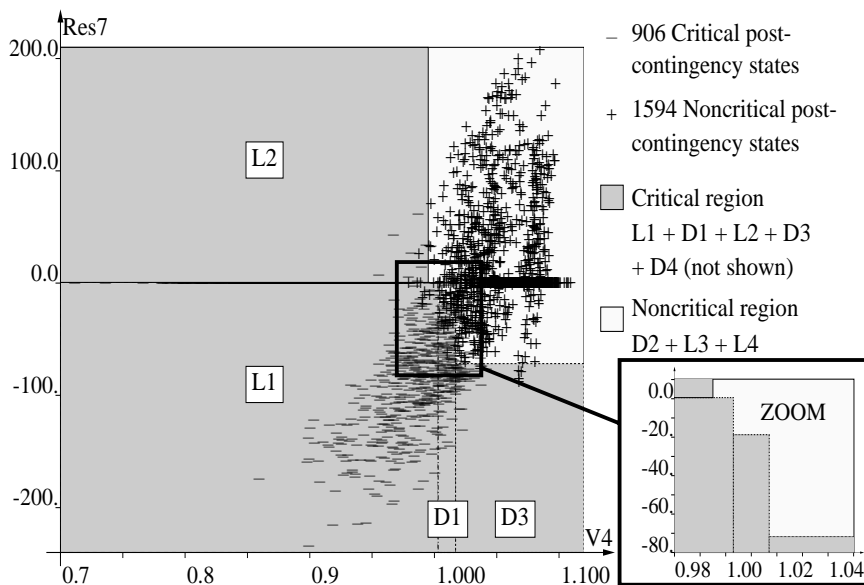


Figure 10.9 Critical vs noncritical regions of the DT of Fig. 10.8. Adapted from [WE93a]

While in the above example we have merely illustrated the idea of building a contingency independent criterion, in real life systems various questions may be raised. For example

Should (or can) the emergency state detection rely on fast identification of the disturbance ?

Should the criterion be built for a large set of possible disturbances or would it be better to use a set of disturbance specific criteria ?

Should (or can) the emergency state detection rely on information concerning the prefault state, e.g. predetermined security margins ?

Should the criteria be built off-line for a large range of prefault situations, as in our example, or should they be tuned to a much smaller range of prefault states and be adapted on-line ?

How can uncertainties about the model used be taken into account when generating a data base ?

Can we assume that the system snapshot is taken at a fixed instant after the occurrence of the disturbance ?

How can we define an appropriate compromise between the early anticipation of emergencies and the selectivity of the detection ?

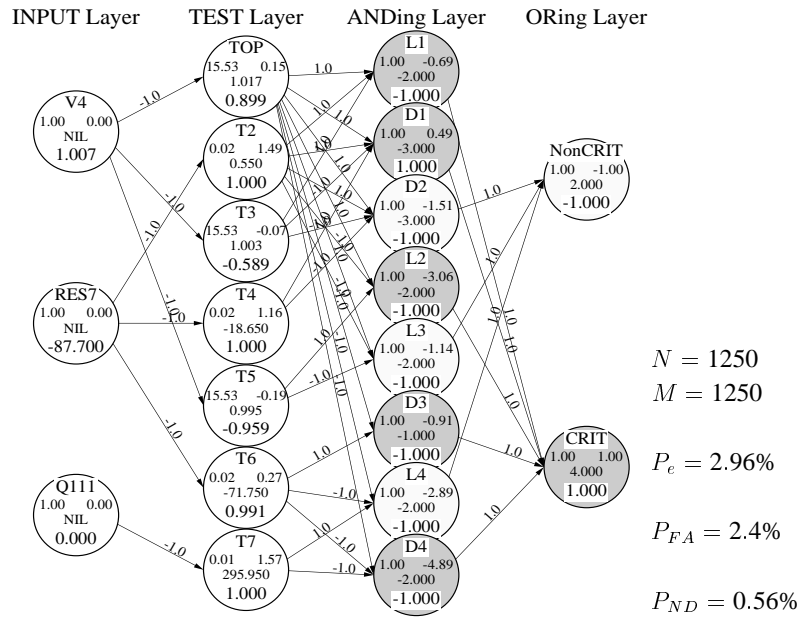


Figure 10.10 Multilayer perceptron derived from the DT of Fig. 10.8. Adapted from [WE 93a]

How can emergency control actions be derived appropriate to cure the detected problem ?

In the sequel we will address some of these questions; the answers may of course depend strongly on the type of security problem considered (in particular on the time scales involved) and the physical characteristics of the considered power system.

11

Meaningful data bases

Having swept through the main considerations concerning learning methods and security assessment contexts, and having fixed ideas about the similarities and differences of various practical learning based security assessment problems, we are now ready to discuss the thorny problem of data base generation.

Any researcher who has been involved in the application of pattern recognition, machine learning or neural network methods to power systems security assessment has realized that obtaining a representative data base is a difficult problem. Indeed, whereas for many learning problems we may consider that the data bases are provided a priori (consider for example load forecasting, letter recognition . . .) in the case of security assessment the samples need to be generated via a computer based simulation of random sampling. This is because it is not possible nor desirable to build these samples solely by collecting data from usual power system situations.

For example, when a hypothetical system is considered (in planning studies, or when testing methodologies on synthetic systems) there is no available statistical information about usual operating regions. Further, when a power system is operated, security and other technical considerations introduce possibly strong correlations among operating parameters¹, and these would lead in practice to represent mostly the secure situations resulting from past security guidelines, whereas the purpose is precisely to build a sample which will contain rich enough information about secure and insecure states so as to improve these guidelines.

Thus, there is a rationale to free the data base generation from too many strong hypotheses about operating conditions, and this introduces the need to define a priori an approach for generating the data base. For very simple systems corresponding to low dimensional attribute spaces, it may be possible to generate data bases in a systematic

¹Here we use the term parameter to denote any kind of variable, either topological or of the continuous electrical state type.

fashion, e.g. by assuming uniform distributions of attribute values in a first step and then perhaps zooming in later on smaller regions overlapping the security boundary [WE 86, EL 89]. Unfortunately, for the study of real, large-scale systems, where a minimum of several tens of degrees of freedom need to be considered, it is quite unlikely that a general and well justified approach could exist for the data base generation which will thus be both ad hoc and empirical in nature.

During the last 8 years we have been involved in the generation of many different data bases for transient stability and voltage security issues of several realistic systems. We will try to synthesize the acquired experience in the following sections, and propose some additional methodological improvements as deemed necessary.

11.1 LOCAL NATURE OF SECURITY PROBLEMS

We have already mentioned that when we consider a security study, the first step consists of defining a study region. While we might consider the particular power system of a utility as a single “tight” system and its security as a single *global* concept, it is well known that the security is up to a certain degree local, and its study will take advantage of decomposing the overall problem into subproblems.

The decompositions may follow various criteria and it is hardly necessary to say that they will depend on the particular security problem at hand, i.e. the characteristics of the considered power system and the considered physical problem. Practical examples are for instance given below.

Site studies, as in the study of the transient stability limits of an important power plant, described in §10.1.

Transmission corridor studies, as in the example of §3.4.1.

Load region studies, as in the study to be described in §14.4.

The main point is that the local nature of security problems is exploited by the utility engineers to decompose the overall system problem into a number of subproblems easier to appraise, and this decomposition is based on *prior* expertise and physical insight. The same should be done when applying computer based learning techniques, to take also advantage of prior expertise. On the contrary, applying these methods to overly general security problems without exploiting prior expertise (which is always conditioned to specific security subproblems) may lead to unduly complex solutions, if not to a disaster.

Once the considered security problem is relatively well understood things become easier. In particular, during the several subsequent steps, existing knowledge may be

injected for

- defining random sampling schemes for generating a data base;
- choosing contingency lists (and models) to evaluate security;
- defining security classes or margins and candidate attributes to characterize security;
- analyzing the resulting security criteria and validating them;
- suggesting feedback information on previous aspects to improve future data bases and security criteria.

These aspects are considered in the next few sections.

11.2 RANDOM SAMPLING OF STATES

Once the considered security problem has been defined, the next step is to specify the random sampling of the data base of relevant *normal* power system states. Notice that even if we are interested in *emergency* states, an appropriate approach consists of first defining the random sampling of *normal* states and then applying disturbances to produce emergency states. This is due to the fact that prior information is available about normal states, which are considered in security studies and which are much more usual than emergency states.

The definition of random sampling requires the decomposition of the power system into a study region, a buffer region and an external system, as we have discussed in §8.3. Recall that the study region encompasses the part of the system corresponding to the primary parameters which may influence its security, and the buffer region the intermediate part where secondary parameters may influence security marginally and which should be taken into account; several approaches to choose values of these latter *free parameters* are discussed below.

11.2.1 Primary parameters

The primary parameters are a subset of those which are known or suspected to have a strong influence on the security, and which are not supposed to take a constant, a priori known value [LE 90b]. For the generation of a representative data base, these parameters could be sampled in a uniform and a priori independent random sampling approach. Note that from a practical point of view, when many possible factors affect security we may assume a priori that there may be interactions among these factors;

hence the necessity of sampling the factors independently, in order to be able to identify the interactions.

For these primary parameters the actual type of distributions used is however often neither uniform nor independent, due to several practical limitations. The first, main limitation is that some of the factors which are to be studied do not correspond to independent input variables of the load flow computation used to construct and validate the operating states. The second reason is that in practice there are some *central* regions of the operating space where we would like to obtain maximally reliable security information. On the other hand, we are not willing to cut off completely the extreme regions; hence the necessity of a compromise between the representation of usual and extreme situations.

In practice these considerations lead to non-uniform distributions and non-independent primary variables. While this is unavoidable for any realistic power system security problem, there are some straightforward safeguards against possible pitfalls. In particular, deterministic rules introducing dependences among variables (such as economic dispatch) should be avoided. Further, we should avoid as much as possible choosing the operating states directly on the basis of their security level; in particular, approaches aiming at generating samples only near a particular security boundary are almost guaranteed to produce misleading results and to be brittle with respect to changing conditions.

It is important to note that any kind of trick trying to cleverly choose the samples so as to improve the reliability of information may possibly cause serious difficulties in interpreting and validating the results. In addition, it is liable to introduce further difficulties when system conditions change and when new security problems are considered. Finally, these tricks turn out to be unable to significantly improve the reliability of the resulting criteria [EL 89]. In our opinion it is certainly preferable to use a more loose sampling approach, exploiting less strongly existing correlations and security information, even if we are required to pay the price of generating somewhat larger samples in order to obtain the desired degree of accuracy. Anyhow, this amounts to cheap computing power, without time consuming human intervention.

The independent variables considered for sampling are generally topology, load and generation. Often, some weak correlation between load and generation is introduced in order to control indirectly the power flows through some critical interfaces and to avoid completely unrealistic situations in terms of these power flows.

11.2.2 Free parameters

Free parameters are those whose influence is known to be quantitatively small in practice and whose explicit modelling is not desirable. There are basically two approaches to

take their effect into account.

The first would merely consist of randomizing their values so as to cancel their mean effects. In order to avoid biasing the study of the primary parameters it is then paramount to choose these values independently from the values taken by the latter parameters. The sampling distributions of the free parameters may be chosen on the basis of statistical information available from the historical data. Otherwise, it is often justified to use Gaussian distributions, at least if they are influenced by many independent considerations, as is often the case.

The second approach, which we could term the “min-max” approach, consists of choosing a particular and constant set of values for the secondary parameters which offer a guarantee of conservatism. These are the so-called “umbrella” configurations used by power system engineers [RI 90]. For most security problems it is not too difficult to identify such situations, due to the monotonicity property of security margins with respect to the usual parameters. Of course, it would also be possible to consider simultaneously the dual extremely optimistic case, and to characterize each combination of primary parameters by the security interval corresponding to the most pessimistic and most optimistic choice of secondary parameters.

11.2.3 Topologies

As we have already mentioned, topology is a parameter similar to load or generation. One of the possible difficulties is due to the combinatorial nature of topological variations. A good approach to define an appropriate sampling scheme for topologies consists of defining one or more hierarchies of topological classes and to choose sampling probabilities for each subclass. Practical examples of this are given in chapters 13 and 14.

11.2.4 Constraining the set of generated states

There are several constraining effects which cause the a posteriori distributions obtained in a data base to be different from those initially specified.

Firstly, a random sampling scheme as suggested above introduces generally some constraints among variables. For example, for a generation plant we might introduce specifications that under certain conditions the number of machines in operation must not be larger than a certain value. Or there may be an upper bound on the total active generation of a power plant in a radial configuration. For the security problems corresponding to real power systems which we have studied, there have always been several such special considerations which cause the resulting a priori probability distributions to be distorted.

Secondly, once a variant has been drawn randomly it must be transformed into a proper specification of an electrical normal state of the system, which implies a load flow or state estimation kind of procedure. This will in turn modify the statistical distributions, for instance because the load flow computations do not converge or because the resulting states do not correspond exactly to the input specification.

A third level of filtering which is generally applied, consists of excluding from the data base unrealistic situations on the basis of tests applied to output variables of the load flow; e.g. voltages and power flows must be within tolerable limits. The effect of this filtering on the resulting distributions should also be analyzed.

A systematic approach to analyze the effect of filtering on the resulting distributions is to construct a so-called a priori data base composed of the input variables drawn randomly for each state, and to classify these states as “accepted” or “rejected”, the latter class being subdivided into subclasses corresponding to the different reasons for rejection. This data base may then be analyzed systematically using the same available statistical tools which are used to analyze the attribute distributions in the a posteriori obtained data base of retained states.

For example, Fig. 11.1 shows the effect of loadflow convergence filtering on the distribution of the power transfer through the James’ Bay corridor of the data base described in 3.4.1. In this case, 15000 states have been drawn a priori to yield the 12497 a posteriori states. As we can see, the effect of non-convergence, while slightly more important for the lower power flows, did not significantly modify the sampling distribution of this attribute.

In a development stage, when the random sampling software is designed for a new set of specifications, this analysis is very important to identify possible bugs and to draw to the attention any software limitations.

11.2.5 How many states should be generated

Of course, the answer to this important question will depend on the number of degrees of freedom in the primary parameters, the complexity of the relationship between these parameters and the security status and also the degree of reliability which is sought. Further, practical tractability limitations put an upper bound on the number of states in a data base.

Ideally, we would like to screen all possible combinations of situations, but with present day technology, and for real medium to large-scale power systems it would be hardly feasible to consider more than say 10,000 to 20,000 states. Further, this limit may become much lower depending on the complexity of security simulations and on the desired response time.

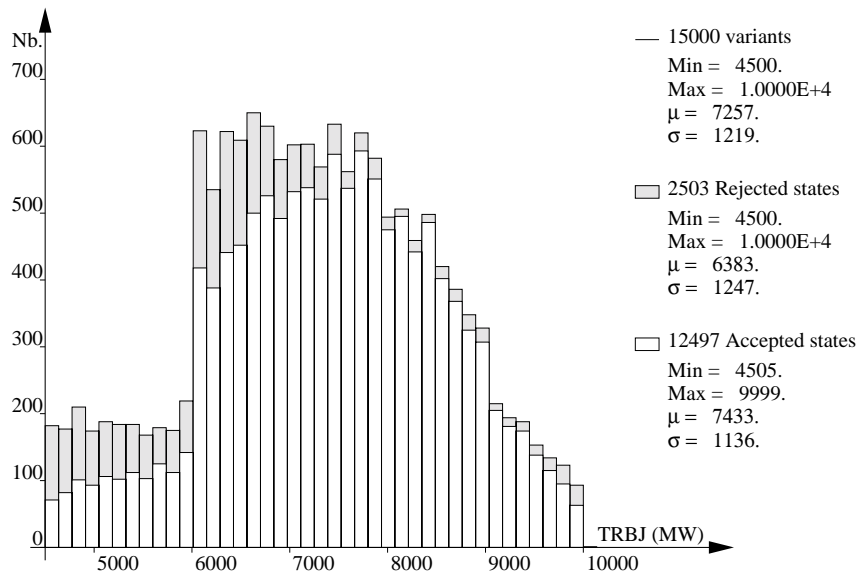


Figure 11.1 *Effect of loadflow divergence on the distribution of a power flow*

On the other hand, there are some indications that below say 20 to 50 samples no information at all may be extrapolated reliably. Thus, a representative data base should contain several times this number of states for all a priori important categories of states. In particular, these categories concern the main security classes and the major topological families. Finally, to estimate test set error rates a sufficient number of test states must be taken out of the data base, say between 500 and 1,000 states, at least.

If we combine these orders of magnitude, we conclude that a realistic data base could contain between say 3,000 and 15,000 states. This number is of course a purely indicative order of magnitude, but if a data base contains less than say 1,000 states we would be very reluctant to draw any valuable conclusions at all from it.

11.3 ALL SAMPLING TECHNIQUES ARE BIASED

Whatever circumspection and precautions are taken when generating a data base, there is no escape in using judgement for deciding to which population we can honestly generalize the results obtained in an experiment based on controlled random sampling.

All sampling approaches are biased, be it only by the choice of the independent variables taken to screen the situations. So, it is our conviction that within this framework, success relies very strongly on the collaboration of utility engineers and their ability and willingness to take the responsibility of analyzing, criticizing and finally validating

the criteria. This is why it is so important to stick quite closely to *their* way of looking at security problems and to provide the security criteria in a form fitting with human analysis faculties. This brings us to the next section, where we give some indications for future strategies to truly assess and validate obtained security criteria.

11.4 HOW TO VALIDATE . . . TRULY

The first step of validation consists of testing the robustness of security criteria derived on the basis of a test set of states of the data base which have not been used for the design of the security criterion.

This is however not sufficient and may be misleading in various ways. Too good results may be due to the exploitation of some correlations which have been built into the data base unduly. Bad results may not be representative of real performance. In particular, false alarm rates may be much higher than they would be in real life due to the fact that the states in the data base are much more concentrated around security boundaries than in real life.

The second step of actual validation will be to compare the security criteria with prior expertise and to determine the plausibility of the modelled security criteria. This presupposes some possibility of interpreting the statistical relationships which are modelled in the data base. This may still be biased, in the sense that if everything goes right, the information reflected in the base itself depends on prior beliefs, and some effects may have been missed. So a good idea would be to generate some cases able to check the hypotheses.

Thus, the third step will consist of generating some independent samples by relaxing some of the hypotheses initially used to generate a data base; in particular it would be very useful to collect real life samples and generate some random variants of these samples to test the data bases.

Finally, when assessing the quality of a security criterion it is clear that different *types* of errors should be identified. In particular the dangerous errors concerning the insecure states which are missed should be analyzed in detail. They may correspond either to normal errors, i.e. errors which are very close to the security boundary and abnormal errors or outliers which correspond to states far away from the sample of states used to derive the security criterion. The analysis of these latter cases should provide guidelines in order to improve the representativity of the data base. How to do this is, however, an open question at the present stage of research.

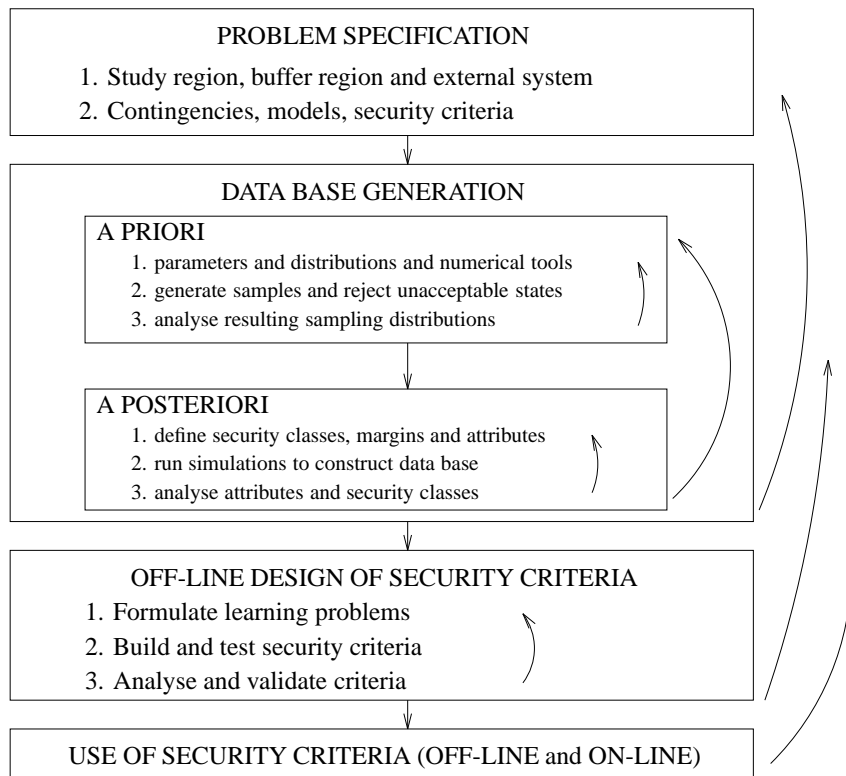


Figure 11.2 Overview of the learning based security assessment approach

11.5 RELATIONSHIP WITH MONTE CARLO SIMULATIONS

Figure 11.2 provides a synthetic view of the various steps involved in the application of the security assessment framework. The four main subtasks concern the problem specification, data base generation, off-line design of security criteria and use of security criteria. The top-down arrows show the logical relationship whereas the bottom up arrows indicate the iterative “generate and test” nature of the overall process.

It is interesting to notice that this overall structure is quite similar to the approaches presently in use at many utilities to determine the security criteria [LE 90b, RI 90]. The main difference between the two approaches is that the present manual approach considers selected power system situations one by one whereas the statistical approach looks simultaneously at large samples of representative states.

On the other hand, Monte Carlo methods are also based on the random sampling of scenarios followed by the simulation and analysis of each scenario. They are used as a

complement to analytical methods, for example in the context of reliability evaluation and probabilistic production costing studies, in order to compute expected values of interesting quantities, such as failure rates or operating costs. Monte Carlo methods are used to take into account complex models, where the analytical computation of expected values would lead to unacceptable simplifying assumptions. Reference [PE 92] gives a good discussion of analytical vs Monte Carlo types of techniques in probabilistic power system analysis.

There are mainly two differences with the framework presented in this thesis.

1. Monte Carlo simulations aim at modelling the *actual* probability distribution of power system configurations in a given time span, while in the learning approach, as we have discussed, it is often preferable to bias very strongly these distributions in a way depending on the particular problem considered.
2. Monte Carlo simulations are basically seeking a precise estimate of the *overall* expected value of an output quantity, whereas in our studies we are more interested in evaluating precisely the effect of some input quantities on the output.

Thus the techniques and tools developed in the context of Monte Carlo simulations could also be useful in the context of our framework. For example, the idea of combining analytical approximations with Monte Carlo simulations in order to reduce variances might be fruitfully exploited in the context of our security assessment framework, to make improved use of available information. At the same time the statistical techniques for data analysis and learning could be very useful in the context of Monte Carlo simulations, for example to assess the sensitivity of the output variable to probability distributions of input quantities or to assess the effect of design alternatives on the expected values of the performance index. One may also imagine that the regions corresponding to the terminal nodes of a decision tree could be used to stratify samples so as to reduce the variance of estimators.

11.6 CONCLUDING REMARKS

In this chapter we have pointed out several major difficulties which arise when we try to generate a representative data base for real systems. One of our objectives was also to make clear that no *universal* one-shot procedure exists - or is likely to exist in the future - to solve this problem. Thus the successful application of the methodologies presented in this thesis will greatly depend on the willingness of power system engineers to inject their knowledge into this process and to compare the resulting security criteria with their own expertise about the problem, yielding an intrinsically iterative cycle of improvements.

This will probably require some changes in present day practices and also some further adaptations of statistical methods. This is to say that apart from some specific cases, there is still a long way to go before potential users get sufficiently familiar with the statistical approaches so as to apply these methods to a large range of security problems. Nevertheless, we note that the techniques proposed in this thesis are somewhere in between the present day *manual* practices and the standard way of applying Monte Carlo methods. Therefore, it will probably be easier to incorporate our statistical methods into system planning studies, where Monte Carlo simulations are already in use for random sampling, and where probabilistic methods have a greater chance of being readily accepted.

On the other hand, one of the main outcomes of our experience is that in order to assess quantitatively security criteria obtained the very close collaboration of engineers in charge of security studies is paramount.

12

Modelling aspects and numerical tools

In this chapter we collect together considerations related to modelling aspects in security assessment studies. In particular, we start by discussing computational feasibility aspects at the data base generation step, and we finish with modelling aspects at the machine learning step.

12.1 SIMULATION MODELS AND METHODS

So far, the type of statistical learning approaches described in this thesis have not yet found actual application to real systems. At best, various feasibility and evaluation studies have been carried out on real systems in order to appreciate the practical pros and cons. The methods have mostly been studied with simplified power system models.

For large-scale power system transient stability, and to a lesser extent voltage stability assessment, computations may be quite time consuming and it might be questionable whether the data base generation and the related numerical simulations are feasible within acceptable response times and realistic computing powers. Actually, we will show that with today's high end workstations which exceed 100MFLOPS computing power on a single processor, and by exploiting trivial parallelism, even the most time consuming simulations among the above become possible with a reasonable number of CPUs.

12.1.1 Voltage security

In the voltage security study described in §14.4, a data base is generated by using the STEC simulator described in [VA 93b]. The corresponding model uses a rather detailed power system model, including the representation of the 90 and 63kV subtransmission network in the study region together with EHV/HV and HV/MV OLTCs. The precise

model used is described elsewhere, but let us merely quote some indicative figures.

1244 buses, corresponding to the complete 400kV network of EDF, the 225kV grid in the buffer and study regions, as well as the subtransmission system within the study region.

1188 branches, corresponding to the EHV lines and transformers.

443 transformers with OLTC, corresponding to the EHV/HV and the HV/MV transformers in the study region.

36 generators, using a static model representing rotor currents and saturation characteristics as well as equilibrium characteristics of voltage and frequency controls.

35 compensation devices, which are automatically switched according to a voltage threshold logic.

Secondary voltage control, which is operating in two independent control regions so as to coordinate reactive resources in order to maintain EHV voltage at a desired profile.

This model can be considered as quite realistic for the study of the mid-term voltage phenomena considered in this research project. To obtain a data base, normal operating states were considered. They were generated by a random sampling approach described in §14.4. On the basis of a sample of 13,513 randomly drawn variants of a base case situation, 5,000 yielded acceptable operating states, whereas the 8,313 remaining ones were rejected mainly due to the divergence of loadflow computations. For each one of these operating states more than 300 attributes were computed and 26 disturbances were simulated with the STEC software. These simulations included the modelling of the secondary voltage control and the 443 tap changer dynamics during several minutes after inception of the disturbance and the determination of a post-disturbance load power margin, for the stable situations. Thus 130,000 simulations were carried out, comprising an important proportion of load power margin computations.

To carry out all the related simulations, four SUN SPARC10 workstations were used in parallel, and the overall elapsed time was approximately one month, corresponding to about 25% of the use of the available CPU time, taking into account other processes running on these systems. The total amount of data generated was about 100MB (compressed) including the 300×5000 pre-fault attributes and the $26 \times 300 \times 5000$ "just after disturbance" attributes for each one of the 26 disturbances.

We note that presently available high level workstations may be up to five times faster in floating point arithmetic than the SUN SPARC10 workstations. Thus, with four such high performance workstations fully dedicated to the data base generation this response time may be reduced to two days, which is quite acceptable in the context of off-line security studies. This means also that with computing hardware which may

be available in the next two years, smaller data bases focusing on a reduced number of disturbances (say 5) may be obtained within less than one hour elapsed time, for this problem. This opens new possibilities for generating or refreshing data bases and security criteria very near to real-time operation, say less than half a day in advance.

12.1.2 Transient stability

Transient stability simulations are known as the most time consuming simulations within security assessment. For example, to compute a single security margin value with respect to transient stability and using realistic¹ modelling would take about one hundred times more instructions than the above computation of a voltage security margin.

This means essentially that with present day technology we must rely on a much higher level of parallelism in order to reduce the corresponding CPU time. However, the required computing power may become available in a very near future, notably due to the wide area interconnection of utility information systems, which opens access to large volumes of inactive (not exploited) computing power.

We notice also that we may expect to see in future control centers and associated computing environments computing powers of the order of 10 GFLOPS and more, which will be exploited only at a small fraction of their nominal power with standard software applications. Thus several GFLOPS will be available without additional cost and could be exploited very systematically for the above kind of simulations. For example, with a computing power of 5GFLOPS it would take less than 30 minutes to run 1,000 transient stability simulations on the Hydro-Québec system. Estimating that it would require about 200,000 such simulations (10 contingencies and 20,000 operating states) to study in detail a large range of topologies of a transmission corridor, this would take about 100 hours of elapsed time.

12.1.3 Coping with model uncertainties

The preceding discussion indicates that using realistic models in the context of systematic large-scale power system security studies becomes computationally feasible with current and a fortiori with future computing hardware. We thus expect to see more evaluation studies in the coming years under these conditions.

Nevertheless, while the type of models and in particular the type of phenomena which are to be taken into account when simulating power system behavior are well known

¹Our orders of magnitude are based on the model presently used at Hydro-Québec for transient stability studies, which considers the 735kV system and all non-radial lower voltage levels, yielding 450 buses, 650 branches, 80 equivalent generators, and 6 equivalent SVCs, all modelled in detail.

for most practical power systems it is not always possible to identify a set of parameters valid for a large range of operating conditions, as may be required to be simulated within the context of security studies.

There may be several reasons for this. For example, in planning studies it is often required to consider hypothetical generation and transmission equipment and some of their parameters are very gross approximations. Similarly, many parameters vary with time due to aging. Another important reason leading to uncertainties is related to unobservable parts of the power system which lead to the use of dynamic and / or static equivalents, which are often unreliable.

For instance, in transient stability studies it may become important to represent neighbor utilities correctly when considering faults at the periphery of a system. Unfortunately, strong competition among utilities tends to impede the exchange of data and measurements, although information technology would make this quite easy to realize. We can foresee that the current trends towards opening the access to the transmission system, as experienced in Great Britain and some other countries, will make this problem much more acute specially in the context of modelling generation equipment characteristics which are of primary importance for security assessment.

Another difficult problem concerns the modelling of the load component [IE 92b]. Here the difficulty is related to the very high number of elementary devices and components in the distribution systems and the fact that only a small part of these are actually observable due to the relatively low number of measurements and signals. Although appropriate static (and to a lesser extent dynamic) equivalent models may be formulated (e.g. active and reactive power as polynomials of voltage and frequency), the major difficulty is due to the fact that the parameters of these models will vary in practice according to time and to the geographical location. In practice however, most utilities use a single constant load model for the major part of the load (say more than 90%) together with some special loads to represent effects of large industrial plants (electrolysis factories, AC/DC converters . . .) which are easier to model than compound domestic and industrial loads.

From the point of view of the machine learning framework, it is interesting to observe that the uncertainties in the power system model can be readily taken into account in a similar way to the free parameters characterizing the state of the buffer region. Since however it is generally difficult to appraise a priori how the model will affect security, it is not always possible to replace the unknown parameters by a single value always yielding conservative results. Thus, a better strategy would be to define random distributions of the unknown model parameters (load sensitivities, external equivalents . . .) and let these vary according to the random sampling procedure.

The machine learning techniques may then be applied in various ways, depending on the type of information desired. For example, if a security criterion, robust with respect

to model uncertainties is sought, the candidate attributes should not provide explicit information about the latter model. On the other hand, if it is desired to assess the quantitative impact of model uncertainties on the security of the system, then some attributes could be used which provide information about this model and the criteria thereby obtained could be compared with the former robust criteria, to assess the impact of the model. This would provide the possibility of making global sensitivity assessments with respect to modelling aspects over a very broad range of conditions represented in a data base.

12.2 PHYSICAL ASPECTS OF LEARNING PROBLEMS

Once a particular security study has been delimited on the basis of considerations discussed in the preceding three chapters, and once a data base providing information about classes of power system states and relevant contingencies has been constructed, several important *modelling* choices have to be made to derive proper security criteria from machine learning techniques. Such modelling issues are discussed below.

12.2.1 Problem decompositions

Although a security study will generally cover a restricted security subproblem, this decomposition may still cover a very wide range of situations and contingencies. For example, the voltage security study in the context of the EDF system covers many different topologies and 26 different contingencies. Similarly, the transient stability study of the Hydro-Québec system, while considering only one of the three main transmission corridors of this system, still considers a very complex security problem since more than 300 different topologies are covered and simultaneous stability with respect to all possible single-line faults in the transmission corridor are considered.

Thus, to obtain good security criteria and also to facilitate their analysis and validation, it is generally appropriate to decompose the complete security problem covered by a data base into a series of simpler, more tractable subproblems. This decomposition may be done by considering either subclasses of power system configurations or subclasses of contingencies, or both.

Power system configurations

This decomposition consists of considering a subset of operating states contained in a data base which a priori are supposed to share some common features for the considered security problem. Generally the criteria used to decompose the data base are provided by major topological characteristics (such as the number of lines in operation in a part

of a transmission corridor, or the number of nodes in an important substation) supposed to influence the kind of contingencies which may become constraining and / or the type of parameters which will influence significantly their severity.

Groups of contingencies

Sometimes it is easier to decompose the problem a priori by considering groups of similar contingencies, or if there are no striking similarities to consider single-contingency security criteria, right from the beginning.

Most often the power system engineer's expertise may suggest how to associate subclasses of power system configurations with groups of contingencies known to be potentially severe for these subclasses.

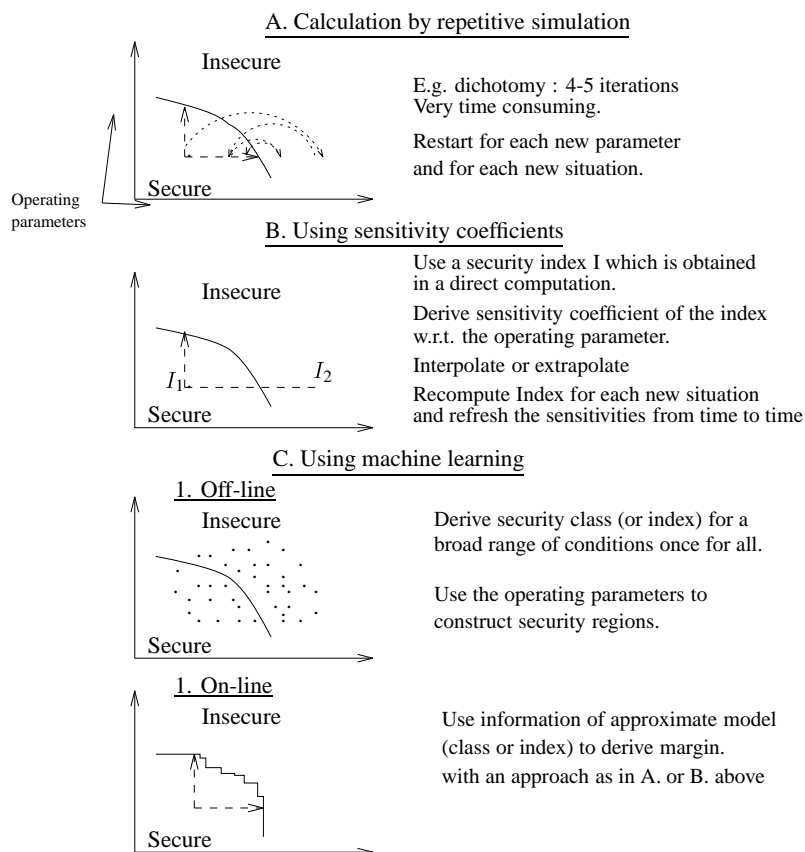
The main interest in the problem decomposition is the possibility to exploit prior knowledge about security assessment. If such prior physical knowledge is rich enough, one may expect to obtain better security criteria, with easier interpretation and validation. Throughout our investigations with the decision tree method reported in the next chapters, it was consistently found that by decomposing an overall problem into sub-problems, simpler and often more reliable trees could be obtained than if too complex complete security problems were considered.

Of course, to increase reliability other means may also be used, such as richer security information in the form of security indices or margins (e.g. see below), or a larger data base, or even more sophisticated attributes; but the problem decomposition approach is a simple and effective means of exploiting existing prior knowledge and providing interesting output information.

12.2.2 Security classes vs margins

Basically the security problems are formulated as a two class problem : the power system in a given state is either sufficiently secure or it is not. In addition to this discrete information, other continuous security *indices* may be determined, such as energy margins or critical clearing times in transient stability, or the voltage security indices proposed by several authors [TA 83, CA 84, VA 91a].

In contrast to these indices, operators use *operating margins* defined in terms of operating parameters, which determine e.g. how much increase in power flow or in system load may be tolerated without the system becoming insecure. Thus, security indices are useful to the operators insofar as they help to derive operating margins, for example through numerical or analytical sensitivity computations. On the other hand, operating margins may also be obtained without calling for the indices, via repetitive security simulations, which are however generally too bulky to be performed in real-time.



In the context of machine learning approaches, it is in principle not required to determine security indices or operating margins while pre-analyzing the security of the generated samples. Indeed, only a precise characterization of each state is required for constructing the security criteria, and if the sample data is rich enough the resulting criteria will implicitly contain information about the margin *in terms of the parameters used as attributes*. Thus, if the attributes are the usual operating parameters, the synthetic security criterion learned may inform about the operating margin for any desired situation, if not directly, at least by the same “dichotomization” approach presently used to derive these margins from repetitive security simulations. The main practical difference is of course the CPU time aspect, since classifying a situation as secure or insecure will be a matter of milliseconds when using the learned criteria, while each security simulation requires a matter of seconds or minutes, even with very simplified models. This is illustrated in Fig. 12.1.

Stated otherwise, the machine learning methods allow us to derive global approximations of the security boundaries which subsequently may be very easily exploited to recover operating margins, even if only a discrete secure vs insecure classification was provided at the learning time.

The above discussion does not however imply that security *indices* would not be interesting to exploit for the design of the security criteria, in the context of the machine learning framework. Actually, within the context of our research we repeatedly found that security indices (and also operating margins) provide indeed very valuable information and may be exploited in practice to increase the flexibility in the construction and validation of security criteria.

For example, in the preceding chapters we have illustrated the use of security indices (CCTs) to distinguish among normal and large errors when assessing the reliability of the security criteria derived from a set of simulations. Further, in the context of classification it may be useful to slightly bias the classification in order to increase the probability of detecting unstable states, and this may be realized easily by exploiting continuous security indices at the learning stage, as we will illustrate in the next chapter. In the same spirit, we have used the hybrid DT-ANN approach and margin regression techniques to increase the reliability of security criteria.

We may conclude that relevant security indices or operating margins may provide richer information and lead to more effective use of the machine learning approach. On the other hand, the off-line determination of security indices leads to higher, but generally not prohibitive, computing times.

12.2.3 Types of attributes

In the preceding discussions we mentioned that in power system operation some particular parameters are privileged attributes because they correspond to the variables which are usually manipulated by operators. These so-called operating parameters mostly correspond to particular power flows (zone import or export, flows through corridors) or power generation reserves in different regions as well as regional load levels. They are used to appraise the overall security situation in terms of operating margins. The values of the latter are generally interdependent and change strongly with the power system configuration.

On the other hand, in terms of decision making and in particular assessing the system security, other physically more appropriate variables may be used, for example to simplify the description of security regions. The choice of these variables is part of the classical representation problem which we have mentioned in chapter 1. Below we will discuss implications of various possible choices of attributes and it is important to realize that whatever the intermediate attributes used to learn a security criterion, in the

end they must be reformulated in terms of the operating parameters normally used in the considered utility.

Controllable attributes

By controllable attributes we refer to those elementary or synthetic parameters whose values may be easily adjusted so as to act on the security of a power system in a given operating context. They may indeed correspond to actual controls, such as voltage set-points or active generation, but they may also correspond to other parameters which may be indirectly adjusted by acting on the former, such as power flows and reactive power generation, provided that there exists a computational tool or a manual approach which can be applied to the corresponding control.

These kind of attributes are more or less equivalent - if not equal - to the basic operating parameters and the translation of a security criterion is more or less trivial. In addition to the usual control variables we include also the logical status type of information describing the topological configuration of the network and load level, which may become controls under particular circumstances.

Ideally, security criteria would be directly expressed in terms of these simple and easy to appraise attributes. However, since the physical relationship between these parameters and security may be quite complex in practice (otherwise security assessment would be a trivial task which is definitely not the case), it may be difficult to derive security criteria which are sufficiently accurate.

Observable attributes

The second level in the degree of sophistication of attributes consists of more complex functions of the power system state and configuration, but which can still be considered to be parameters available in the security assessment environment, and which essentially characterize the situation independently of any hypothesis about a particular contingency.

The most simple parameters may be active and reactive losses and angular spreads. More complex combinations of topology and operating point, such as short-circuit power for example, are available in many control centers. Other, even more sophisticated quantities may involve the computation of internal angles of generators or reactive reserves derived from the current operating point and capability diagrams. Moreover, some standard contingency independent security indicators may provide very valuable information, such as the pre-fault load power margins computed normally for preventive voltage security assessment.

Complex attributes

Attributes start to become truly complex as soon as their definition (and computation) depends on an assumed contingency. For example, in the context of preventive security assessment one may use information about the fault clearing scheme, so as to compute quantities as defined above but corresponding to the post-fault configuration.

Various more or less sophisticated attributes may be thought of and have been proposed in the literature. For instance, in the context of transient stability, attributes have been derived on the basis of Lyapunov functions, such as initial kinetic energy and accelerations, computed immediately after fault clearing.

Several comments can be made.

First, increasing the sophistication of attributes can certainly lead to improved performance in terms of accuracy, however this is always at the expense of a reduction in interpretability and a corresponding difficulty in validation.

Second, in some sense the more sophisticated the attributes the less interesting the information provided by the machine learning approach. At the extreme the attribute may be so sophisticated that it is almost equivalent to the security information which is sought. Then using the learning approach merely reduces to tuning a few thresholds on this “super attribute”. This may be an interesting approach to compare and systematically analyze on the basis of a large sample various relationships among various security indices, but is not generally an interesting avenue for the development of security criteria.

Third, the more sophisticated the attributes the more important the computational involvement to determine their values. For instance, using attributes derived from the Lyapunov direct method may be obtained at the beginning of the post-fault period. This will then require us to simulate the system in the during fault period, for each contingency and for each operating state, which will significantly reduce the computational advantage of using security criteria obtained by a machine learning approach.

Finally, the more intricate the computations required to obtain attribute values the more information we need about the relevant modelling aspects. For example, if we compute attribute values in the JAD state for voltage security assessment, we need to make a hypothesis about the load model; similarly for transient stability assessment. This means that these attributes will implicitly exploit information about the load model, and care must be taken to account for uncertainties on the latter values, in order to avoid overestimating the quality of these attributes.

Part III

APPLICATIONS

13

Transient stability

In this last part we report on simulations we carried out in the context of machine learning approaches. The description is organized in terms of the physical problems and corresponding practical application studies.

13.1 INTRODUCTION

Our research on the application of artificial intelligence methodologies to power systems was initiated some 8 years ago, in the context of on-line transient stability assessment. The objective was to assess what and which kind of AI methodologies could be helpful to solve this highly nonlinear problem, conventionally tackled via long numerical simulations, impossible for on-line applications.

Since experts derive their transient stability knowledge mainly from off-line simulations, it was judged that a machine learning approach could automate this process to a certain extent. In particular, such an approach was expected to be potentially able to exploit off-line large amounts of computing powers, which were starting to become available. This motivated us to identify ID3 as a plausible machine learning method, able to treat large-scale problems; to assess its feasibility, we first adapted and applied it to various “academic” problems.

Of course, our research was closely related to other tentative applications to this problem of pattern recognition techniques, in particular artificial neural networks. However, while the latter methods - as they were formulated - mainly relied on a small number of pragmatic features, our main goal was to stick as closely as possible to the way experts tackle the problem in real life, so as to take advantage of their collaboration and their feedback, paramount for the success of such a method. In turn, this imposed the use of standard operating parameters as attributes and required us to formulate the resulting criteria in as simple as possible manner to make their interpretation accessible to the

experts. It was also deemed necessary to decompose the strongly nonlinear problem of transient stability into simpler subproblems, in order to derive simple and at the same time reliable decision trees. This yielded essentially single-contingency trees; our corresponding investigations are collated in §13.2.

This initial research has shown the credibility of the proposed approach and consolidated the tree building methodology as it is formulated today. The following step has concerned a research project started in the early 1990's in collaboration with the R&D department of EDF; the objective was to assess the feasibility of the approach in the context of the particularly large-scale EDF system. Initially, transient stability assessment was tackled for on-line purposes. But it soon became clear that this method could be interesting within the contexts of planning and operational planning as well; thus the evaluation of potentials and weaknesses and the possible improvements of the methodology concerned a rather broad field.

Note that while simplified dynamic models were used to save computation times, we were able to answer many practical questions, in particular those relating to the specification and generation of a data base, and the improvement of the quality of decision trees to reduce non-detections of unstable situations. Later on, the research was extended to multicontingency decisions trees and considered compromises between these and single-contingency ones. These investigations are reported in detail in §13.3.

Finally, a second research project was started in 1992, in collaboration with the operation planning and control center teams of Hydro-Québec, aiming to assess the decision tree methodology in the context of their system. The long-term objective was to provide a tool for the operational planners, by allowing them to determine in a systematic way the operating guidelines for their system concerning the transient (and also mid-term) stability limits. It was thus hoped to advantageously replace the presently used methods. The first, promising results obtained within this research are reported in §13.4 together with the projected future research.

Having gradually gained confidence in the methodologies of data base generation and of learning methods, we started investigating complementary features of statistical and neural network methods; they led us to make some additional tests with the data bases generated for the EDF and Hydro-Québec systems. They are reported where appropriate.

13.2 ACADEMIC STUDIES

Three studies were carried out on three different academic systems of growing size, a simple One-Machine-Infinite-Bus (OMIB) system, an outdated 14-machine version of the Greek system and a 31-machine North-American system.

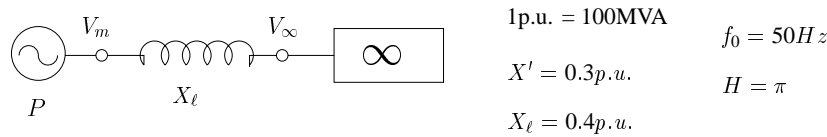


Figure 13.1 OMIB system

These simulations, described in [WE 90a, WE 91a], allowed us to obtain a good understanding of the ID3 decision tree induction method initially used; they also validated the adaptations made on the automatic optimal threshold search for numerical attributes, and on the stop-splitting criterion [WE 89a].

Below we briefly describe the three study systems and summarize the main outcomes, in order to provide some insight into the research process gradually leading to the present formulation of the decision tree method. Thus, in this retrospective, description of detailed quantitative results will be avoided.

13.2.1 Study systems and data bases

OMIB system

The simple OMIB system is represented in Fig. 13.1. The single machine is represented by a classical model, i.e. by constant mechanical power (equal to the prefault electrical output active power) and constant electromotive force (equal to the its prefault value) behind transient reactance X' , which models the effect of the actual direct axis transient reactance X'_d and short-circuit reactance of the machine's step-up transformer. The transmission system is modelled by a constant equivalent short-circuit reactance X_l and infinite inertia. The operating point of the OMIB system is defined by the prefault active generation of the machine and the voltage magnitudes at the machine EHV bus V_m and at the infinite bus V_∞ [WE 90a].

The learning and test samples were generated according to uniform and independent sampling of P , V_m , and V_∞ in the following intervals (in per unit) :

$$\begin{aligned} V_m &\in [0.9 \dots 1.1] \\ V_\infty &\in [0.9 \dots 1.1] \\ P &\in [0.3 \dots 0.7]. \end{aligned}$$

Transient stability was assessed with respect to a lateral¹ three-phase short-circuit ($3\phi SC$) at the machine EHV bus. The CCT was computed by the equal-area criterion [PA 93].

¹A "lateral" fault is a fault with identical prefault and post-fault configurations.

Decision trees were built on the basis of the standard ID3 method augmented with the optimal threshold search algorithm. The candidate attributes used were the above three independent variables. Two-class decision trees were grown for various sizes of learning sets, using various CCT thresholds to define the stable and unstable class. Several naive stop-splitting rules were experimented with, which showed the need for a rule combining both apparent information quantity and learning subset size. The test set error rates varied between 2% and 3%. These simulations allowed us also to appraise the biased character of the resubstitution error rates computed on the basis of the *LS*.

While this example was clearly too simple to allow extrapolations to real large-scale systems, it had the advantage of enabling us to generate random learning and test sets very efficiently and with a great flexibility. We gained some experience with the decision tree building method and in particular with the dependence of the decision trees on the random nature of the learning set. We have learned that this randomness of trees did not disappear with a growing learning set. Due to the lack - at that time - of an effective stop-splitting rule, increasing the learning set size yielded an almost proportional increase in tree complexity and, while the attributes chosen and their thresholds tended to stabilize at the top-nodes, the deeper test nodes kept being rather sensitive to the random samples.

Thus, there was a true need for a stop-splitting criterion capable of preventing the method developing nodes on the basis of too small samples. Our first idea was to stop splitting at nodes corresponding to a too small learning set, in terms of a threshold N_{min} on the number of states. Later, guided by the discussions in [KV 87] the idea came to use a hypothesis test for this purpose.

Notice that our initial motivation for introducing the hypothesis test was not to improve the generalization capabilities of the trees nor to identify noisy attributes, but rather, to prevent the method from developing nodes which were overly dependent on the random nature of the learning set. Only later did it become apparent that this strategy could also identify noisy attributes and improve significantly the reliability of trees, in particular when the classification problem becomes non-deterministic because of missing information in the attributes.

Greek system

To confront the methodology progressively with more realistic systems, we applied it to a 14-machine version of the Greek EHV system. For this study, we considered a complete 150kV and 380kV system representation, comprising 92 buses and 112 lines and studied the transient stability limits of an important power plant with respect to a lateral three-phase fault at its EHV bus. Thus, this was essentially equivalent to the above OMIB simulation, where the infinite bus is replaced by the complete system model.

The data base was composed of 201 different operating states of fixed base case topology, by combining in a deterministic procedure variations of active power generated in the study plant, with variations of active and reactive generations and load nearby the study plant [WE 90a]. Due to the very small size of the data base (at that time, we used a DEC 20 computer of about 2 MIPS which limited somewhat the possibility of generating a large data base) we built decision trees on the basis of the complete data for two, three and four classes. Ten different candidate attributes were proposed concerning the active and reactive power generations and the load and voltage near the faulty bus. In the context of these simulations we introduced for the first time the improved hypothesis test based stop-splitting criterion described in §3.4.4.

To evaluate the reliability of the obtained decision trees with respect to unseen situations, we have used the *leave-one-out* cross-validation scheme. This procedure has the advantage of producing unbiased error estimates without requiring an independent test set (see §2.5.7). The obtained error rates varied between 5.0% for the two-class problem and 9.5% for the four-class problems.

The main conclusion was that the complexity and error rates of the decision trees increased progressively with the number of security classes. The second main outcome was that the method was able to identify among the candidate attributes a subset of most discriminating ones, and the resulting trees were able to provide interesting and interpretable physical information. Thus, the method was able to exploit the more or less local characteristics of a given security problem. This motivated further, more systematic investigations on a larger power system model.

31-machine system

In the meanwhile, the available computing powers had grown enough to allow tackling larger power systems and to initiate systematic studies with sufficiently large data bases, providing representative independent test sets to evaluate the resulting decision trees. We have thus considered the 31-machine system described in [LE 72], composed of 128 buses and 253 EHV lines (345kV and 765kV). It is an equivalent system of an (unknown) North-American utility and its interconnection. It was deemed sufficiently large to provide interesting simulation results and sufficiently simple to avoid unnecessarily bulky computations.

Further, to provide an unbiased estimation of the method we have generated a single *global* data base, independently of any fault specific considerations (as opposed to the fault-dependent data base constructed for the Greek system). Admittedly, we would not normally advocate this method (cf. the discussion of chapter 11); nevertheless, this blind procedure provided an unbiased evaluation of our methodology. Had we introduced fault specific considerations, we would have made the conclusions depend on the quality of the physical knowledge injected in the data base generation procedure, whereas the purpose was to evaluate the knowledge which could be acquired via the

learning method. The other reason for generating a global data base, by screening a broad range of power system situations with more or less independent changes in the power flows and voltage distributions, was to attempt to represent all major effects. Thus, it was possible to check the local nature of single-contingency transient stability limits expressed in terms of the static prefault parameters.

The data base was randomly generated on the basis of plausible scenarios, corresponding to various topologies, load levels, load distributions and generation dispatches. Hereafter we sketch the way used to generate them, to analyze them from the transient stability point of view, and to build the attribute files [WE 91a]. The buses of the power system were grouped into zones and the operating states composing the data base were generated *randomly* according to the following *independent* steps.

1. Topology. Selected by considering base case topologies, single outages of a generator, a load or a shunt reactor, and single or double line outages. The outaged elements were selected randomly.
2. Active load. The total load level was defined according to a Gaussian distribution ($\mu = 32GW$ and $\sigma = 9GW$). It was distributed among the zones according to the random selection of participation factors, then among the load buses of each zone proportionally with respect to their base case values. The reactive load of each bus was adjusted according to the local base case power factor. This resulted in a very strong correlation of the loads of the same zone, and a quite weak one among loads of different zones.
3. Active generation. In a similar fashion the total generation corresponding to the selected load level was distributed among the zones according to randomly selected participation factors, then among the generators of each zone according to a second random selection of participation factors. Thus, neighbor generators were less correlated than neighbor loads. The reactive generations were obtained by a load flow calculation; the voltage set-points were kept constant.
4. Load flow calculation. To check the feasibility of an operating point and to compute its state vector, it was fed into the load flow program, and accepted if the latter converged properly. A total of 2000 states were accepted corresponding to 90% of the generated states.
5. 31 × 2000 approximate CCTs. For a $3\phi SC$ at each one of the 31-generator buses the CCTs were computed using the very fast extended equal-area criterion [XU 88, XU 89]. This gave us good information about the relative severity of these contingencies in relation to the states represented in the data base and allowed us to select three “interesting” ones for detailed investigations. The CCTs of the latter were determined by the SBS method using the classical simplified transient stability model.

Table 13.1 *Tree features and number of classes. Adapted from [WE 91a]*

Gen. bus #	2 classes			3 classes			4 classes		
	# \mathcal{N}	$P_e(\%)$	#A	# \mathcal{N}	$P_e(\%)$	#A	# \mathcal{N}	$P_e(\%)$	#A
2	7	2.27	3	17	5.67	4	27	9.60	7
21	9	3.73	3	17	3.60	4	25	8.40	6
49	5	1.53	1	9	4.33	2	15	7.53	3

6. 300 × 2000 attribute values. These comprised zonal statistics on loads, generation and voltage, voltage magnitudes at all buses, voltage angles at important buses, active and reactive power of each generator, and topology information as well as power flows.

The diversity of the above data base allowed us to investigate a diverse set of aspects discussed below within the context of a diverse set of security problems. This enabled us to gain confidence in the methodology which in turn motivated us to carry on with our investigations on real systems.

13.2.2 General trends

Below we give an overview of the general trends observed from the above “academic” studies for the main parameters of the decision trees, namely their accuracy and complexity. It is important to note that since the learning and test sets are random samples, the reported tendencies describe the mean expected behavior, and it is possible that in a particular situation a slightly different behavior would be observed.

However, the indicated characteristics have been determined on the basis of a large number of simulations (more than 1300 decision trees; three different power systems; hundreds of different candidate attributes; more than 100 different classification problems). Furthermore, they have been very systematically confirmed by the subsequent simulations on real life power systems described in the sequel.

Number of classes

Table 13.1 summarizes typical tree characteristics as influenced by the number of security classes. The trees were grown for three different faults of the 31-machine system on the basis of 500 learning states and 81 static candidate attributes. The value of $\alpha = 0.0001$ was used in the stop-splitting rule. This value is further discussed below. # \mathcal{N} denotes the total number of nodes of the trees and #A its number of selected test attributes among the 81 candidates. P_e denotes the test set error rate, estimated on the basis of the 1500 test states not used to build the trees.

We notice that overall the trees remain quite simple and reliable. The number of nodes

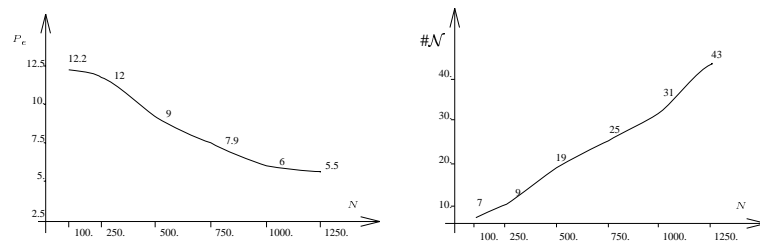


Figure 13.2 Tree features and number N of learning states. Adapted from [WE 91a]

is about proportional to $m - 1$ (m being the number of classes). The error rate becomes large only for four classes; however, this is compensated by the fact that the errors of four-class trees are less dangerous since they happen in general among adjacent classes. The number of selected attributes remains very small and (which is not apparent in the table) the attributes selected for different faults are quite different. This confirms the fact that the trees are able to exploit the local characteristics of each transient stability problem.

Influence of the learning sample

Figure 13.2 shows the typical effect of the size of the learning set on the complexity and reliability of the decision trees.

The trees were built for a four-class problem for the 31-machine system. We observe that their complexity and accuracy increase steadily with the learning set size. At the same time, the number of selected attributes is found to increase from 2 to 11. Thus, the more information provided to the tree induction algorithm the more detailed the information it will be able to represent in the derived decision tree. It is also interesting to notice that for small and moderate sample sizes the decision tree characteristics may strongly depend on the random nature of the learning set.

On the effect of α

Figure 13.3 shows the typical effect of the pruning parameter α used in the stop-splitting rule. Recall that a value of $\alpha = 1.0$ amounts to growing the tree completely, so as to classify correctly all the learning states, whereas the theoretical value of $\alpha = 0.0$ would amount to producing a trivial single node tree.

Each point of each curve provides information of the mean relative size and error rate of 12 different trees corresponding to 4 different learning samples and 3 different contingencies. The curves show that for a two-class tree very small values of α ($\approx 10^{-4}$) tend to provide a very good complexity vs reliability compromise. Indeed, the tree

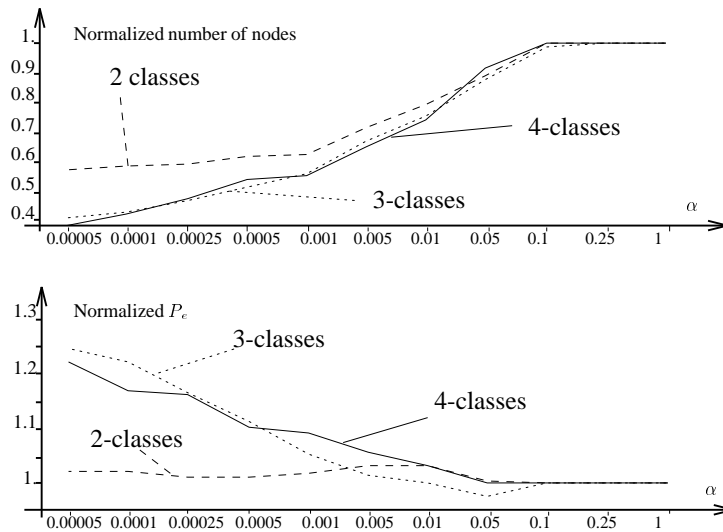


Figure 13.3 Tree features and pruning parameter α . Adapted from [WE 91a]

size is less than 60% of the completely grown tree, whereas its reliability is close to optimal. On the other hand, when the number of classes increases, the optimal values of α tend also to increase. For three or four classes, the complexity of trees decreases more quickly with α and a value of $\alpha \approx 10^{-3}$ seems to be appropriate. Notice that the variation of the test set error rates expressed in the curves is of the same order of magnitude or smaller than the standard deviation of these estimates; thus, the slight increase in test set error rate systematically observed is below the significance level.

These results were confirmed by the very large range of simulations carried out on a very diverse set of problems. A discussion of this behavior is provided in [WE 92b, WE 93h].

Type and number of attributes

Another investigation concerned the effect of the candidate attributes on the tree characteristics. This concerned the so-called masking of attributes selected by the method, in order to assess the degree of complementarity of attributes and the effect of adding new attributes to the candidate list. Since the decision tree induction algorithm is able only to optimize locally at each test node, it is possible - but not very likely - that masking a selected attribute may actually improve the resulting tree, and conversely adding new attributes may also lead occasionally to a degradation of the tree quality.

However, in the very large majority of situations the expected "normal" behavior is observed, and generally the above abnormal variations are rather marginal when they are observed. In order to provide a more detailed assessment of the decision

trees, the detailed information about score measures and information sharings among attributes illustrated in Tables 3.12 and 3.13 have been developed and integrated in the tree building software. In particular, this allowed us to assess the complementary or correlated nature of candidate attributes, for the real life problems discussed below.

Error rate estimates

In the context of the simulations with the 31-machine system the cross-validation estimate was compared systematically with the test set error rate. It was found that it may underestimate (and also overestimate) significantly the test set error rate. This led us to reject this method for our later simulations.

Computational aspects

In the above simulations, we used our own software implementation of the TDIDT method written in CommonLisp, that we found largely efficient. We have provided earlier some comparative performance figures, for real sized problems. In the context of the above academic research, we have checked that the computational complexity of the learning algorithm is slightly super-linear in the number N of learning states and slightly super-linear in the number of candidate attributes. Observe that the “theoretical” upper bound of “ $n \times N \log N$ ” does not take into account the effect of garbage collection and swapping overheads which may become more important for a larger number of attributes.

13.2.3 Discussion

The above investigations took about 4 years in order to understand, develop and evaluate a new methodology for power system security assessment. It crystalized into what we have called DTTS for decision tree transient stability method, since initially this method was applied to transient stability assessment. This research included a bunch of orthogonal investigations concerning the use of the decision trees and in particular the definition of distances in the attributes space, which we do not report here for the sake of conciseness [WE 88, WE 90a].

We have already mentioned that these investigations are not sufficient to assess the practicality of this kind of approach within a particular power system and a given physical problem. In fact, in the context of learning methods we must be very cautious to avoid extrapolating unduly from one problem to another : a given method may work very nicely on the $k - 1$ first problems and fail on the k th one. Nevertheless, the investigations have shown the systematic character of the technique. They also have shown that to handle a new problem the main task is the proper definition of a data base; the subsequent application of the decision trees will be rather systematic even if the

physical characteristics described in the data base are very different. Thus, the practical feasibility of this method for real systems relies mainly on the proper generation of data bases and on the validation by experts of the derived criteria.

These and other important practical questions were considered in the simulations described below.

13.3 EDF SYSTEM

When we started our research collaboration with EDF, there were many open questions about the practical feasibility of the decision tree approach. Some of these questions were of a very general scope, others were specifically linked with transient stability in the context of the EDF system. Before discussing the particular test bed used for our research, let us quote the most important questions initially considered within the basic single contingency DTTS method. In particular we quote the following.

Is it possible to exploit and adapt the decision trees to take into account the strong effect of topology on transient stability ?

How can we generate sufficiently rich data bases and in particular obtain a sufficient number of unstable situations, given the very small actual clearing times of the protection system ?

Given the above indications, is it possible to build sufficiently reliable trees, on the basis of a reasonable number of learning states, say at most several thousand states ?

In the course of the research other additional questions appeared to be very important, concerning the impact of the type of candidate attributes, pragmatic quality aspects and multicontingency considerations. In particular we quote the following.

What is the quantitative impact of the type of candidate attributes on tree complexity and reliability and which appropriate compromises may be identified ?

How can pragmatic quality measures be defined taking into account the different kinds of classification errors and how can we adapt the decision tree induction method so as to improve this pragmatic quality, in particular so as to reduce the number of non-detections of unstable situations ?

What kind of global or contingency dependent multicontingency information is required for on-line operation ?

How can appropriate groups of contingencies be identified to be efficiently treated by a single tree ?

In addition to the above practical questions, the data bases constructed for the EDF system were also exploited in order to make various theoretical and comparative studies, which are indicated below in §13.3.6.

Finally, the investigations of this long-term research project contributed to gain further confidence in the *practical* feasibility of the framework presented in this thesis and allowed many improvements whilst making the methodology more practical. In particular, they yielded the data base generation approach described in chapter 11 and led to improving the pragmatic decision tree quality in order to identify and reduce the dangerous errors. On the other hand, our close collaboration with the operational planning and planning departments of EDF has allowed us to acquire a better understanding of practical needs and thence of potential applications of our methodology.

13.3.1 Study system and data base description

To answer the above questions a particular test problem was considered. It concerns the stability assessment of an important nuclear power plant. It was chosen on the basis of available prior experience along with a preliminary study for screening a broader region.

All in all four different data bases were generated throughout this research. To provide insight into the iterative “generate and test” nature of this process, we describe in detail the options concerning the two first data bases, used in the studies reported in §§13.3.2 - 13.3.4, and give some indication of the main differences of the two other data bases used in the simulations described in §§13.3.5 and 13.3.6. The reader not interested in these details may skip them, and read only the description of the base case conditions.

The results obtained within this study are described and discussed in refs. [WE 90b, WE 91d, WE 91e, AK 93, WE 93a, WE 93d].

Base case conditions

The considered system is an earlier version of the EDF system formerly used for operation planning studies. It encompasses the complete 400 kV grid of the French system as well as the most important part of the 225 kV network, yielding a 561-bus / 1000-line / 61-machine system. Equivalent representations were used for the surrounding European interconnections (Germany, Switzerland, Italy, Spain and Belgium). The overall generation produced by the 61 (equivalent) machines corresponds to about 60,000 MW of national generation and 50,000 MW of external equivalents. Its one line diagram is sketched in Fig. 13.4.

The case study concerns the stability assessment of an important plant situated in Normandy (North-West part of France). This *study plant* was selected via a preliminary

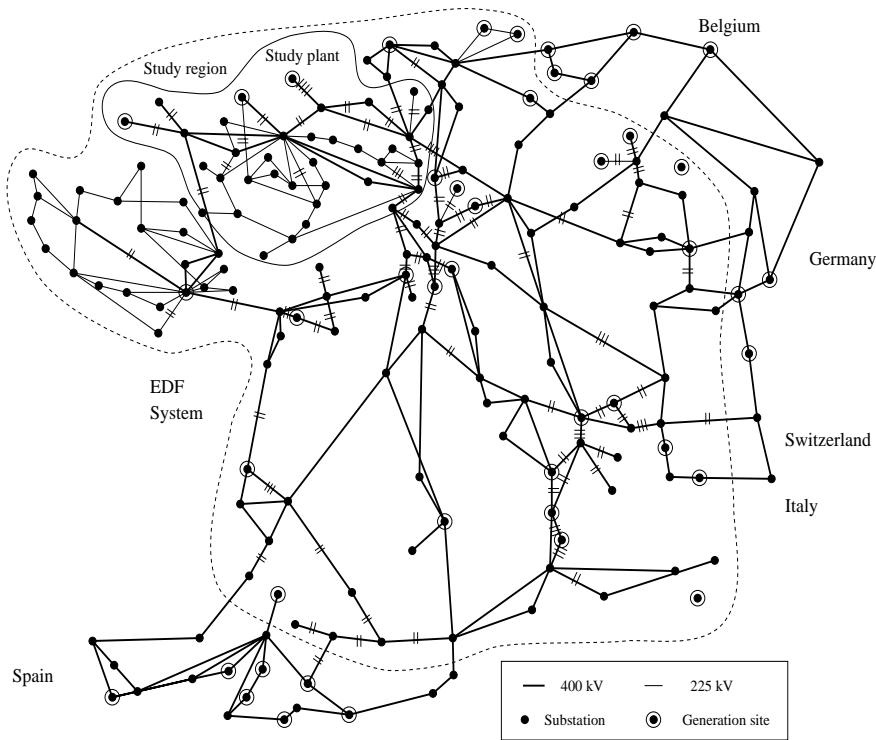


Figure 13.4 One-line diagram of the EDF system

investigation of 60 different contingencies at the 400 kV level for 9 different operating states. Figure 13.5 describes its substation and immediate neighbors at the 400 kV level.

The data bases were generated from a base case via modifications described below. They essentially concern the “study region”, but of course all load flow and stability computations were run on the entire system. This study region presumably encompasses all components liable to influence the stability of the study plant. It was determined by EDF engineers in charge of stability studies. Interestingly, it was also identified, in an independent way, using the “Combined Electromechanical Distance” approach [BE 91b].

The study region is composed of three large power plants (the study plant is the plant number 3) along with the surrounding substations, and about 60 lines at the 400 kV and 225 kV levels. The overall installed generation capacity of these plants is about 10,000 MW and the base case load is approximately of 5,600 MW (corresponding to winter peak load) shared among 42 different load buses.

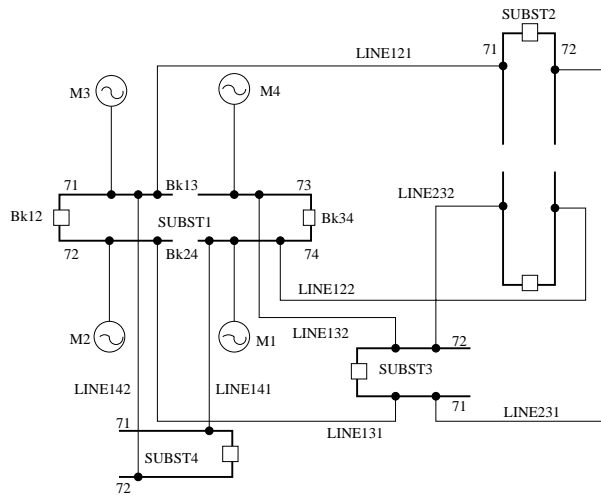


Figure 13.5 One-line diagram of the study plant substation. Adapted from [WE 93d]

Initial data base

The primary objective was to obtain a sufficiently rich data base, which at the same time contains plausible operating states of the region and covers as much as possible weakened situations. For this purpose, a certain number of independent variables, liable to influence the system stability were defined, concerning both topology and electrical status. For each variable a prior distribution was fixed on the basis of available statistical information about the usual situations, so that all interesting values would be sufficiently well represented in the data base. Moreover, to exclude unrealistic situations, constraints were imposed on values taken by different variables. 3000 operating states were thus randomly drawn, their stability was assessed and the values of various types of candidate attributes were computed (see §13.3.3).

For each state the following tasks were executed :

1. definition of the load level in the study region, the topology of the 400 kV regional network, local active and reactive generation scheduling;
2. building of the load flow and step-by-step data files;
3. load flow computation and feasibility check;
4. appending of the states attribute values in the corresponding attribute files;
5. computation of the CCTs of the considered disturbances via step-by-step simulations.

The following three severe contingencies have been identified, classified in a decreasing

order of criticality.

Busbar fault : three-phase short-circuit located on the busbar section 71 in substation 1; cleared by opening the lines 121 and 142, tripping machine M3 and opening the breakers 12 and 13 to isolate the faulted section.

Double-line fault : 2 simultaneous three-phase short-circuits near sections 71 and 74 of substation 1, on the lines 141 and 142; cleared by opening both lines.

Single-line fault : a three-phase short-circuit on line 131 near the busbar section 72; cleared by opening this line.

The CCTs of the above contingencies were computed by a standard step-by-step program.

The main parameters used to draw randomized prefault operating states were topology, active generation / load, voltages, as outlined below.

Topology. It was defined by the 18 regional 400 kV lines and by the number of nodes in the 4 substations represented in Fig. 13.5.

Line outages. 10% of the states were generated with a complete (i.e. base case) topology, 50% with a single line outaged, selected randomly among the 18 lines of the region; the remaining 40% corresponded to the simultaneous outage of two “interacting” lines : 40 pairs of interacting lines were defined, consisting of lines either in parallel in a same transmission corridor, or emanating from the same bus.

Study plant substation. Substation 1 was restricted to 1 node (breakers 12 and 34 were closed) if a single generation unit was in operation; otherwise it was 50% of the time configured with 1 node and 50% with 2 nodes, and so are substations 3 and 4. Substation 2 was 90% of the time configured with 2 nodes.

Load. The total regional active load level was drawn according to a Gaussian distribution of $\mu = 3500$ MW and $\sigma = 1000$ MW. It was distributed on the individual load buses proportionally to the base case bus load. The reactive load at each bus was adjusted in order to keep a constant reactive vs active power ratio ($\frac{Q}{P} \approx 0.15$).

Active generation. The active power generations of the three power plants were defined independently, according to the following procedure.

1. Unit commitment. Given a plant, the number of its units in service obeyed a plant specific distribution. Thus, for plant 1, 0 to 4 machines may be in service, according to a priority list, and with uniform probabilities. For plant 2, the 4 following combinations were used : no unit in operation (10%); either unit 1 or unit 2 in operation (30%); both units 1 and 2 simultaneously in operation

(60%). For the study plant, 10% of the cases corresponded to a single unit in operation, 20% to 2 units, 30% to 3 units and 40% to 4 units; the units being committed were drawn randomly, under the restriction of an as uniform as possible share of the generation on the two nodes of the substation 1, if the latter was configured with 2 nodes.

2. **Active power generation.** Once again, to maximize the interesting cases the rules were plant specific. For plants 1 and 2, a random generation was drawn in the interval of the global feasibility limits of its operating units. For the plant 3 of Fig. 13.5, the first two units in service were rated at their nominal power of 1300 MW each, the next two were rated according to a random number drawn in the feasibility limits of the units. This enabled the generation of a maximum number of highly loaded situations, without losing information about intermediate, albeit less realistic cases.

Voltage profile. A simple strategy was used to produce sufficiently diverse voltage profiles, near the study plant. The EHV setpoint of its operating units was drawn randomly in the range of 390 kV to 420 kV, independently of the local load level. Furthermore, the voltage setpoint of plant 1 (the next nearest one) was drawn in the same range and independently. This produced a quite diverse pattern of reactive generations and flows in the study region (see below).

The above randomized modifications of the base case provide, via load flow computations, the 3000 operating states of the data base. The diversity of situations covered by them is reflected by the statistical distributions portrayed in Figs. 13.6 corresponding to key variables of the study plant. Figure 13.6a sketches the total active generation of the plant : the vertical bars at 1300 MW (resp. 2600 MW) represent the number of operating states (OSs) with one (resp. two) units in operation, rated at their nominal power; the bars between 2850 MW and 5200 MW represent the OSs where at least three units are in operation, two of which are rated 100% and the remaining at an intermediate level, ranging from 250 to 1300 MW. Figure 13.6b shows a typical distribution of the reactive generation of a given unit of the study plant; its Gaussian shape nicely reflects the regional load pattern.

Figure 13.6c shows the multimodal CCT distribution of the busbar fault. The OSs around 235ms correspond to the great majority of “normal” situations; those near 0ms correspond to topologically exceptionally weak OSs; those above 350ms to “unusually” stable states. Figure 13.6d illustrates that the sole attributes P-M4 and V-M4 are unable to properly separate the stable (●) and unstable (+) states, despite the important role played by these attributes in the various decision trees shown below.

Incremental data base

In order to investigate the possibility of improving the decision trees by expanding a particular subtree, an incremental data base was generated for a subrange of situations

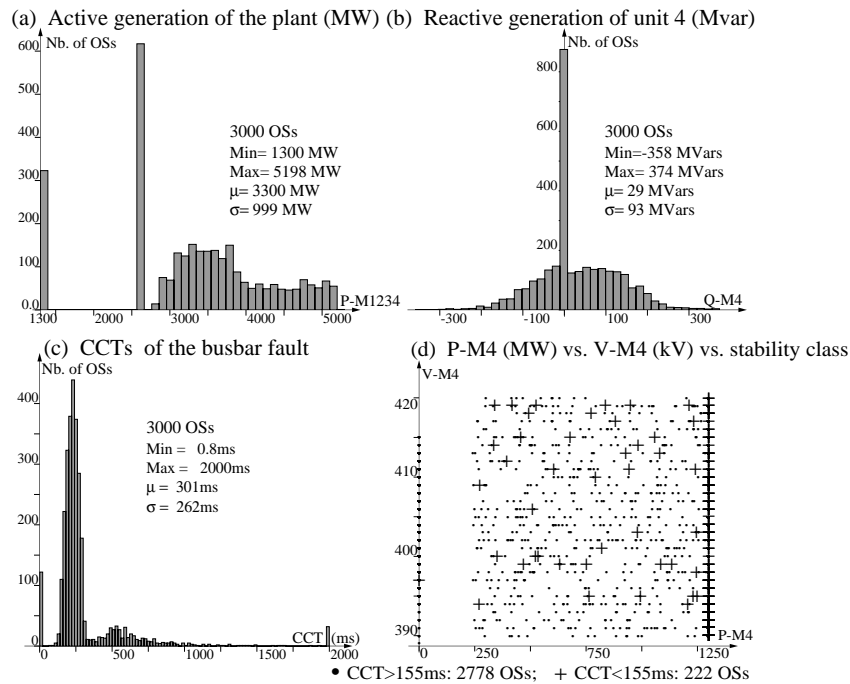


Figure 13.6 Statistics relative to the study plant. Adapted from [WE 93d]

corresponding to the constraints defining a particular deadend node of a tree built on the basis of the above “global” data base. This resulted in an additional set of 2000 situations corresponding to a single-node configuration of the study plant substation and the lines 132 and 141 systematically taken out of operation. The related investigations are reported below in §13.3.4.

Multicontingency data base

A third data base has been generated in order to investigate multicontingency aspects, and, in particular, the complete set of contingencies which could possibly constrain the operating state of the power plant. Seventeen such potentially harmful contingencies were preselected by the operation planning engineers; they are detailed below in §13.3.5. To take advantage of the experience gained with the first data base, a new set of 3000 operating states were generated on the basis of slightly different specifications. The main differences in the random sampling procedure are the following.

The study plant substation was kept in a constant single-node configuration. This simplified the stability assessment.

The line outages were restricted to the 6 outgoing lines. The probabilities were respectively of 0.1 for no outage, 0.35 for one line outage, 0.35 for two-line outages and 0.2 for three-line outages. This tended to weaken the prefault situations and hence to increase the number of unstable scenarios.

The active regional load was drawn according to a Gaussian law of $\mu = 2500$ MW and $\sigma = 800$ MW. This yielded a lower level of reactive generations in the study plant and thus also weaker situations from the transient stability viewpoint.

The active generation of the units in operation in the study plant were generated according to triangular distributions instead of the uniform distributions used above. The objective was to increase the diversity of high generation situations by creating more such situations, and by distributing them on a slightly larger range of values.

Constant topology data base

Finally, a simplified data base was constructed corresponding to a constant base case topology. In this data base, the number of units in operation in the study plant was however kept variable while the total plant generation was distributed according to a triangular distribution and shared uniformly by the different units in operation.

This data base was mainly exploited in the preliminary investigations of the hybrid DT-ANN method reported in [WE 93a] and in §13.3.6 below. Only a single lateral fault in the study plant substation was considered.

Discussion

The above description illustrates the iterative “generate and test” nature of the development of an appropriate data base. All in all, the successive data bases generated for the EDF system correspond to 11,000 different prefault operating states and 18 contingencies. A total number of 65,000 CCTs have been computed and about 1,300,000 attribute values !

Incidentally, we mention that in addition to the investigations on the DTTS method, the data bases were extensively exploited in another parallel research project concerning the development of an improved version of the DEEAC method [EU 92, XU 92, XU 93a, XU 93b]. This is a typical byproduct of the data bases generated within the machine learning framework.

13.3.2 General parameters

A main goal of the first broad investigation carried out on the EDF system was to determine appropriate values of the parameters of the DTTS method. About 40 DTs

were thus built for the following range of parameters :

- learning set size : $N=500$ and 2000 ;
- values of α : $\alpha = 10^{-1}$ and 10^{-4} ;
- classifications : 11 different classifications were considered, relative to the three faults, and different numbers and values of thresholds defining the classes.

Stability classes

For each contingency, various classifications were considered, for two-, three-, four-class trees and various threshold CCT values.

Threshold at the “actual clearing time”. As indicated in Fig. 13.6c, despite the extreme severity of the contingencies, only a small proportion of the learning states were found to have a CCT lower than the “actual clearing time” (equal to 155ms for a busbar fault and 100ms for line faults). An important question is therefore : how does this imbalance between stable and unstable states influence the DTs?

Threshold at the median of the CCT distribution. If the threshold is taken as the median of the CCT distribution (e.g. 235ms for Fig. 13.6c), the class boundary is situated in a very dense region of the attribute space. Two competing effects are thus expected : (i) more learning states near the class boundary provide richer information on the attribute vs stability relation; (ii) more test states near the class boundary will yield higher error rates.

Multiclass trees. Four-class trees were built by using three threshold values including the actual clearing time and two larger values. In addition, for the busbar fault two-class trees using a threshold of 350ms, and three-class trees using two threshold values of 155ms and 350ms have also been built.

Candidate attributes

In §13.3.3 we describe the 13 lists of candidate attributes of growing complexity (all in all 160 different attributes) which have been proposed for the decision tree method. They characterize the study region by its topology, its electrical status, and/or the combined effect of the two, via more or less complex combinations. These candidate attributes may be classified into one of the three following categories, according to the type of information they convey and the type of applications that the resulting decision trees could handle in practice.

Controllable attributes include the regional load level, voltage set-points and active generation of units, as well as the topological variables. The corresponding DTs would yield straightforward analysis and control tools.

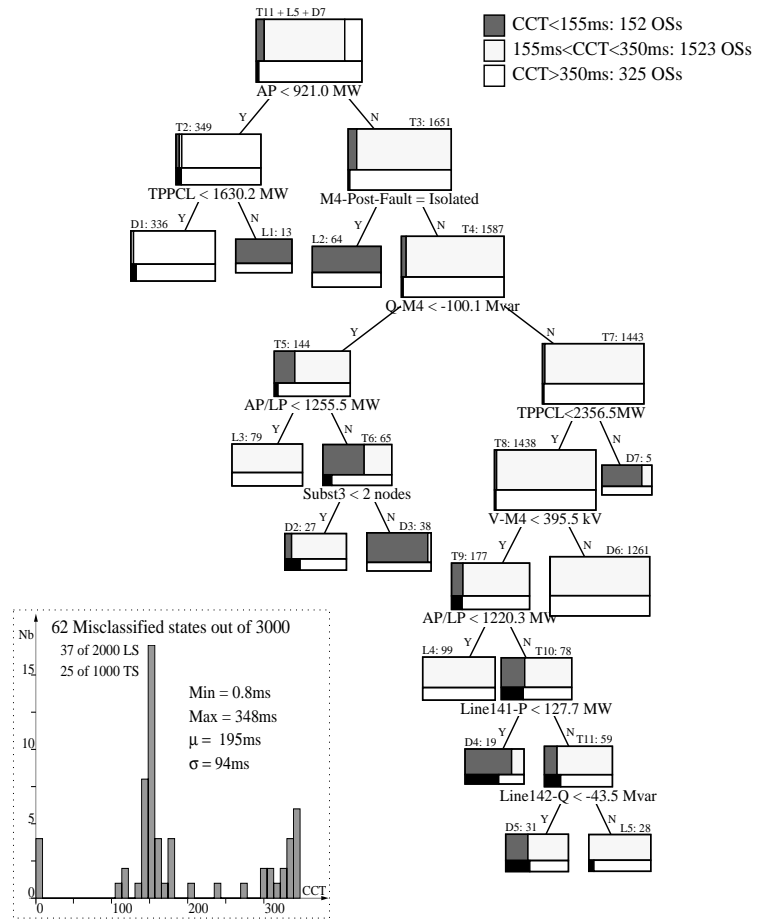


Figure 13.7 3-class DT. Adapted from [WE 93d]

Observable attributes include in addition dependent variables such as reactive generations, power flows and/or relative phases. Corresponding trees would require auxiliary post-processing tools to allow control applications but their information could still easily be appraised by operators.

Complex attributes may take into account any kind of information concerning the fault location and clearance scenario as well as prefault operating state information. These may be combined to yield complicated “ad hoc” attributes, which at the expense of a lesser intelligibility may sometimes increase significantly the reliability of the trees.

The trees have been tested on the basis of 1000 test states, not considered during the learning stage. Figures 13.7 and 13.8 portray two such representative trees correspond-

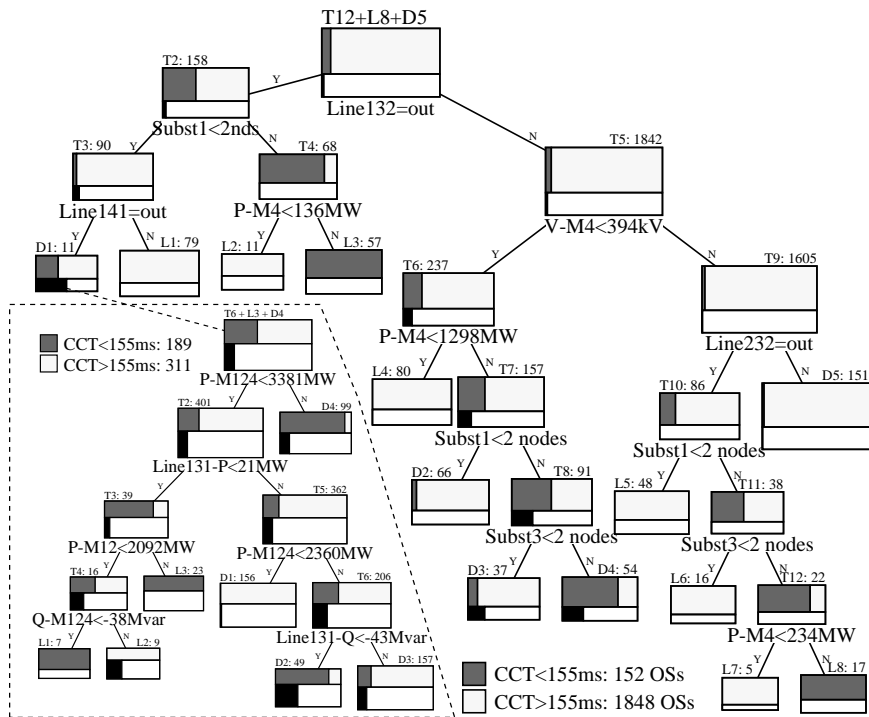


Figure 13.8 DT1 of Table 13.3 subtree for node D1. Adapted from [WE 93d]

ing to the busbar fault.

Discussion

The main outcomes of this investigation are outlined below.

“Optimal” parameters. A good compromise of complexity vs reliability is achieved with $N = 2000$ and $\alpha = 10^{-4}$. By using such low values of α , one dramatically reduces the tree size without deteriorating its reliability, and often improving it : e.g. for DT11 of Table 13.3, the size reduces from 63 to 19 nodes for α decreasing from 0.1 to 10^{-4} ; at the same time, its error rate improves (slightly) from $P_e = 1.5\%$ to 1.3%. Even more drastic complexity reductions are observed in the case of 3- and 4-class trees.

Effects of topology and electrical status. The method was able to formulate in an effective and transparent way the combined effect of topology and electrical status on the system stability. This is illustrated by the three-class tree represented in Fig. 13.7. The selected test attributes are of the following three types : **topology** :

Subst3 (Nb. of nodes), M4-Post-Fault (isolated or not); **electrical status** : Q-M4, V-M4, Line141-P and Line142-Q; **“ad hoc” combinations** : Accelerating Power (AP), Total Prefault Power of Cleared Lines (TPPCL), Accelerating Power divided by the number of remaining Lines in the Postfault configuration (AP/LP).

Classification w.r.t. the “actual clearing time”. The obtained trees² are very simple (15 to 30 nodes, less than 10 test attributes) and quite reliable ($P_e \approx 1$ to 2%). Such typical trees are portrayed in Figs. 13.7 and 13.8. (The LH subtree attached to node D1 of Fig. 13.8 is discussed in §13.3.4.) Relating to the tree of Fig. 13.7, the typical error-bar diagram in the lower part of the figure provides more refined information about the classification errors, in terms of their CCT : out of the 62 states misclassified by the tree³, a very large majority are clearly concentrated in the $\pm 10\%$ range of the thresholds of 155ms and 350ms defining the stability classes.

As concerning the tree of Fig. 13.8, it is interesting to note that most of its test attributes (5 out of 7) are topological ones; a further analysis (not given here) shows that they carry about 67% of the “information quantity” (i.e. classification capability) of the tree.

Classification w.r.t. the median. The corresponding trees are generally much more complex (e.g. about 50 nodes, up to 20 test attributes) and present significantly higher error rates than the previous ones ($P_e \approx 7$ to 11%). However, considering their CCT distribution, one again observes that the errors essentially concentrate in the vicinity of the class boundary. As an illustration Fig. 13.9 provides the CCT distribution of the classification errors of DT26 of Table 13.3 ($P_e = 7.1\%$, 51 nodes, 14 different test attributes) : 90% of the errors concentrate in the $235ms \pm 10\%$ range and only 3% of the errors (5 cases out of 3000) fall below 210ms.

Four-class trees. The error rate and the complexity are even more important. However, here most of the errors are located in adjacent classes and correspond to a less misleading diagnostic than for two-class trees. This is illustrated in Table 13.2, where the diagonal entries correspond to correctly classified states, the entries below the diagonal to overly optimistic diagnostics, those above the diagonal to overly pessimistic ones.

13.3.3 Effect of attributes

The “optimal” parameters determined in the above investigation ($\alpha = 10^{-4}$, $N = 2000$) were used to assess the different types of candidate attributes described below. Trees

²For the actual clearing time as well as for other thresholds located in a valley of the CCT distribution.

³All the available 3000 states are used for the purpose of this error analysis, in order to ensure that no “large” errors are missed, should they be learning states. The 62 errors are composed of 25 out of the $M = 1000$ test states and 37 out of the $N = 2000$ learning states.

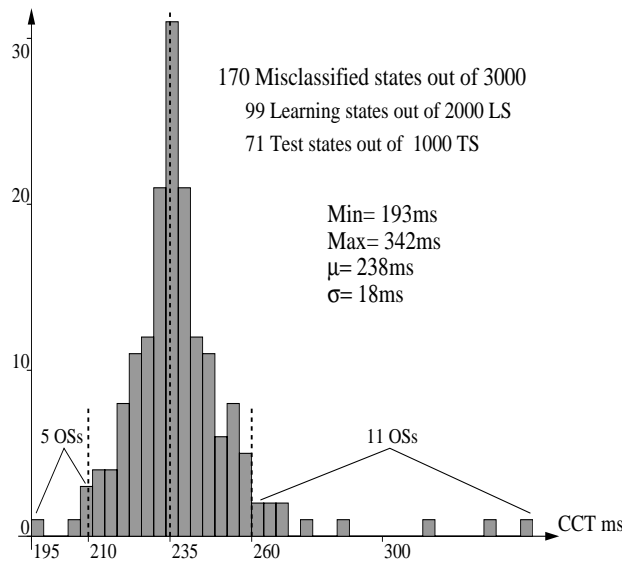


Figure 13.9 CCT distribution of errors of DT26. Adapted from [WE 93d]

Table 13.2 Distribution of errors of a 4-class tree. Adapted from [WE 93d]

		True class (thresholds in ms)				Total
		Nb. of test states (Nb. of all states)				
		<155	155-200	200-250	>250	
Class via DT (ms)	<155	59 (197)	12 (25)	3 (3)	1 (2)	75 (227)
	155-200	11 (25)	139 (496)	36 (71)	0 (2)	186 (594)
	200-250	0 (0)	24 (53)	286 (867)	41 (88)	351 (1008)
	>250	0 (0)	1 (1)	30 (70)	357 (1100)	388 (1171)
Total		70 (222)	176 (575)	355 (1011)	399 (1192)	1000 (3000)

were thus built for the busbar fault, corresponding to various lists of candidate attributes and to the two 2-class classifications obtained with the threshold values of 155ms and 235ms.

Different types of candidate attributes

13 lists of candidate attributes have been used during our application study.

1a. Controllable attributes. This minimal list contains the 38 following variables :
 (i) active generation of each unit of each plant of the region; (ii) their EHV voltage;
 (iii) global regional load; (iv) logical attributes describing the topology. Ideally, the

DTs should rely on this kind of attribute only.

- 2. Observable attributes.** The following 9 lists are composed of pre-fault attributes of growing complexity, i.e. of decreasing controllability. Lists 2a-2e are elementary parameters of the operating state, easily available in a control center. Lists 2f-2i refer to composite attributes, combining information about several power system components, which however are restricted to be fault independent.

2a = 1a + Reactive generation of each unit of the region.

2b = 2a + Power flows on important lines

2c (resp. 2d.) = 2b (resp. 1a.) + Phases of the main substations of the study region, relative to the external load area.

2e = 2b + Linear combinations of P, Q and V allow us to take into account with a single tree test, the combined effect of two different characteristics via a linear combination.

2f = 2b + Topology combinations

2g = 2f + Power combinations

2h = 2g + Short-circuit admittances/powers quantify the combined effect of line outages as well as substation and plant configuration on the strength of the topology.

2i = 2h + Linear combinations of P, Q and V

- 3. Complex attributes** obtained by including in list 2g attributes of arbitrary complexity possibly taking into account the during and / or post-fault configuration. They generally yield simpler and more reliable trees, but require more complex computations and a certain expertise to use them.

3a = 2b + “Ad hoc” combinations suggested by prior experience and physical interpretations. They take into account the effect of topology and electrical status on the accelerating power during the fault on period and of the number of available lines to exchange the stored energy during the post-fault swings.

3b = 3a + Post-fault information provided in the form of equivalent post-fault equilibrium parameters (Thévenin e.m.f., power angle, maximal electric power . . .) of an empirical “one machine infinite bus” representation used by planning engineers, as a rule of thumb for first shot stability assessment.

3c = 3a + Linear combinations of P, Q and V.⁴

⁴Notice in Table 13.3 the good performances of DT13 and DT26 obtained by using attributes of this type.

Table 13.3 *Effect of the types of candidate attributes. Adapted from [WE 93d]*
 $N = 2000$ $M = 1000$ $\alpha = 10^{-4}$ $H_m = 10^{-2}$

ATTRI-BUTES		$\tau = 155\text{ms}$ 222 Unst & 2778 St					$\tau = 235\text{ms}$ 1493 Unst & 1507 St				
List	Nb.	DT	Quality		Compl.		DT	Quality		Compl.	
			$P_e\%$	$I_Q^{DT}\%$	$\#\mathcal{N}$	$\#\text{A}$		$P_e\%$	$I_Q^{DT}\%$	$\#\mathcal{N}$	$\#\text{A}$
1a	38	DT1	1.8	63.7	25	7	DT14	11.3	70.7	49	14
2a	48	DT2	1.5	74.5	29	9	DT15	11.5	71.4	51	13
2b	60	DT3	1.7	76.6	23	10	DT16	9.9	71.9	53	15
2c	72	DT4	cf. DT3		cf. DT3		DT17	10.1	71.4	55	18
2d	50	DT5	2.3	58.7	21	9	DT18	10.6	70.9	45	13
2e	63	DT6	2.0	80.2	23	9	DT19	10.0	75.3	55	14
2f	72	DT7	1.6	76.1	21	10	DT20	cf. DT16		cf. DT16	
2g	92	DT8	cf. DT7		cf. DT7		DT21	11.1	73.5	61	18
2h	120	DT9	1.7	78.7	23	11	DT22	11.3	75.1	67	20
2i	123	DT10	1.4	79.9	21	8	DT23	9.7	76.6	55	18
3a	100	DT11	1.3	78.2	19	9	DT24	8.2	79.2	43	14
3b	119	DT12	1.9	85.0	9	4	DT25	7.3	88.2	41	15
3c	103	DT13	1.0	78.6	15	7	DT26	7.1	81.5	51	14

Tree characteristics

Some interesting characteristics of the resulting trees are summarized in Table 13.3, for growing attribute complexity : list 1a corresponds to purely controllable attributes, lists 2a-2i to observable ones, and lists 3a-3c to complex ones. The first two columns of the table identify the name of the list and the number of its candidate attributes. The next five columns relate to trees built with the “actual clearing time” threshold (155ms), whereas the following five use the “median” threshold (235ms). For each one of these two blocks, the following columns are listed :

- “DT” : the tree name
- $P_e\%$: the test set error rate
- $I_Q^{DT}\%$: the information quantity provided by the tree, evaluated as a percentage of the learning set entropy; it reflects the degree of tree classification capability in a global way [WE 91a]
- $\#\mathcal{N}$: the tree complexity in terms of its node number
- $\#\text{A}$: the number of test attributes selected by the tree.

A comprehensive discussion about the rich, multiform information provided by a tree would necessitate much space. We will restrict ourselves to observe again that globally, the trees can indeed provide a clear picture of the intricate transient stability phenomena. At the same time, they assess the stability behavior of an operating state in terms of

solely the test attributes relevant to this state. Further, the influence of each test attribute may be appraised by means of its relative position in the tree and by its information quantity or classification capability.

These and many other pieces of information may be very useful to the system operators; they corroborate and/or complement their own experience obtained via tedious everyday learning of the system behavior, and help them get a refined and confident understanding of the phenomena.

The sheer classification ability of the tree through its hierarchical structure is another fundamental property worth mentioning again; it is nicely highlighted by comparing DT1 of Fig. 13.8 with the extreme intertwining of states' stability degree suggested by Fig. 13.6d, drawn for two quite important numerical test attributes of the tree : P-M4, the active power generated by unit 4 which appears at three different test nodes and contains about 19% of the tree's information quantity, and V-M4, which, although used only once, contains 14% of the information quantity of the tree.

Coming to more specific information stemming from the results of this section, we observe the following.

Stop-splitting rule. The effect of parameter α on the tree complexity and reliability, observed in previous studies, is fully corroborated : using low α values allows one to reduce the tree complexity by a factor of 2 to 3, while improving (albeit slightly) reliability.

DTs built w.r.t. the “actual clearing time”. Table 13.3 shows that even the most elementary attributes (list 1a) yield DTs of satisfactory reliability (this is confirmed by a more refined analysis of the classification errors). A very good compromise thus seems to be DT1, if sensitivity analysis and preventive control applications are sought. On the other hand, DT13 seems to be a good choice if only analysis is considered : it is more reliable and the used attributes, although interdependent and fault specific, remain quite easy to appraise in a control center environment. Thus the combined use of both DT1 and DT13 would allow us to achieve both reliability and flexibility of use.

DTs built w.r.t. the median. For DT14 to DT26, the effect of candidate attributes on tree parameters is more strongly marked : complexity and reliability vary in an important fashion. Likewise DT1 and DT13, DT14 and DT26 appear to be a good choice.

Linear combinations. The automatic linear attribute combination allows one in general to somewhat improve the tree reliability (lists 1f, 2d and 3c). Their slightly better performances are however obtained at the expense of less straightforward tree interpretability capabilities. A further use of such attributes will be illustrated in §13.4.

13.3.4 Quality improvement

In its information theoretic formulation, the decision tree induction algorithm does not distinguish between the different natures of information. In particular, it provides a tradeoff among the detection of states of different classes which does not take into account the pragmatic non-detection costs. Basically, the method aims at predicting class probabilities as precisely as possible. However, in security assessment practice one is more interested in a highly sensitive detection of unstable states than of stable states. Hence the necessity to define pragmatic quality measures and to bias the tree induction method in order improve the latter, if required.

Pragmatic quality measures

The detailed assessment of the pragmatic quality of decision trees led us to distinguish between the following types of errors :-

False alarms. Stable states classified as unstable.

Non-detections. Unstable states classified as stable.

Dangerous errors. Fairly unstable states classified as unstable. A state is fairly unstable if its CCT is smaller than 0.9τ , where τ is the threshold used to classify states, normally equal to the actual clearing time.

Normal errors. Unstable but not fairly unstable states classified as stable.

In practice, one is more interested in reducing the number of non-detections, and among these, more particularly the dangerous errors than the normal errors.

In regard to these error types, the trees obtained so far via the “pure” DTTS (Decision Tree based Transient Stability) method achieve very low error rates, with very few dangerous errors. Yet, for real life uses, it is desirable to further reduce as much as possible the dangerous diagnostics, without generating, however, too many “false alarms”. To achieve this goal, the three following “pragmatic quality measures” have been used to account for different types of errors of a tree : P_{FA} the proportion of false alarms; P_{ND} the proportion of non-detections; P_{DE} the proportion of dangerous errors.

Reducing the number of dangerous errors

Among several techniques investigated, we mention the most efficient ones : (i) using a CCT threshold slightly (5-10%) larger than the desired one, so as to increase the number of states classified unstable; (ii) biasing the probability of unstable states, by increasing their weight; (iii) using high relative non-detection **costs** for the unstable states when determining the class labels of terminal nodes according to the rule described in Table 3.8. This amounts to labelling a deadend as stable, only if a large enough majority of

Table 13.4 *Quality improvement of DT3 of Table 13.3*

Technique for improving Q	P_{FA}		P_{ND}		P_{DE}	
	Nb.	(%)	Nb.	(%)	Nb.	(%)
Basic	9	(0.30)	48	(1.60)	12	(0.40)
(ii)	23	(0.77)	32	(1.06)	9	(0.3)
(ii) + (iii)	47	(1.57)	17	(0.57)	2	(0.07)

its states are stable. In other words, the “small” deadend nodes, located on the stability boundary, are preferably labelled unstable.

The simulations show that the combined use of either technique (i) or (ii) at the tree building stage and technique (iii) at the tree application stage, yields very satisfactory results. This is illustrated in Table 13.4, which lists the different types of errors of DT3 of Table 13.3, and its improved versions. (The percentages are given with respect to the 3000 states used to evaluate the tree qualities.)

Reducing the number of false alarms

As one may see, the previous techniques allow us to efficiently reduce the number of dangerous diagnostics of a tree, but at the price of an increased number of false alarms. We therefore propose to use an incremental tree development scheme, in order to compensate for the latter increase.

This is illustrated in Fig. 13.8, where 500 additional learning states have been used to build an incremental subtree for node D1. This node corresponds to the following constraints : “Line132=out” (introduced at the top-node); “Subst1<2” (introduced at node T2); and “Line141=out” (introduced at node T3).

Further investigations were carried out on this subregion of the attribute space, for which a data base composed of 2000 states was generated. This rather specific range of operating conditions corresponds more closely to a characteristic range of situations which would be studied in operation planning. Thus, the resulting trees may reflect more closely the type of criteria which could be used in practice, rather than the previous very “general” trees. In the multicontingency study described below, we have generated a data base for a similar range of conditions.

In the above subregion of D1 of the operating space the percentage of unstable states is equal to 44%, which is much higher than overall. Thus, without incremental tree growing the local error rate at the terminal node D1 would be 44% of non-detections or, if we use high non-detection costs of unstable states, 56% of false alarms. To improve the tree, a subtree is grown on the basis of 1500 states. It is slightly more complex than our “global” tree, reflecting the fact that more refined information is required to distinguish among unstable and stable states in the corresponding region.

Table 13.5 *Quality improvement of subtree of DT1 of Fig. 13.8*

Technique for improving Q	P_{FA}		P_{ND}		P_{DE}	
	Nb.	(%)	Nb.	(%)	Nb.	(%)
Basic	64	(3.2)	44	(2.2)	17	(0.85)
(ii) + (iii)	195	(9.75)	9	(0.5)	2	(0.1)

This tree presents error rates between 5 and 10 %, depending on the type of “false alarm vs non-detection” compromises sought. The quality measures corresponding to an unbiased and a very conservative compromise are given in Table 13.5. Thus, if we replace the deadend node by the biased subtree, we are able to reduce the false alarm rate from 56% to 9.75%, at the price of a negligible number of non-detections.

The above suggested iterative tree enhancement requires further investigations to become truly effective, e.g. to quantitatively evaluate the amount of additional learning states required and the proportion of tree nodes that need to be expanded. It appears however to be very promising, as shown by preliminary investigations. In particular, a large majority of the false alarms of a tree are generally located at a small number (3 or 4) of its “weak” deadends. Thus only a small part of a tree would need an iterative enhancement, and consequently only a reasonably small number of additional learning states would be required.

Discussion

The above presentation shows the possibility of controlling the “false alarm vs non-detection” compromises of decision trees. One of the main tools used to analyze and improve the trees is the precomputed stability margin, here in the form of the CCT. While the computation of these margins is costly in terms of computing times, this is largely paid back by the increased flexibility of security assessment. Other approaches to exploit the margin information are discussed in §13.3.6.

On the other hand, if no margin information is available, we need to develop alternative approaches allowing us to shift the thresholds in the decision trees so as to control their “false alarm vs non-detection” compromise. Other possibilities have already been mentioned in chapter 6 concerning the combination of decision trees and distance computations in the attribute space, in order to obtain complementary information from the nearest neighbors in the data base. This needs further research however.

13.3.5 Multicontingency study

For multicontingency security assessment the following is a sample of questions which may be raised.

What are the global stability limits of an operating condition within which it is simultaneously stable with respect to all contingencies ?

Which are the contingencies for which a state is unstable ?

What is the overall ranking of the contingencies in terms of their severity, for a range of operating conditions ?

The first two questions may be easily tackled via single-contingency trees. However, multicontingency trees may also be considered in order to take explicitly into account the similarities and differences among contingencies.

Within this context we may distinguish *global* and *contingency dependent* multicontingency trees.

The former kind of tree was illustrated in §10.1.3 and in Fig. 3.16. They classify an operating state as unstable as soon as it is unstable with respect to at least one contingency, without however indicating which one. Their main advantage is interpretability : they are able to provide the type of information which is necessary to an operator in order to quickly appraise the security of the system and identify potential problems and possible control actions.

The other type of multicontingency decision trees are essentially aiming at compressing the single-contingency trees without loss of information about the identification of the dangerous contingencies.

As concerns the third question, it may be answered by various statistical analyses of the data base and in particular the so-called contingency ranking trees discussed below.

The multicontingency study results are detailed in ref. [AK 93] ; below we merely give some examples and the main conclusions.

Simulated contingencies

We selected contingencies which are possibly constraining for the study plant. Exploiting symmetry to exclude redundant contingencies, a total of 17 faults have been defined.

12 Line faults comprising :

3 Single Line Faults (SLF) : three-phase short-circuits ($3\phi SCs$) on one of the outgoing lines which is cleared by opening the line. The fault clearing time is 90 ms. i.e. $\tau = 90\text{ms}$.

3 Single Line Delayed reclosure Faults (SLDF) : a delayed reclosure (after 20 secs) of circuit breakers for SLFs is considered, assuming a permanent fault. The system equilibrium reached when the circuit breakers reclose is computed by running a load

flow. A $3\phi SC_s$ is simulated starting with this initial condition. ($\tau = 110\text{ms}$.)

3 Double Line Faults (DLF) : two simultaneous $3\phi SC_s$ on the double circuit lines towards each of the three neighboring substations are considered. ($\tau = 90\text{ms}$.)

3 Double Line Delayed reclosure Faults (DLDF) : these are the DLF with delayed reclosure of breakers. $\tau = 110\text{ms}$.

5 Busbar Faults ($\tau = 155\text{ms}$) comprising :

2 Single Busbar Faults (SBF) : $3\phi SC_s$ on the busbar sections cleared by isolating the busbar section, tripping the machine and opening the lines on the section. (Faults numbered 13 and 14.)

1 Double Busbar Faults (DBF) : when a busbar section is out of operation, the machines and lines on it are transferred to the opposite busbar section. A $3\phi SC_s$ is assumed on this section. (Fault numbered 15.)

2 Central Busbar Faults (CBF) : when a busbar section is out of operation, if a $3\phi SC_s$ were to occur on the central busbar section, up to two lines and one machine would be lost and breakers would be opened, resulting in 2 nodes at the substation. (Faults numbered 16 and 17.)

Global decision trees

The upper part of Table 13.6 gives a comparison of the test set error rates and complexities of various strategies used for simultaneous stability assessment with respect to all 17 faults. For the single-contingency DTs, the complexity is the sum of the number of nodes of all the DTs. To allow direct comparisons, the DTs built for the two strategies should be based on the same set of candidate attributes. The set1 of candidate attributes is a very rich set composed of 241 attributes including fault independent and fault dependent ones; set2 attributes correspond to a simpler set of fault independent specific and global attributes of list 2g; set3 corresponds to elementary observable attributes of list 2b.

The characteristics of the tree obtained via the two strategies and for the two last lists of fault independent candidate attributes are shown in rows 2 and 4, and 3 and 5 of Table 13.6. Observe that the increase in error rate P_e of the global trees vs the corresponding single-contingency trees (e.g. 13.0% vs 10.5%) is accompanied by a dramatic decrease in complexity (e.g. 47 vs 315 nodes).

The second part of Table 13.6 shows the characteristics of the global trees obtained for a more homogeneous group of 14 contingencies, where the double and central busbar faults have been excluded. The tree corresponding to the Set2 attributes (similarly to the DT in Fig. 13.10) was described earlier in Fig. 10.3 in §10.1.3.

Table 13.6 DTs for collective stability assessment

All 17 contingencies			
No	Type of DT	P_e %	# \mathcal{N}
1	17 Single-contingency DTs (Set1 attributes)	6.6 %	201
2	17 Single-contingency DTs (Set2 attributes)	10.5 %	315
3	17 Single-contingency DTs (Set3 attributes)	14.7 %	214
4	1 Global DT (Set2 attributes) (see Fig. 13.10)	13.0 %	47
5	1 Global DT (Set3 attributes)	16.6 %	25
14 similar contingency			
No	Type of DT	P_e %	# \mathcal{N}
6	14 Single-contingency DTs (Set1 attributes)	4.4 %	204
7	14 Single-contingency DTs (Set2 attributes)	9.5 %	264
8	1 Global DT (Set1 attributes)	5.7 %	53
9	1 Global DT (Set2 attributes) (see Fig. 10.3)	7.4 %	41

The type of information provided by a global DT is illustrated at Fig. 13.10. It is interesting to notice that its test attributes are referring to general, fault-independent parameters of an operating state. For example, the test selected at the top node shows the influence on the stability of the number of lines in operation in the prefault phase. Other test attributes account for the total active prefault power generated or flowing through different groups of lines.

Coming back to the respective advantages of global vs single-contingency DTs, we first note that the latter often allow us to take better advantage of contingency-specific attributes; they are able to provide richer stability information and to identify potentially dangerous contingencies. On the other hand, global trees characterize in a very simple and compact manner the structural stability limits of a subsystem of the overall power system. However, their quality depends strongly on the set of contingencies which are grouped and also on the type of attributes used. In terms of practical uses, the global trees are more likely to provide a control tool for the operator, whereas the single-contingency trees are able to express more refined information which may be usefully exploited by the engineers in the context of off-line studies and as an analysis tool for on-line operation.

Contingency dependent decision trees

With respect to global DTs, contingency-dependent multicontingency DTs aim at telling also which contingencies are unstable under particular conditions. They therefore classify *stability cases* which belong to the Cartesian product set of the prefault operating states (OSs) and of the contingency set. Starting with N operating states and

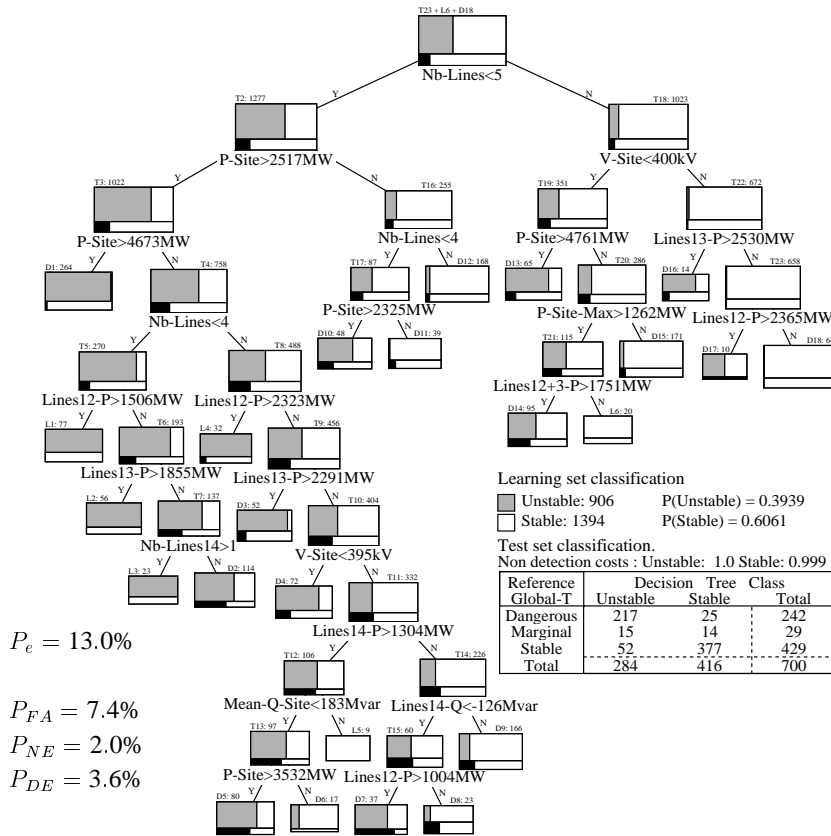


Figure 13.10 Global decision tree for all 17 faults

C contingencies this yields possibly $N \times C$ stability cases. These are generally characterized by three types of candidate attributes : (i) contingency independent attributes characterizing the OS; (ii) OS independent attributes characterizing the contingency; (iii) combined attributes taking into account both the OS and the contingency (e.g. such as post-fault topology ...).

One of the potential advantages of these trees is their ability to uncover and exploit similarities among contingencies. The partial tree shown in Fig. 13.11 illustrates this possibility. This tree was built for the three faults defined in §13.3.1 : (i) the “busbar” fault (denoted “BF”, in the tree), cleared after 155ms; (ii) the “double line” fault (“DLF”); (iii) the “single line” fault (“SLF”), both cleared after 100ms. The three contingencies together with the 3,000 operating states yield 9,000 stability cases : a random sample of 6,000 are used as the learning set, and the remaining 3,000 as the test set. To save space, LH and RH parts of the tree have not been represented in the figure. Note that the nodes where the retained test attribute is “Fault” are encircled by

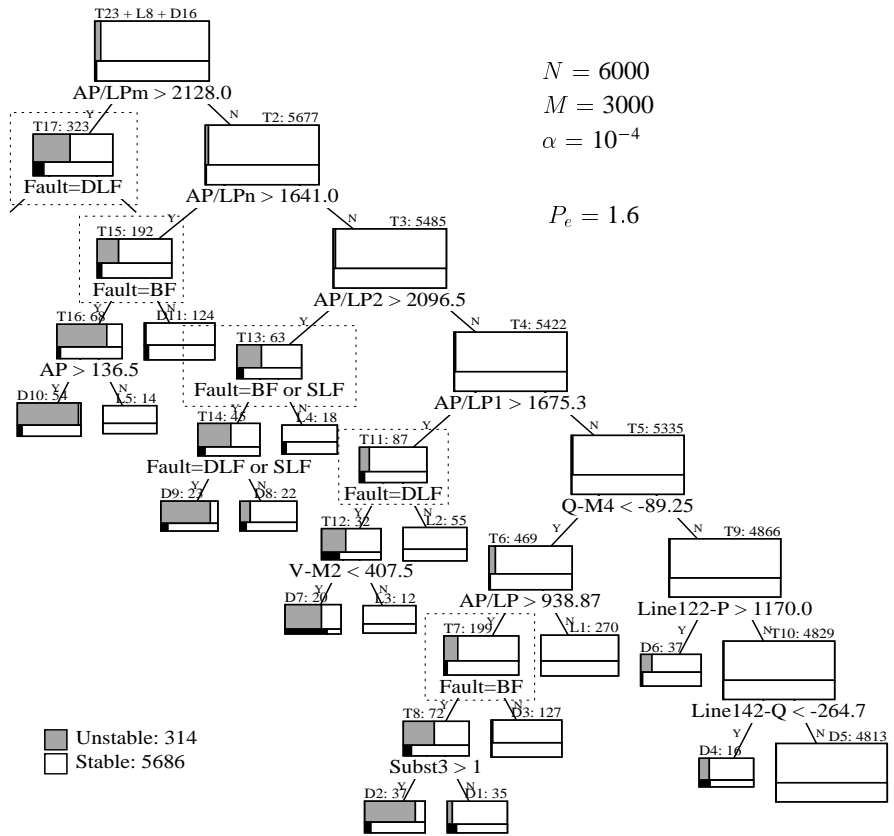


Figure 13.11 Partial view of a contingency dependent tree. Adapted from [WE 93d]

dotted line boxes.

Comparing this tree with the corresponding single-contingency ones, we observe that it has (i) a complexity of 47 nodes vs 45, the total number of nodes of the three single contingency trees; (ii) an error rate of 1.6% vs 1.7%, the mean error rate of the single contingency trees; (iii) 14 different test attributes (including the attribute “Fault”) vs 18, the total number of different test attributes of the single contingency trees.

Thus, without loss of reliability, the multicontingency tree provides a more synthetic view of the stability relationship than several single contingency trees. Moreover, similarities among contingencies are identified and highlighted by the tree (e.g. the operating states corresponding to node D10 are unstable with respect to fault BF; states corresponding to node D11 are stable for the SLF and DLF faults etc.).

Further, inspection of Fig. 13.11 suggests that, although equivalent to the information

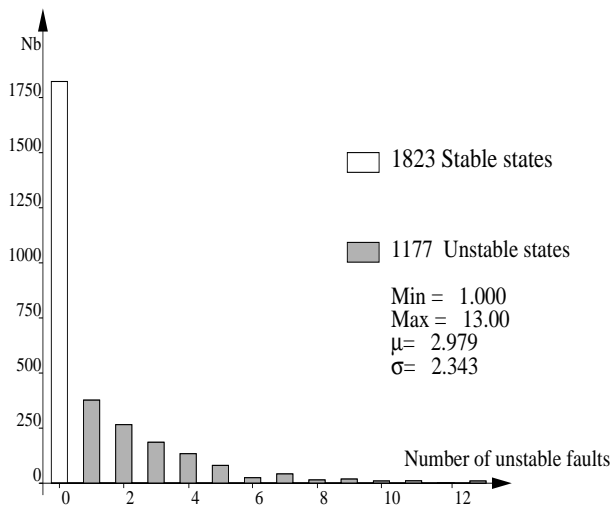


Figure 13.12 Frequency diagram of the number of simultaneously unstable faults

provided by a set of single-contingency trees, the information provided by the corresponding multicontingency tree is presented in a more compact and easier to exploit fashion. This can be explained by the fact that similarities of different contingencies are exploited during the tree building so as to simplify the resulting tree. In particular, overlappings of unstable (resp. stable) regions are identified and embedded in the tree : hence, combinatorial explosion, inherent in multicontingency control on the basis of single contingency trees, is avoided as much as possible.

Overall, though it is still too early to assess advantages of a multicontingency tree, we observe that it directly provides any of the following types of information :

- for a given fault (among those used to build the tree) is the considered operating state likely to be unstable or not ?
- for a given operating state, are there faults likely to create instability ?
- which conditions characterize the prefault attributes of stable operating states for a given set of possible faults ?

Conceptually, the trees introduced here are similar to the emergency state detection trees introduced in §10.2.3. They both classify stability cases; however, their purpose is quite different : in the above trees we aim essentially at analyzing contingency similarities while in the context of emergency state detection we aim at building a robust and to possible extent a contingency independent tree, classifying stability cases in terms of attributes determined in the just after disturbance (JAD) states.

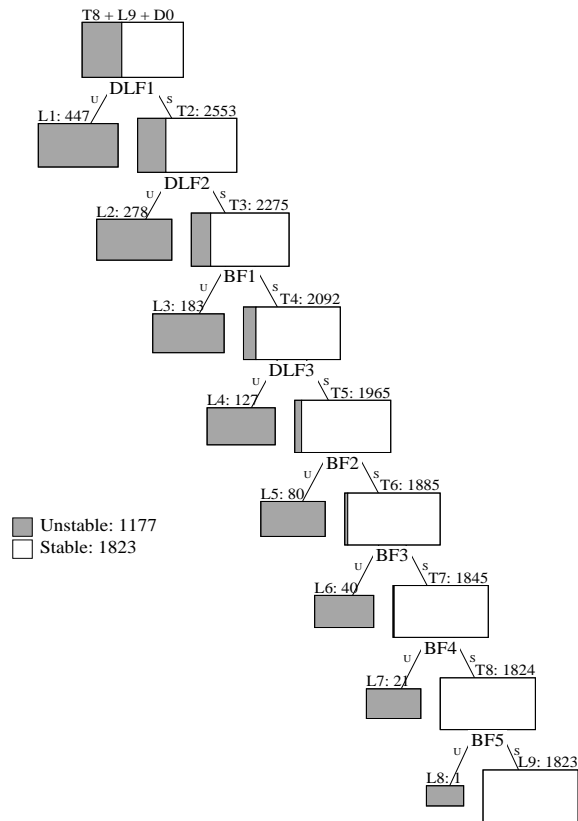


Figure 13.13 Contingency ranking via a global DT. Adapted from [PA 93]

Contingency ranking

The frequency diagram in Fig. 13.12 describes the overall frequency of OSs, which are simultaneously unstable for 0, 1, 2, . . . 17 faults. It shows that it is rather unlikely to observe OSs simultaneously unstable for more than 7 faults.

Another kind of analysis is illustrated by the *contingency ranking tree* shown in Fig. 13.13. It is constructed on the basis of the complete data base classified globally with respect to the 17 contingencies, a state being classified unstable if it is unstable for at least one contingency. On the other hand, the attributes used to build the tree are the 17 elementary single-contingency classifications, denoting a state as unstable if it is unstable with respect to the corresponding contingency. The fact that only 8 out of the 17 contingencies have been necessary to recover completely the global classification indicates that there is some redundancy among the different contingencies.

Table 13.7 Contingency ranking

Node	DLF1	DLF2	BF1	DLF3	BF2	BF3	BF4	BF5	DLF4	DLF5	DLF6	SLF1	SLF2	SLF3	SLF4	SLF5	SLF6
L1	447	164	103	87	84	85	111	63	215	106	51	46	29	24	77	31	31
L2	0	278	33	120	56	90	53	2	0	120	86	0	13	25	0	26	33
L3	0	0	183	24	6	58	113	0	0	0	12	0	0	0	0	0	2
L4	0	0	0	127	32	24	6	6	0	0	66	0	0	0	0	0	0
L5	0	0	0	0	80	1	9	2	0	0	0	0	0	0	0	0	0
L6	0	0	0	0	0	40	3	1	0	0	0	0	0	0	0	0	0
L7	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0
L8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
L9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tot	447	442	319	358	258	298	316	75	215	226	215	46	42	49	77	57	66

Each one of the 8 left-most leaves of the tree corresponds to a subset of unstable states; e.g. L1 corresponds to the 447 set of states which are unstable w.r.t. DLF1; L2 corresponds to the 278 states which are unstable w.r.t. DLF2 and which are stable w.r.t. DLF1 The right-most leaf L9 on the other hand corresponds to the subset of stable states. Its interpretation is that if an OS is stable with respect to the eight contingencies in the tree, then it is also stable with respect to the other 9 contingencies. The tree identifies a minimal set of most constraining contingencies. Further, it provides a ranking of the latter, the top-most contingencies being the most severe ones.

Table 13.7 provides a more detailed description of the states corresponding to each of the 9 leaves of the tree, in terms of the number of states which are unstable with respect to any of the 17 contingencies. Since the leaves correspond to a non-overlapping partition of the complete data base for each contingency, they partition its unstable states. Thus the total number of states within each column corresponds to the total number of unstable states of the corresponding contingency. For example, considering the columns DLF4, SLF1 and SLF4 we observe that the unstable states with respect to any of these contingencies are also unstable with respect to DLF1. Similarly, the unstable states w.r.t. DLF5, SLF2, SLF3 and SLF5 are covered by DLF1 or DLF2.

13.3.6 Other learning approaches

In this section we describe some investigations made in the context of the EDF system with other methods than the basic decision tree induction described in detail above. These are interesting from several respects, since they offer for the first time a comparative assessment of different computer based methods in the context of a real life power system and are based on a representative data base.

We will first describe briefly the results obtained with the sporadic investigations concerning the hybrid DT-ANN and DT-NN methods presented in chapter 6. Then we reproduce the results obtained in the Statlog project with the data base that we have provided. This unique comparative study offers a very broad and systematic assessment

of state-of-the-art methods in computer based learning, with respect to a typical and realistic preventive transient stability assessment problem.

Hybrid

In §10.1.3 we have illustrated the use of the hybrid DT-ANN approach to improve the accuracy of a single-contingency tree for a double-line fault.

In ref. [WE93a] a more systematic exploration is reported, concerning a simplified problem corresponding to the constant topology data base described above. In this study, various methods have been compared, in particular a bare DT, a nearest neighbor classifier used to interpolate CCTs in the space corresponding to the test attributes selected by the tree, and hybrid as well as standard multilayer perceptrons. The results of this study are summarized below.

The data base of 3000 OSs was divided into a LS composed of 2000 states, used for DT building and MLP learning, and a TS composed of the remaining 1000 states, exclusively used to estimate error rates and accuracies of CCT approximations. A DT was built for a lateral fault, and was translated into an MLP composed of 4 input, 15 test, 16 anding and 2 output (classification) neurons, containing 138 parameters. The MLP's output was also exploited to avoid errors nearby the stability boundary : rejecting states for which the activations of the two output neurons were not sufficiently different.

The two output neurons were further merged, in order to obtain a “margin regression MLP”. Several ways were considered to normalize the CCTs among which we report the “full” hybrid approach, where the full range of CCTs is used and the “truncated” hybrid approach, where only a small subrange around the classification threshold is used. In the simulations using neural networks, the parameters are adapted on the basis of the CCTs of the 2000 learning states.

Results are summarized in Table 13.8 showing the main features of the different methods. Accuracy is evaluated on the 1000 test states, and characterized in three ways : (i) the global fit is reflected by the correlation coefficient ρ (a value of 1 would indicate a perfect fit, hardly reachable on test states); (ii) near threshold τ the fit is evaluated by P_e , the percentage of erroneous classifications when using the method to classify test states w.r.t. the initial threshold of 0.240s ; (iii) classification errors are described by the lower and upper bound of their CCTs, and their Mean Absolute Deviation (MAD) w.r.t. τ . This number may be compared to the precision of the CCTs computed by the SBS method, which is here of ± 5 ms.

It is seen that the hybrid approaches significantly improve the accuracy of the classification, the error rates being reduced by a factor of 2 w.r.t. the DT. At the same time, the CCTs of the classification errors fall within the SBS tolerance around the

Table 13.8 *CCT approximation via MLPs*

CCT Max	Nb. of inputs	Nb. of	Accuracy on TS (#TS = 1000)		
	Network structure	Iter.	ρ	P_e %	CCTs of Errors INF-SUP MAD
Full Hybrid MLP (BFGS)					
1.000	4 (as DT)	50	0.915	1.7	230–248 (4)
	4-15-16-1	200	0.918	1.9	220–248 (5)
Truncated Hybrid MLP (BFGS)					
0.350	4 (as DT)	50	0.979	1.5	230–258 (5)
	4-15-16-1	200	0.982	1.4	220–239 (5)
Direct MLP (Conjugate Gradient Polak-Ribiere)					
1.000	17 cand.	50	0.939	5.5	220–286 (12)
	17-25-1	200	0.973	3.8	220–277 (9)
		900	0.986	1.6	230–258 (8)
10-Nearest Neighbor interpolation					
1.000	4 (as DT)	–	0.903	2.4	210–258 (7)
Decision Tree					
1.000	4 (as DT)	–	–	3.3	202–267 (11)

threshold τ . The results also show that the “truncated” version outperforms the “full” version. In particular, the low ρ obtained for the latter approach indicates that a precise approximation of the CCT, in its full range is not possible with the DT test attributes. It is therefore preferable to use truncated CCTs, to avoid overfitting problems in regions where the attributes lack information.

Often, overfitting results from too many learning iterations; this is illustrated by the (albeit small) degradation of accuracy for the full hybrid network after 200 iterations. What is not apparent from statistical figures, as given in Table 13.8, is that the overfitting problem may cause CCT values to oscillate dangerously in the less densely represented regions, which may lead to completely erroneous extrapolations.

In terms of the computational involvement, the “truncated” version is definitely superior to the others, since it requires only precise CCTs in the interesting range around τ . In practice, since most of the states fall outside this range, this would allow simulation times to be reduced by at least a factor of 2. The computing times corresponding to the off-line (not accounting the SBS simulation times) and on-line use of the different methods are indicated in Table 13.9.

To obtain comparable accuracy with the “direct” approach requires a very large and often prohibitive number of learning iterations. For example, after 200 iterations its error rate is still higher than for the initial DT; to reach the accuracy of the “truncated” hybrid approach more than 900 iterations are required. Taking into account the fact

Table 13.9 CPU times on a 28MIPS Sparc2 SUN workstation

Method	Off-line (seconds)	On-line (ms)
Decision trees	Grow, Prune, Test : 170	0.3/state
Truncated Hybrid	100 BFGS iters. : 8,000	4/state
Direct MLP	900 CG iters. : 89,000	10/state

that in real life problems the number of potential attributes could be much larger than in our example, the viability of the direct approach seems questionable.

Statlog

The results obtained within the Statlog project [TA 94] were obtained on the first data base constructed for the EDF system, corresponding to the results obtained above in Table 13.3 and to the candidate attributes of list No. 2b, and the classification with respect to the actual clearing time of 155ms.

This problem was chosen as the most representative one of preventive-wise transient stability assessment. It corresponds to rather elementary observable attributes which do not play in favor of the decision tree methods, which obtains the best results in terms of accuracy with more sophisticated attributes. Using a larger set of candidate attributes would also have been at the disadvantage of the statistical and neural network approaches from the computational point of view.

The results obtained are summarized in Table 13.10. The first column describes the particular algorithm used; for the sake of clarity we have grouped together the methods according to the three families of algorithms discussed in Part 1. (Among the machine learning methods the first seven are of the TDIDT family : Cart, Indcart, NewID, AC2, BayTree, C4.5, Cal5.) The three following columns indicate the amount of virtual memory and of CPU time in seconds required during the learning and testing stages for each algorithm. This gives an indication of the relative performance of different algorithms, which have mostly been determined on standard workstations (e.g. SUN SPARC2). Finally, the last two columns indicate the error rates obtained in the learning and test set. The difference between these two numbers gives an indication of the degree of overfitting of various methods.

We quote the conclusions given in ref. [TA 94] :

Smart comes out top again for this data set in terms of test error rate (although it takes far longer to run than the other algorithms considered here). Logdiscr hasn't done so well on this larger data set. The machine learning algorithms Cart, Indcart, NewID, AC2, Bayes Tree and C4.5 give consistently good results. Naive Bayes is worst and along with Kohonen and ITrule give poorer results than the default rule for the test set error rate (7.4%).

Table 13.10 Results obtained in the Statlog project. Adapted from [TA 94]

Algorithm	Maximum Storage	Time(sec)		Error Rate %		
		Train	Test	Train	Test	
Statistical methods						
Lin. Discrim	75	107.5	9.3	4.8	4.1	
Quadrat. Discrim	75	516.8	211.8	1.5	3.5	
Logist. Discrim	1087	336.0	43.6	3.1	2.8	
SMART	882	11421.3	3.1	1.0	1.3	
Kernel. dens.	185	6238.4	*	5.7	4.5	
$K - NN$	129	408.5	103.4	0.0	5.2	
NaiveBay	852	54.9	12.5	8.7	8.9	
Machine learning methods						
TDIDT	Cart	232	467.9	11.8	2.2	2.2
	Indcart	1036	349.5	335.2	0.4	1.4
	NewID	624	131.0	0.5	0.0	1.7
	AC2	3707	3864.0	92.0	0.0	1.9
	BayTree	968	83.7	11.8	0.0	1.4
	C4.5	1404	184.0	18.0	0.8	1.8
	Cal5	103	62.1	9.8	3.7	2.6
					
	Castle	80	9.5	4.3	6.2	6.4
	CN2	4708	967.0	28.0	0.0	2.5
ITrule	291	9024.1	17.9	8.0	8.1	
Neural network methods						
Kohonen SOM	585	*	*	6.1	8.4	
Dipol92	154	95.4	13.1	3.0	2.6	
MLP bprop	148	4315.0	1.0	2.1	2.2	
Rad. Basis Fun.	NA	*	*	3.7	3.5	
LVQ	194	1704.0	50.8	1.8	6.5	

We observe that the results obtained with any of the TDIDT methods are quite consistent with our own results. Indeed, the error rates range from [1.4 . . . 2.6] with a mean value of 1.86 %, whereas our own algorithm has obtained 1.7%. In terms of learning CPU times, the times range between [62 . . . 3864] seconds with a mean value of 735 seconds, which may be compared with the value of 288 seconds obtained on a SUN SPARC2 workstation with our own algorithm. On the other hand, in terms of testing CPU times our own algorithm takes about 2 seconds to complete the DT testing, which is among the fastest methods.

13.3.7 Summary

The very broad and at the same time in-depth investigation made on the transient stability of an important generation site of the EDF system took all in all 42 months and reached its conclusion some months ago. Although simplified modelling of the machines was used throughout this study, mainly for convenience, we believe that most of its conclusions would remain valid if a realistic detailed modelling of the machines were used. It is worth mentioning that a research project is currently progressing towards the integration within the very fast DEEAC method of the most important “first order” effects of speed and voltage regulators [XU 93b, XU 93d]. Hopefully, this method will allow us in the near future to build data bases more closely representing the real behavior of the power system, with similar or even reduced computing times than in our study.

At the end of this study, there are still some open practical questions. They concern in particular the best way to exploit the decision trees in planning, operational planning and operation. Certainly, the unique capability of the decision trees to identify the most influential variables and to explicitly represent the physical relationships among these and stability, make the method particularly appropriate for the determination of operating guidelines in the context of operational planning. On the other hand, the resulting tree should be exploitable as a control tool for the operator.

Since the scope of our study was from the beginning restricted to the consideration of plant mode instabilities and to the study of plant operating limits, its conclusions can hardly be extrapolated to the other more complex area mode instabilities which may appear in some parts of the EDF system. But we believe that the conclusions would certainly remain valid for similar site studies, and although the random generation procedure was very closely tailored to the specific study plant, it might be transposed quite easily to the study of other power plants or regions of the EDF system.

On the other hand, we will see in the next section that meanwhile the DTTS method has been applied to more complex situations, involving in particular a larger number of different topologies and intricate interactions of the latter with other variables.

Finally, as concerns the data base generation, which is one of the main practical problems which must be solved in applying the method, the maturity acquired on the basis of the above research contributed to the development of a new data base generation software and methodologies in the context of the research projects described in the sequel.

13.4 HYDRO-QUEBEC

This system is characterized by very long UHV transmission lines carrying large amounts of power (735 kV lines carrying over 1500 MW, on distances over 1000 km). Hence, the transmission capacity of this system is strongly related to transient stability limits. The criterion of concern here is the system's ability to withstand the loss of any single 735 kV line, following a short-circuit of 100 ms [RI90], possibly with unsuccessful reclosure. Another interesting characteristic of this system is that it is not synchronized with any neighboring utility, all interconnections being through DC links or back-to-back connections.

13.4.1 Transient stability power flow limits

The on-line strategy presently used by Hydro-Québec is an interesting system specific approach.

It consists of comparing the actual system's state with a large number of states, pre-analyzed and preclassified off-line. These latter states result from the combination of topology and load-generation-consumption scenarios and of preassigned disturbances [VI86]. It gives rise to a very large combinatorial. For example, the number of energized equipment alone (lines, static compensators, synchronous condensers, . . .) amounts to over 200 equipment statuses. The resulting combinatorial process is difficult to develop in a systematic way while identifying "interesting disturbances", leading to the loss of critical lines. The difficulty is increased by the fact that the stability assessment of such a complex system must take into account refined system modelling, and thus calls for heavy time-domain computations (fast, direct methods are here hardly acceptable because of the major role played by SVCs and DC links and because of stability criteria which require to check upper and lower bounds on voltage and frequency during 10 seconds in the post-fault state).

On the other hand, the limiting contingencies and parameters depend essentially on the topology, and the present strategy consists of decomposing the overall system into more or less independent corridors, and to study the limits on each corridor by assuming a pessimistic hypothesis for the remaining corridors. For each corridor, the topologies are then grouped within families according to the number of links (i.e. the minimal number of parallel lines in operation along the corridor). For each such study, the engineers determine, on the basis of their physical insight, a small set of parameters for which stability limits are determined, essentially independently. In addition to the highly empirical character of this methodology, heavily relying on engineering judgement, one of the weaknesses of the method is that it introduces a potentially very high degree of conservatism.

Up to now this strategy has proven satisfactory, essentially because the number of simulations which could be run within acceptable response times was rather restricted. However, the increase in system complexity and stringent operating conditions makes this strategy nearing its own limits; in particular, the very off-line generation of appropriate scenarios becomes quite laborious, in particular due to tedious *manual* selection and analysis of scenarios. For example, the determination of the stability limits of the “four link” James’ Bay corridor, considered below, amounts to about 3 man-years. In short, while the computers tend to become fast enough to run much larger numbers of simulations, the bottleneck within this approach tends to shift to the tasks which are presently done more or less manually, namely the setting up of scenarios and the analysis of results.

Another increasingly stringent difficulty consists of coping with the very rapid changes in the system behavior. For example, in recent years series compensation and HVDC links have been put in operation; both affect very strongly the stability limits. Although these new equipments would presumably allow an increase in the power flows, it would require systematic redetermination of the stability limit values to take actual advantage of this possibility, which would take several man-years. Further, in the near future additional power plants will come into operation within the James’ Bay complex, and the transmission capacity will be increased through additional lines and series compensation. In addition, in future operation it may be necessary to strengthen the security criteria, so as to enable the system to cope with three-phase faults. Finally, in recent years, mid-term voltage instabilities have started being observed and must be incorporated into the operation strategies. All these rapid changes make the off-line determination of security limits a more and more challenging task.

The presently used on-line strategy amounts to extrapolating stored diagnostics on simulated data, to assess whether the actual system state is safe enough to withstand preassigned disturbances. This on-line strategy relies on a dedicated software, LIMSEL (for “LIMit SElection”), which is basically an ad hoc data base tool to store and fetch the relevant limit values and operating strategies predetermined off-line.

The DT methodology seems to be particularly well designed as an interesting alternative or complementary approach. The stability limits determined presently are essentially contingency independent limits of simultaneous stability with respect to all potentially constraining contingencies, similar to those which would be obtained from a global DT. The machine learning framework could provide a valuable tool in order to assist the engineer by making automatically some of the presently manual tasks, while taking full advantage of existing expertise. It could consist of using precontingency operating states classified with respect to a set of contingencies rather than a single one. A precontingency state would be classified as “stable” if it is simultaneously stable with respect to each one of these contingencies in the set and unstable otherwise. The candidate attributes, on the other hand, would be chosen from important precontingency topological information (important transmission lines in or out of service) as well as

precontingency power flow levels on important transmission lines.

The resulting trees would therefore be similar to the global trees discussed in the context of the EDF system. They would allow us to assess whether a new operating state, characterized in particular by its topology and power flow levels, is indeed able to withstand all preassigned contingencies. Thus, the DTs would provide clear and accurate assessment with, moreover, known tolerance. At the same time, they could help engineers in charge of the data base generation to identify systematically the critical power flows and to augment and adjust the data base in a way suggested by the trees test attributes, thus providing richer and less conservative information.

A research project has been started in June 1992, to assess the potential use of the decision tree method within this context. In a first stage, the objective was to appraise the functional capabilities of the method, without aiming at a quantitative evaluation of the accuracy of the decision trees as compared to the present strategy. A second research stage will be required to assess the method in terms of the incumbent computing burdens and resulting accuracy characteristics which would be representative of a realistic application.

13.4.2 Study system and data base description

Within this research, a data base was generated for the Hydro-Québec system corresponding to the situation of summer 1992. The first goal was to screen systematically all relevant “four-link” configurations of the James’ Bay corridor, yielding a highly complex set of topologies. The reasons for choosing this situation were the high level of complexity, and the availability of optimized stability limits in the LIMSEL function.

Data base specification

In order to generate the data base, a specification was decided on the basis of existing expertise in order to screen all relevant situations. In particular, the following variables were chosen as parameters of the random sampling procedure.

The power flows in the three important corridors of the Hydro-Québec system are drawn independently in the intervals indicated in Fig. 13.14. The James’ Bay corridor corresponds to the study region whereas the Manic-Québec and Churchill Falls corridors are outside the study region but may influence the value of its stability limits.

The generation of the main complexes of hydro-electric power plants are adjusted so as to obtain the chosen power flows, while the distribution among the individual Lagrande and Manic/Outardes plants are randomized to yield a wide diversity among the power flows of the individual lines. Since the power flows and load

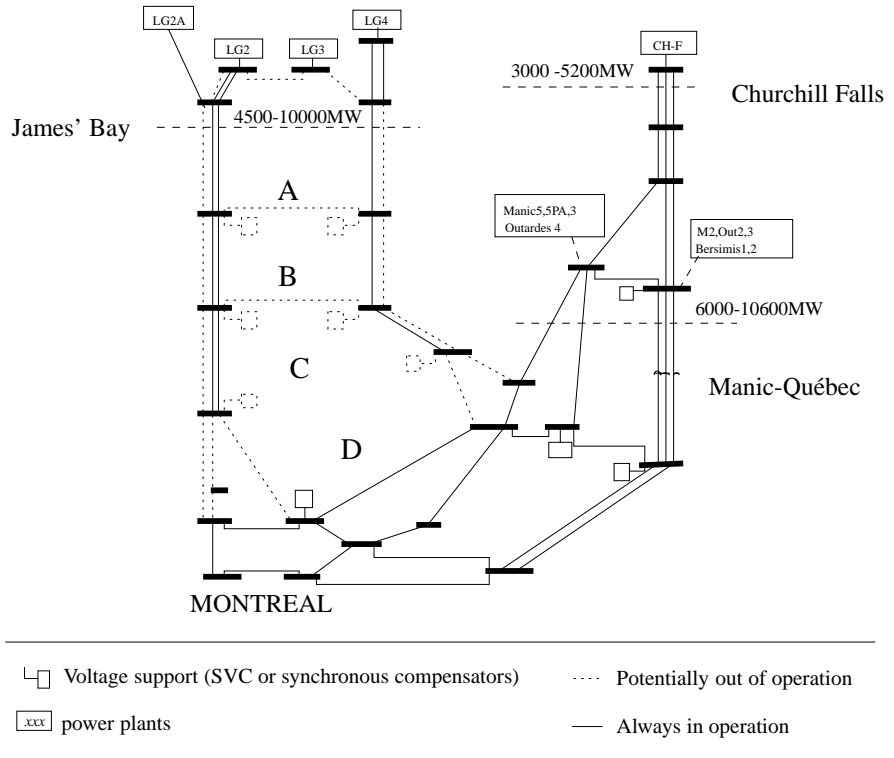


Figure 13.14 Main transmission corridors of the Hydro-Québec system

levels vary considerably, the active losses are also highly variable. In this 735 kV longitudinal system, the active losses may represent more than 1500 MW, i.e. more than 5% of the total system load. In order to avoid unrealistic generation at the slack bus, the losses are taken into account to adjust the overall load level. They are first approximated as a quadratic function of the total generation, then iteratively adjusted during the load flow computation, so as to bring the generation at the slack bus within predefined bounds [WE 93c].

The topology is chosen independently according to a pre-defined list of possible combinations of line outages with respect to the complete five-link topology. Only the James' Bay corridor is affected and only so-called four link topologies are generated. A four-link configuration is a topology where at least one of the longitudinal lines of the James' Bay corridor is out of operation, and at most one in each of the 4 sections, A, B, C, D. This yields a total of more than 300 possible topologies, grouped into 3 important classes [BE 91a].

The voltage support devices (SVCs and synchronous condensers) available in the six substations of the James' Bay corridor, indicated in Fig. 13.14, are widely variable during the random sampling since their influence on the stability limits is very strong. Their total number is drawn between 0 and 12 according to predefined probabilities, and their distribution in the substations is also randomized.

The precise specification of the random sampling scheme is described in [WE 93c]. This specification has led to the development of a program which allows us to systematically generate and analyze the data bases. Due to the complexity involved we will briefly comment on this below.

Data base generation

An important difficulty which we knew in advance we had to face with this system was related to the load flow convergence problem. Indeed, while most West-European systems are characterized by a highly meshed EHV system and many generation sites uniformly distributed with respect to the load, thus presenting a good anchoring of the EHV voltages, the Hydro-Québec system has only a few very remote generation sites and its longitudinal grid leads to very loosely controlled voltages.

The important variation of the power flows in the random sampling, induces highly variable reactive losses and hence voltage drops, which may be large enough to prevent a standard load flow computation from converging properly. Further, in order to be realistic the situations in the data base should represent normal operating conditions, which implies that the reactive compensation devices (mainly shunt reactors in the 735 kV transmission system, and shunt capacitor banks in lower voltage subtransmission systems) are adapted to the power flows and load level so as to maintain the UHV and HV voltages within tolerances. For the UHV system, this is normally done manually by the system operator who switches shunt reactors on the basis of his experience, so as to adjust the voltage to its nominal value. Up to recently, this manual approach was also used by operational planning engineers in order to set up their scenarios for the stability studies.

In order to simulate this voltage control loop, an *automatic reactive compensation* loop was developed and included into the RP600 load flow program used at Hydro-Québec. In spite of this important improvement, the first random samplings yielded a very high percentage (up to 70%) of diverging load flow computations. To be able to analyze the physical or algorithmic reasons for such high divergence ratios, various frequency diagrams were drawn for the a priori data bases, corresponding to the specifications of the randomly selected variants, classified as *diverging vs converging*.

Figure 13.15 shows a typical frequency diagram, similar to those obtained in the earlier data base generations. The proportion of converging and diverging load flow computations is represented in terms of the specified values of the power flow in

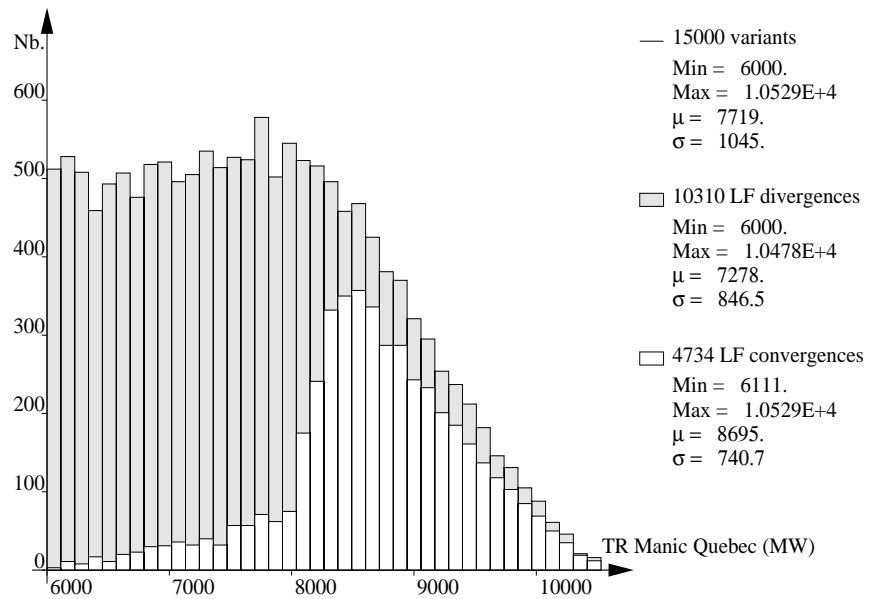


Figure 13.15 Convergence diagram of Manic-Québec power flow (6 base case files)

the Manic-Québec corridor. One can see that only a small proportion of states did actually converge, and it appears clearly from the diagram that the cases of divergence predominate mainly for power flows below 8,000MW. The reason for this is linked to the fact that the initially used base case solutions⁵ corresponded to a power flow of 10,000 MW in the Manic-Québec corridor, which prevents the load flow from converging properly when the desired power flows in this corridor are too far away from this value.

All in all, several iterations were required in order to obtain satisfactory data base generation. For example, in order to improve the convergence of the cases corresponding to a low power flow in the Manic-Québec corridor, we have used six additional base case solutions corresponding to a power flow of 7,000 MW in this corridor. This yielded a panel of 12 base case solutions corresponding to the combinations of low and high power flows in the James' Bay and Manic-Québec corridors and 3 topological variants. To each random variant the most similar base case was associated, according to its power flows and topology. This resulted in a final divergence rate of 16.7%, and a further systematic analysis showed that the corresponding cases were more or less uniformly distributed in terms of all the important parameters. As an illustration of the final result, Fig. 13.16 reproduces the final distribution of the cases of load flow divergence in terms of the Manic-Québec power flow. With respect to the diagram of

⁵Rather than starting the load flow computation from a flat voltage profile, the solution corresponding to the base case is used as an initial guess.

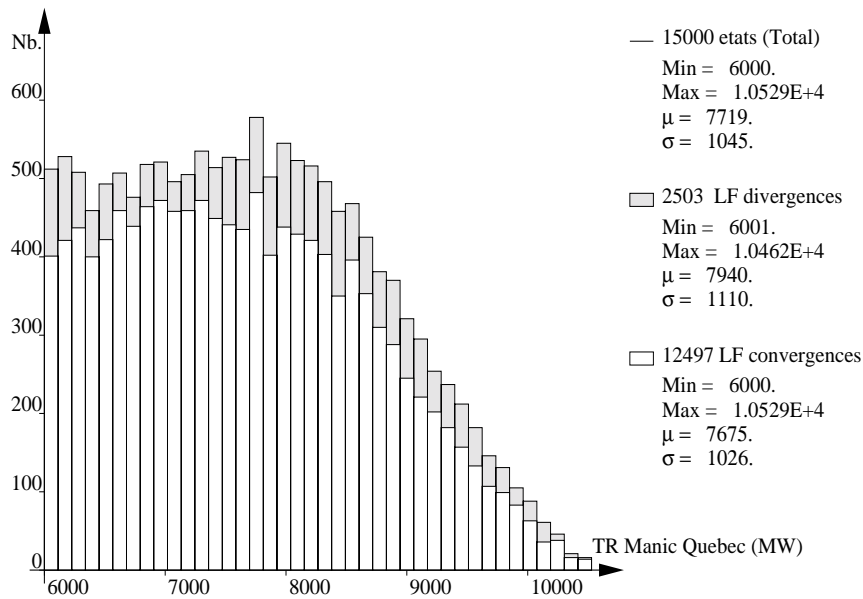


Figure 13.16 Convergence diagram of Manic-Québec power flow (12 base case files)

Fig. 13.15, one can observe that the proportion of divergences is strongly reduced and they are more or less uniformly distributed.

The above example indicates the possible difficulties one may encounter in generating a data base. A practical solution to this problem is sketched in Fig. 13.17. It consists of systematically generating in parallel the a priori and the a posteriori data base and analyzing the corresponding statistical distributions.

The a priori data base corresponds to the randomly selected variants. In the above example, 15000 such variants were drawn randomly. Each variant is described by a certain number of a priori defined attributes, corresponding to the independent variables and input variables of the load flow computation. In the above case, they correspond mainly to the power flows and corresponding generation vs load pattern as well as topology and availability of var compensators. Each such variant leads to a base case specification and an incremental input file for the load flow program. The latter are fed into the load flow computation, and the state is classified according to its convergence or non-convergence.

This data base may thus be analyzed with the statistical methods presented before in order to appraise the reasons of divergence, and to modify, if required, the data base generation or load flow algorithms. This analysis is also useful to identify early enough the correlations possibly introduced, which may influence the representativity of the actually obtained data base.

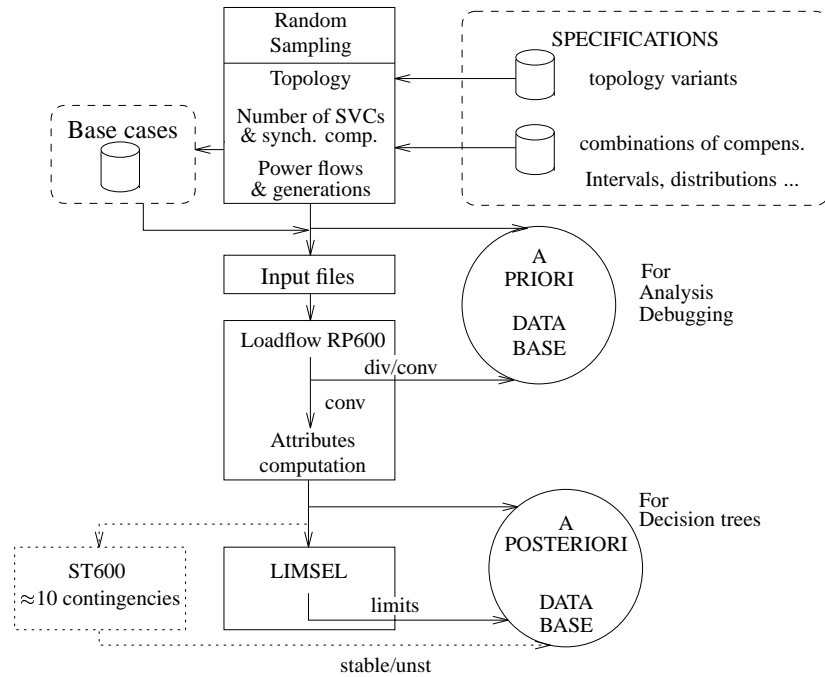


Figure 13.17 Data base generation procedure

The a posteriori data base is composed of the obtained states for which the load flow computation did successfully converge. In the case of the Hydro-Québec data base, 12497 such states were finally obtained. In addition to the independent variables characterizing the variants, each state may be described by additional attributes obtained as a result of the load flow computation. In particular, the effectively obtained power flows, generations and load level were considered in the present case.

The final data base generation phase took about one week of elapsed time on a SUN SPARC10 workstation used at 30% of the available CPU time. The total amount of uncompressed data is about 70Mbytes.

Stability classification via LIMSEL

The states of the a posteriori data base have been classified by using the LIMSEL program together with a snapshot of the on-line data base of stability limits made in August 1992.

For each state, the LIMSEL program receives information about its key variables and returns the existing stability limits corresponding to the state. While LIMSEL provides

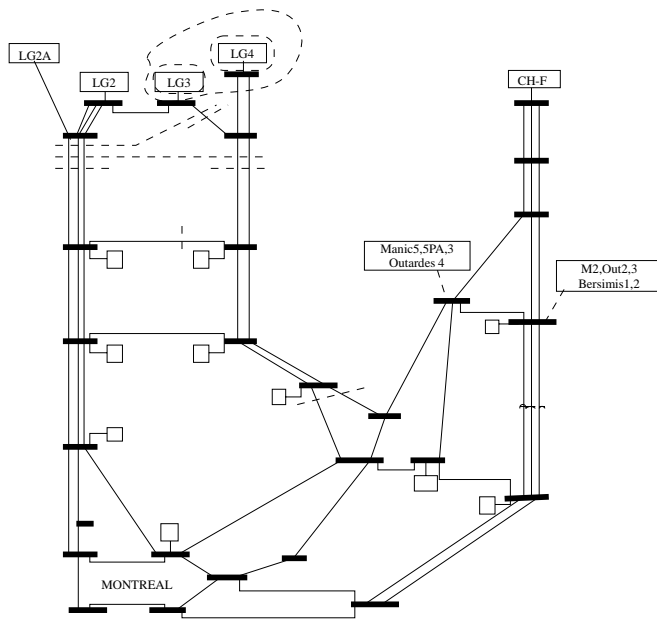


Figure 13.18 Groupings of generators or lines defining stability limits used for the global stability assessment

stability limits for each corridor, in our investigation we have exploited only the stability limits corresponding to the power flows and generations within James' Bay corridor. The corresponding set of constraining contingencies correspond to about 10 different single line faults all located in this corridor. The different stability limits used are identified by the dotted lines in Fig. 13.18; their values depend mainly on the topology and on the number of compensators in operation. If at least one of the actual power flows or generations is larger than the corresponding limit value provided by LIMSEL, the state is classified unstable. The above classification resulted in 3938 stable states and 8553 unstable states.

In addition, the relative difference between the limits and the actual values provide stability margins. They have been exploited to distinguish among the unstable test states, the fairly unstable ones from the marginally unstable ones. Namely, the marginally unstable states are states which do not violate stability limits by more than 2%. Thus if either the limit values were increased by 2% or the corresponding critical power flows were reduced by 2%, they would be classified stable. In the complete data base there are 393 such marginally unstable states. In addition, the stability margins have also been exploited in order to improve the tree quality (see 13.4.5).

As we have mentioned earlier, there are no stable states with a power flow larger than 8700 MW in the James' Bay corridor. However, this upper bound given by LIMSEL

is quite conservative, and in a stability classification based on numerical simulations we would expect to observe a significant number of stable states in this region. Thus, the sampling of power flows up to 10,000 MW power flows in the James' Bay corridor will be justified in the second stage of the research.

Classification by numerical integration

The dotted box in the left part of Fig. 13.17 shows the alternative approach to realize the stability classification, consisting of transient stability simulations, e.g. using the ST600 program of Hydro-Québec. In a future quantitative assessment, this would be a prerequisite to obtaining an unbiased comparison of the decision trees and the present strategy coded in the LIMSEL data base.

About 125,000 simulations would be required, considering that about 10 contingencies must be screened for each of the 12,500 states of the data base. Exploiting an equivalent power of 100 SUN SPARC2 stations to run the simulations in parallel these would take about two weeks using the ST600 program (which takes about 15 minutes/simulation). The same response time could easily be reached with a much smaller number of the faster workstations now available, which may offer more than ten times the computing power of a SUN SPARC2.

Two years ago, running these simulations would have been hardly feasible within acceptable response times. This justifies the fact that when the research project started it was decided to first evaluate the *functionalities* provided by the DTTS approach. This was indeed possible with a reasonable computational investment by exploiting the LIMSEL data base rather than numerical simulations. It is true that using LIMSEL as the reference does not allow us to extrapolate the error estimates straightforwardly; in particular, the decision trees obtained below could hardly outperform LIMSEL. Nevertheless, this approximate approach allowed us to get, through the derived decision trees, a good idea of the type of decision trees which could be obtained with a data base preclassified via SBS simulations. Using the ST600 program in the near future will make it possible to re-classify the data base and to compare quantitatively the performances of the decision trees with those of LIMSEL.

13.4.3 Global decision trees

The investigations summarized below are reported in detail in [WE 93f].

One of the main challenges for the decision tree method was to cope with the increased complexity due to the very high number of topologies covered in this study. Thus, in contrast to the research on the EDF system where we started our investigations with simple single-contingency trees, in the present research we start with the most complex global decision trees. In the next section we consider more elementary sub-

problems corresponding to subclasses of topologies and assess the potential advantages of problem decompositions.

Candidate attributes

The research was conducted in very close collaboration with the engineers responsible of the stability limit determination at Hydro-Québec, who proposed initially the following 67 candidate elementary attributes, all concerning the James' Bay corridor.

Topology. A total of 27 topological variables were used, comprising 5 attributes identifying classes of topologies, 17 elementary line status indicators and 5 attributes indicating the number of compensators and shunt reactors in operation in various substations.

Power flows and generation. A total of 40 power flows and generations were used, comprising 19 global power flows, 6 generations and 15 individual power flows of important lines.

In a second stage, the attribute list was completed with some combinations of the above, in particular some linear combinations and some power flows divided by the number of lines in operation in specific parts of the James' Bay corridor. This list of 87 candidate attributes thus contains practically the same information as the initial one, but in a more appropriate fashion for the decision tree method.

Pruning vs stop-splitting

Concerning the determination of the tree complexity, we have occasionally used the pruning approach (e.g. in the example of §3.4) and mostly the stop-splitting rule with $\alpha = 10^{-4}$. It was found that the optimal pruning level corresponds generally to α in the range $[5 * 10^{-5} \dots 10^{-3}]$, with a tendency of being slightly larger than in our preceding investigations.

Learning and test sets

In order to estimate the quality of the decision trees, we have kept the last 2497 states of the data base. The remaining 10,000 states were used as learning or pruning states.

In a preliminary investigation, learning sets of variable size were used to build the decision trees. Table 13.11 reports the results obtained with the basic list of 67 candidate attributes, a value of $\alpha = 10^{-4}$, and for various numbers N of learning states. We notice that the error rates stop improving when N reaches about 5,000 states, while the complexity of the trees as well as the number of test attributes increase

Table 13.11 *Tree characteristics for various learning set sizes*

N	P_e	$\#\mathcal{N}$	$\#\mathcal{A}$	N	P_e	$\#\mathcal{N}$	$\#\mathcal{A}$
1000	13.7	29	11	6000	8.6	125	29
2000	11.7	50	16	8000	8.0	157	30
4000	8.5	93	25	10000	8.0	207	35

further. Notice that except for the case of $N = 10,000$, where a single tree was built, the values provided in the table correspond to mean values of several trees built for randomly selected learning sets.

Linear combination attributes

The above error rates are rather high given the very large number of learning states; as indicated above, the first possibility investigated to improve them consisted of determining some combined attributes on the basis of the experience gained. This yielded indeed a significant improvement of quality. For example, let us consider the tree partially represented at Fig. 13.19 which is the direct cousin of the tree discussed in our illustrative example of §3.4, represented in Fig. 3.16, the latter being built with the augmented list of candidate attributes. Similarly to the tree of Fig. 3.16 a tree was constructed on the basis of the first 8000 states of the data base and $\alpha = 1.0$, which yielded an overall number of 703 nodes. It was then tested on the basis of a pruning set (PS) composed of 2000 states not used to build it and its pruning sequence was generated. Finally, the pruned tree partially represented in Fig. 13.19 was selected using the “1 standard error rule”. It reduces to 253 nodes and, on the basis of the 2497 test states (used neither as in its LS nor in its PS), yields an error rate of 7.17%.

Let us have a closer look at the two trees to further analyze the effect of using a richer set of candidate attributes. On the one hand, in the tree in Fig. 13.19 the attributes selected at the first two levels are respectively (i) “PLG” the total power generated in the Lagrande power plant; (ii) “Trbjo” which is the power flow in the western part of the James’ bay corridor; and (iii) “Nb_Comp” which is the total number of var compensators in operation in the corridor. On the other hand, in the tree of Fig. 3.16, the attributes selected at the two first levels are the following combined ones : (i) at the top-node and its left successor the linear combination of “Trbj”⁶ and the number of compensators “Nb_Comp”; (ii) at the right successor of the top-node the attribute denoted “Tr7069” which is the power flow in the northern part of the east corridor divided by the number of lines in operation in this part. Thus, the elementary attributes have been replaced by more sophisticated ones, leading to a more efficient discrimination among the stable and unstable states.

⁶“Trbj” denotes the total power flow in the James’ Bay corridor, which is equivalent to “PLG” the total generation of the Lagrande power plant

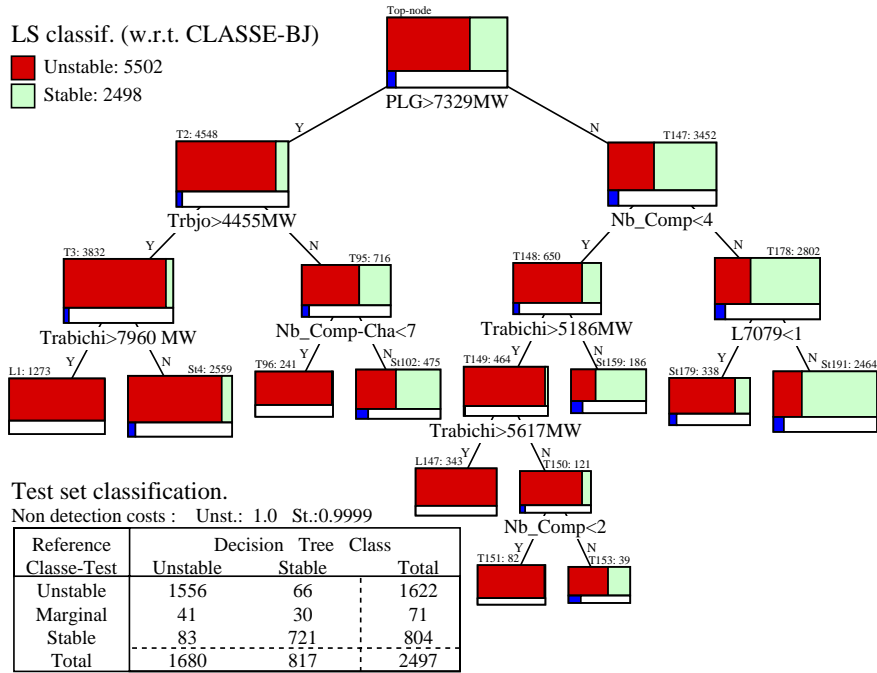


Figure 13.19 Partial view of decision tree built with 67 attributes : $N = 8,000$
 $M = 2497$

Table 13.12 Effect of improved attributes on tree characteristics,

Attributes	$P_e\%$	$P_{FA}\%$	$P_{ND}\%$	$P_{DE}\%$	$\#\mathcal{N}$	$\#\mathcal{A}$
67 basic (Fig. 13.19)	7.17	3.32	3.84	2.65	253	37
67 basic + 20 combinations (Fig. 3.16)	4.21	2.09	2.12	1.20	115	24

The comparison of the two trees is further summarized in Table 13.12, which indicates clearly that the richer list of candidate attributes is able to improve very significantly the tree quality. In particular, the proportion of dangerous errors is strongly reduced and similarly the complexity.

13.4.4 Problem decompositions

The above shows the interest of using composite attributes allowing us to take into account simultaneously several physical effects. However, although the resulting trees provide satisfactory accuracy, they are still quite complex and rather difficult to interpret. This suggests that the global stability problem formulated here is probably too complex to take full advantage of the decision tree approach. Hence the idea of decomposing

the data base into subdatabases corresponding to subclasses of configurations.

In the study reported in [WE 93f] several such decompositions have been systematically considered and the resulting decision trees were compared with the corresponding global trees. It is interesting to observe that all these decompositions have improved significantly the decision trees, even if some were more effective than others. For example, decomposing the overall data base into the three main classes of topology used in ref. [BE 91a], allowed us to reduce the mean error rate from 8% to 5.8%, while keeping the 67 basic candidate attributes.

In terms of interpretability, let us explain the resulting simplification on an example. Figure 13.20 illustrates a decision tree built for the subdata base corresponding to the 22-North configurations, i.e. situations where at least one line in the Western part A or B of Fig. 13.14 is out of operation and all lines in the Eastern part A and B are in operation. The tree, composed of 33 nodes is built on the basis of the 2746 such situations found among the 10,000 first states of the data base, using $\alpha = 10^{-4}$ and the 87 candidate attributes. It was tested on the basis of the 657 22-North situations among the last 2497 states of the data base, yielding a test set error rate of 3.50%, corresponding to 1.97% of non-detections and 1.53% of false alarms.

We first note the high simplicity of the tree; more importantly, it represents, according to the experts, sound information. In particular, each one of the selected attributes could be explained on the basis of the prior information available. This was possible thanks to the use of standard operating parameters to formulate it. Moreover, although the linear combination tests were slightly more difficult to interpret, this was compensated by the consequent simplification of the tree structure and its higher reliability.

13.4.5 Quality improvement

Similar to the EDF research project, we have also applied the quality improvement techniques in the present case. In particular, in order to reduce the number of non-detections, already very small in the standard trees, the classification of the decision tree was biased by biasing the classification of the learning states via an artificial reduction of the limits provided by LIMSEL.

This technique has shown to be rather effective. For example, shifting the limits by 4% allowed us to reduce the proportion of non-detections of a global tree to 0.56% (instead of 2.12%), while the rate of false alarms increased to 6.45% (instead of 2.09%). Even more effective results are obtained in the case of the tree of Fig. 13.20 where the non-detections reduce to 0.30% while the false alarms are increased only to 5.17%. The resulting tree is represented at Fig. 13.21; as indicated in the figure non-detection costs twenty times higher for the unstable states has been used, so as to bias the classification of the tree. On the other hand, due to the biased classification of the learning states the

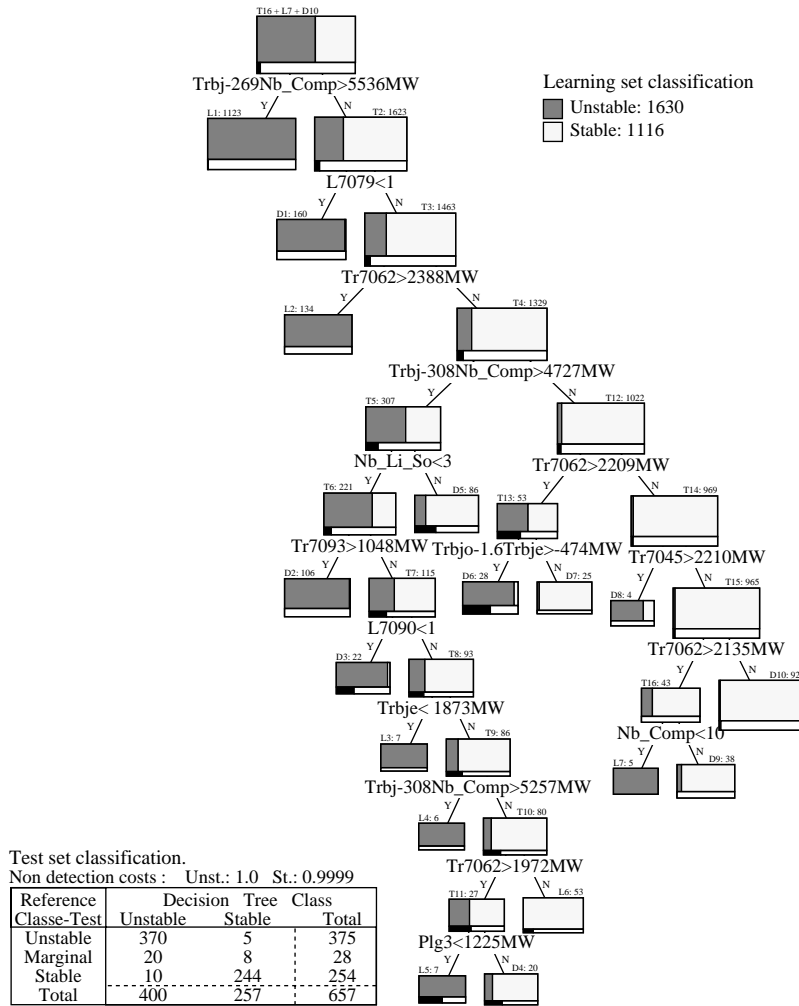


Figure 13.20 Decision tree built for the 22-North configurations : $N = 2746$ $M = 657$

thresholds have been adjusted, leading to pure stable terminal nodes, instead of those in Fig. 13.20 which contained often a small minority of unstable states.

13.4.6 Other approaches

For the sake of completeness and further appraisal of the decision tree approach, we provide some recent results we obtained with other learning methods.

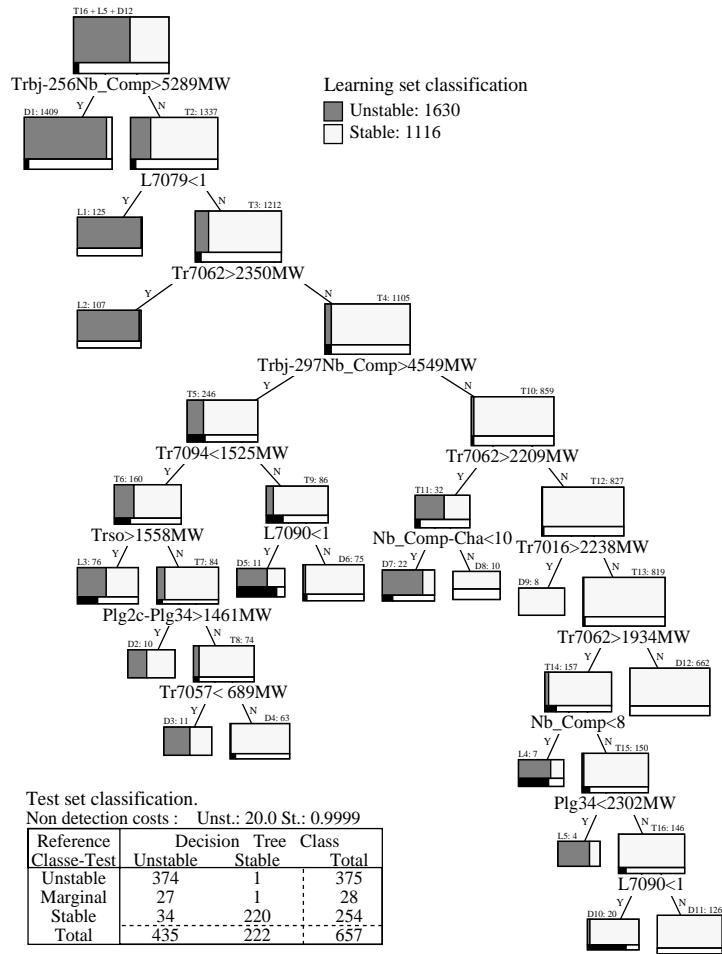


Figure 13.21 Improved DT built for the 22-North configurations : $N = 2746$ $M = 657$

Multilayer perceptrons

These results were already mentioned in chapter 5. We merely indicate that three multilayer perceptron simulations were carried out on the complete global stability problem, corresponding to the trees in Fig. 3.16 and 13.19. The test set errors rates obtained varied between 2.44% (obtained with the BFGS procedure and a “regularized” MSE criterion) and 3.93% (obtained with the BFGS procedure and a standard MSE criterion).

Thus, the decision trees appear to give very satisfactory results, provided that the richer

Table 13.13 $K - NN$ results for the Hydro-Québec system

K	1	3	5	7	9
67 candidate attributes					
$P_e\%$	12.58	11.33	10.53	10.21	10.25
24 attributes of DT of Fig. 3.16					
$P_e\%$	6.93	6.73	6.13	6.13	6.61

list of candidate attributes is used. More importantly, in addition to their very good performances in terms of reliability, they are able to tell us which physical relationships they have identified. We recall also the fact that the decision tree building is about two orders of magnitude faster than the backpropagation training procedure of the multilayer perceptrons.

Nearest neighbor

Table 13.13 shows the accuracy results obtained for the same global problem, with the $K - NN$ classifier for two different cases. The first set of results corresponds to the use of all 67 attributes in the distance computation⁷. The results are quite deceiving with respect to the decision trees and multilayer perceptrons. We note that the value of $K = 7$ provides the best results. The second set of results corresponds to using only the attributes identified by the decision tree of Fig. 3.16 : the reliabilities are significantly improved with respect to the preceding ones but the level of performance of the best DTs or MLPs are not reached; here again the value of $K = 7$ yields the best results. The well-known high sensitivity of the nearest neighbor to the attributes used in the distance computation (and more generally to the weights used in the distance) is observed here very clearly.

The comparatively good results obtained with the attributes selected by the decision tree suggest that using the latter and further adjusting the weights on the basis of the learning sample seems to be a promising direction.

13.4.7 Discussion and perspectives

It is still too early to draw definite conclusions about the application of machine learning methods to transient stability assessment of the Hydro-Québec system. Nevertheless, the above results suggest conclusions similar to those made in the context of the EDF system.

First of all, taking into account the fact that the limits implemented in the LIMSEL data base are representative of realistic stability limits, we may conclude that the decision

⁷The attribute values are however pre-whitened.

trees are indeed able to extract interesting and interpretable information, on the basis of a representative sample of power system situations. Compared with other pattern recognition methods, they are efficient and score well (if not the best) in terms of reliability. Further, the method provides means to suggest and enable experimentations with various types of attributes and problem decompositions, and this is yet another asset for the success of the method.

Coming back to our initial goal of evaluating the functionalities of the machine learning approach, we deem it has been reached even though no actual transient stability simulations were carried out for this research. We indeed found that the method is able to screen systematically very complex classes of situations and to determine stability characteristics. As an anecdote, we mention that during our investigations we have been able to detect a set of about 30 states for which a particular limit value was erroneously stored by LIMSEL; this was found to be a transcription error made when the LIMSEL data base was updated, and corrected subsequently.

More specifically, the overall problem was found to be slightly too complex to enable the extraction of easily interpretable security information without decomposing it, although the method could cope quite well with it from the reliability viewpoint. However, once a data base has been determined for such a broad class of situations, appropriate problem decompositions may be found out a posteriori by building various decision trees, in a trial and error fashion and on the basis of the information held in the data base. This helps us also to gain insight into the problem specifics.

The future research direction is clearly to use numerical simulations in order to pre-classify the data base states. Decision trees could be systematically built for problems decomposed in terms of both families of topologies and families of contingencies. These trees could then be compared with the present day practice, codified in the LIMSEL data base.

14

Voltage security

14.1 INTRODUCTION

The application of the decision tree approach to voltage security assessment was initially proposed by our research colleagues of EDF [GO 89b]; they were motivated by the voltage collapse incidents experienced in the EDF system [HA 90]. In 1990, a data base, constructed in the context of *emergency* state detection of the Brittany EHV subsystem [ZH 90], was thus exploited to yield a first set of decision trees [WE 91b]. During the same period, a student at the University of Liège investigated in his “final project” the decision tree based approach to *preventive* voltage security assessment, in a fashion similar to the single-contingency DTTS method [WE 91c, VA 93a].

Following these preliminary investigations, a research collaboration was initialized in early 1992 between the R&D department of EDF and the University of Liège, to explore feasibility aspects of the decision tree approach to emergency voltage insecurity detection. In addition to decision trees per se, simulation models and numerical tools were accordingly examined. The main results of this first research stage are summarized in [MI 92, WE 92a], and discussed below.

In mid 1993, the collaboration was diversified to encompass the development of appropriate data base generation tools, and a much broader multicontingency study, looking both at preventive and emergency wise security assessments. Although it is still too early to draw conclusions, we will describe the data base generation software and the first related results thus obtained.

Before concentrating on this broad EDF research, we recall the academic system study presented earlier to comment on the results obtained within the Statlog project with its data base.

Table 14.1 Results obtained in the Statlog project. Adapted from [TA 94]

Algorithm	Maximum Storage	Time(sec)		Error Rate %		
		Train	Test	Train	Test	
Statistical methods						
Lin. Discrim.	588	73.8	27.8	2.2	2.5	
Quad. Discrim.	592	85.2	40.5	3.6	5.2	
Logist. Discrim.	465	130.4	27.1	0.2	0.7	
SMART	98	7804.1	15.6	0.3	0.6	
Kernel. dens.	125	3676.2	*	2.6	4.4	
$K - NN$	86	1.0	137.0	0.0	5.9	
NaiveBay	276	17.4	7.6	4.6	6.2	
Machine learning methods						
TDIDT	Cart	170	135.1	8.5	0.9	3.4
	Indcart	293	86.5	85.4	0.7	3.4
	NewID	846	142.0	1.0	1.7	2.7
	AC2	222	1442.0	79.0	0.0	3.4
	BayTree	289	24.7	6.7	0.0	3.0
	C4.5	77	66.0	11.6	1.0	4.0
	Cal5	62	13.9	7.2	2.5	2.9
	Castle	279	230.2	96.2	2.9	4.7
	CN2	345	272.2	16.9	0.0	3.2
	ITrule	293	1906.2	41.1	4.3	6.5
Neural network methods						
Kohonen SOM	216	7380.6	54.9	2.6	5.6	
Dipol92	49	43.0	11.9	1.5	1.8	
MLP bprop	146	478.0	2.0	1.1	1.7	
Rad. Basis Fun.	NA	121.4	29.3	2.1	3.4	
LVQ	115	977.7	32.0	0.2	5.4	

14.2 ACADEMIC STUDY

In §10.2 of chapter 10 we described the problem formulation of emergency voltage insecurity detection on the basis of an academic type synthetic system designed for the purpose of experimentation. The corresponding data base was passed to the researchers of the Statlog project, who used it to compare a wide range of methods.

Table 14.1 collects the obtained results. We observe that projection pursuit (SMART) together with the logistic discriminant produce significantly better results than the other algorithms ($P_e \approx 0.65\%$); but SMART is about 50 times slower than the logistic discriminant. The neural network algorithms (MLP and Dipol92) provide also very good results ($P_e \approx 1.75\%$). The TDIDT algorithms (Cart, Indcart, NewID, AC2, BayTree, C4.5, Cal5) provide intermediate results ($P_e \approx 3.26\%$), similar to those

obtained in §10.2. On the other hand, the Kohonen SOM (and LVQ) as well as the $K - NN$ method are much less accurate ($P_e \approx 5.6\%$).

A possible explanation of the good performance of the linear model (Log. Discrim.) is the reduced problem size of the present example, which certainly plays in favor of the parametric estimation techniques. Thus, this is not likely to hold in general.

As already noted in §4.2.1 we observed the high sensitivity of the linear models (Lin. Discrim vs Log. Discrim.) to the learning criterion used. On the other hand, the results obtained by the various TDIDT approaches are quite close to each other, which suggests that these non-parametric approaches are quite robust with respect to changes in their learning criterion.

Since SMART has performed so well on our two power system security *classification* problems, it should certainly deserve further investigation. In particular, it should be possible to exploit very effectively security margins with this method, since it is actually a *regression* technique (see §4.3.2). For example, in ref. [WE 94c] we suggest how these regression techniques could be exploited usefully in the context of voltage security assessment.

In terms of reliability, we observe that decision trees score slightly less well for voltage security than for transient stability. This is probably related to the fact that in voltage security the individual attributes are less discriminating or, in other words, that the security boundaries are more diffuse. Whether this is a general property of emergency voltage insecurity detection is not yet clear. In §14.4.4 we will reconsider this comparison on the basis of a more realistic example.

14.3 PRELIMINARY INVESTIGATIONS

We briefly report on the investigations carried out on the EDF system in a preliminary stage of the research.

14.3.1 Preventive mode

The proposed method is a replica of the DTTS method. It assesses the ability of a precontingency state to withstand a preassigned contingency in terms of the state parameters preselected by the tree, built for this contingency [WE 91c]. It is worth mentioning that refs. [LI 89, LI 91] propose a quite different tree approach for the purpose of voltage optimization.

The precontingency states used for the tree building are obtained in a way similar to that of §10.1.2. The contingencies of concern here are generally single or double outages

of EHV transmission and/or generation equipment, and the question asked is whether the system will be able to reach an acceptable mid-term equilibrium in the minutes following the outage.

The power system behavior subsequent to the disturbance may be assessed via the various methods discussed in §8.2.2. In this research we used a simple post-contingency load flow computation. A state is thus classified “secure” or “insecure” according to whether the load flow converges or not towards an acceptable post-contingency operating state. Feasibility limits, such as upper and lower bounds on voltage magnitudes are evaluated at the solution point together with sensitivity coefficients, checked to ensure that a state may be classified as secure. Admittedly, this type of classification is quite simplified but sufficiently realistic, given the preliminary nature of the investigations.

The trees built in this context concern the Brittany region of the EDF system which has in the past experienced voltage problems [HA 90]. A data base composed of 2000 pre-fault operating states was generated, using a 320-bus, 55-generator, 614-branch model of the EDF system, representative for the modifications imposed in the Brittany region. The latter states were obtained by imposing random variations concerning (i) the active power generation schedule in a large enough region surrounding Brittany, (ii) the local reactive resources (power plant configuration, voltage set-points, HV and MV compensation, synchronous condenser), (iii) the regional active and reactive load level, (iv) single (400 kV or 225 kV) line or transformer outages. The candidate attributes used for the tree building comprise 21 EHV voltage magnitudes, 8 load or compensation levels, 47 power flows (through lines, transformers, and cut-sets), 13 active or reactive power generations, and 12 reactive power generation reserves.

Figure 14.1 gives a typical tree built for a contingency corresponding to the loss of a 600 MW generation unit in the study region. It was built with a value of $\alpha = 5 * 10^{-5}$ on the basis of $N = 1000$ states; it was tested on the remaining $M = 1000$ test states, and provided an error rate of 5.3%. Below the figure we indicate how the information quantity $NI_C^T = 818.8bit$ provided by the tree is shared by its different test attributes. Observe that more than 65% of the information is provided by the two first attributes “Qatcor” (the power flow through the 400kV/225kV transformers in an important substation) and “Res-Comb” (the reactive generation reserve in the power plants within or nearby the Brittany region). Note that the test attributes were selected among 101 candidates proposed to the tree building procedure.

While it is difficult to explain the reason why “Qatcor” was selected at the root of the tree, we mention that if we remove it from the list of candidates, “Res-Comb” will instead be selected. Figure 14.2 shows the projection of the secure and insecure states of the data base on the subspace of the above two attributes, and shows also the “hyperplanes” corresponding to the thresholds used in the tree. This scatter plot allows one to appraise correlations among the two attributes together with the way the decision tree discriminates among secure and insecure states.

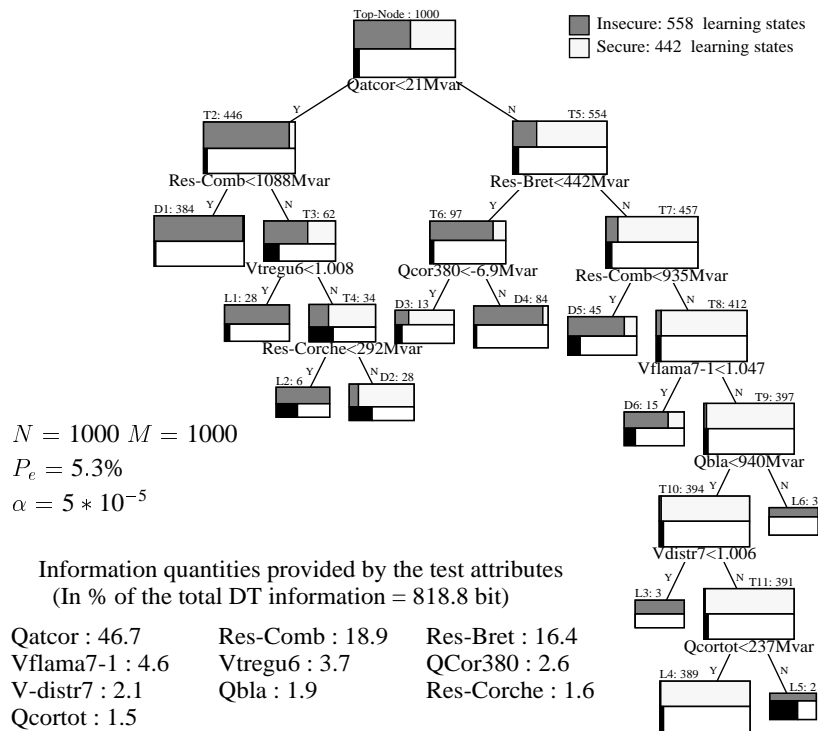


Figure 14.1 Preventive voltage security DT. Adapted from [WE 91c]

In addition to the above example, three-class trees were built, enabling one to distinguish among the insecure states those which may be corrected via the rapid action of gas turbines. Also, two-class trees were built for contingencies consisting of the loss of one or two circuits of an important 400kV line. These preliminary investigations, based on rather simplified modelling and security criteria and using learning sets of moderate size, were however able to show the potential of the decision tree approach for preventive voltage security assessment. In particular, their ability to provide physically sound and interpretable information was highly appreciated.

14.3.2 Emergency mode

In the context of emergency state detection, the proposed approach and resulting procedure are quite different from the previous ones. The leading idea is that voltage instability following a contingency generally does not develop as fast as the transient one (typically voltage collapse takes several minutes whereas electromechanical loss of synchronism takes only a few seconds); this leaves time to detect the potentially critical states after the contingency occurrence and to take corrective actions.

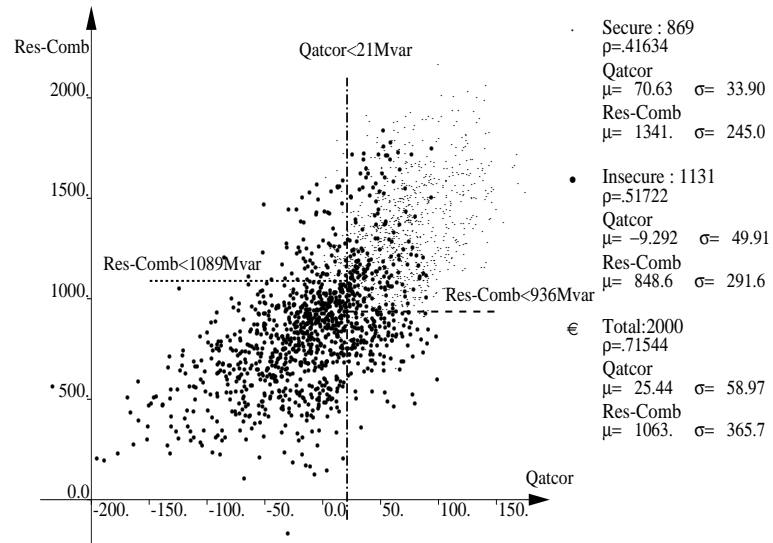


Figure 14.2 Distribution of 2000 random states in the $(Qatcor, Res-Comb)$ space. Adapted from [VA 93a]

A main difference of this method with respect to previous approaches is the type of considered system states. They result from various operating conditions, supposed to be subjected to a set of disturbances; they are determined after a short-term intermediate equilibrium has been reached, i.e. after the electromechanical transients have vanished (approximately 10-20 seconds after the disturbance inception). Such “just after disturbance” (JAD) states along with their classification (non critical if the state ultimately reaches a new acceptable equilibrium, critical otherwise) are used to build a tree, which therefore is relative to a set of disturbances. Subsequently, the tree may be used on-line to decide, in terms of JAD attributes whether, following a disturbance a system state is critical or not.

In the study described in [ZH 90] a data base composed of approximate JAD states was constructed for the Brittany region by using a simplified model, consisting essentially of a load flow computation using voltage sensitive load representations. In this data base the JAD states were generated directly without computing the corresponding precontingency states. Moreover, the procedure used aimed at generating a majority of borderline samples, in a particular kind of dichotomization approach.

The resulting trees, not reported here to save space, were of satisfactory accuracy but quite difficult to interpret. This was mainly due to the highly biased data base, where the generated states were correlated to the secure/insecure classification of previously generated states, destroying in particular the property of statistical independence [WE 91b].

14.4 PRESENT DAY RESEARCHES FOR EMERGENCY MODE VOLTAGE SECURITY

This section deals with the first stage of our research in collaboration with EDF, where emergency voltage insecurity detection is a main objective. Based on the experience reported in §§14.3.1 and 14.3.2, it was decided to develop a new data base generation approach. This consisted mainly of adapting the procedure described in §8.2.2 to the specifics of the EDF system. In particular, the JAD states are obtained in a two step procedure : (i) generation of a representative sample of independently drawn normal prefault states; (ii) application of various disturbances to yield the corresponding contingency specific JAD states, and possibly merging the latter to build multicontingency trees.

The main advantage is that this approach uses basically the same philosophy as the data base generation for preventive security assessment. It allows us to control the statistical representativity independently of the pre-disturbance states and of the disturbances themselves, which is paramount for the validation of the resulting security criteria. Further, it allows us to carry out in parallel preventive and emergency wise security assessment on the basis of the same data bases which may provide interesting analysis possibilities, as we will illustrate below.

14.4.1 Data base generation

Level of modelling

In addition to the above methodological changes in the data base generation, it was deemed necessary to use a more realistic model of the power system and thence to use more sophisticated simulation tools. With respect to the previous models there are mainly two refinements.

Secondary voltage control. To generate representative sets of pre-disturbance situations the effects of secondary voltage control and automatic shunt compensation have been taken into account. These contribute much to maintain predisturbance voltages close to their nominal values and influence strongly voltage security limits as well as the reactive generation and EHV voltage patterns observed in the normal or JAD states.

Time-domain simulation. In the previous data bases we used only post-contingency load flow computations to classify and compute the candidate attributes in the JAD state. Here we simulated the system evolution and reproduced the sequence of events following a disturbance inception, in order to take into account OLTC and rotor field current limitation delays interacting with secondary voltage control actions. For this purpose, a simplified voltage stability oriented time-domain simulation method

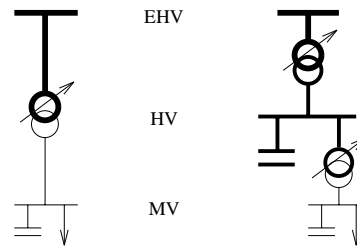


Figure 14.3 *Compound OLTC - Load - Compensation model*

developed at the University of Liège [VA 93b] was adapted to the specifics of the EDF system. The main advantage of this method with respect to standard time-domain numerical integration is computational efficiency, allowing one to handle a realistic large-scale system with acceptable response times. However, to limit complexity a simplified representation of the effect of EHV/HV and HV/MV transformers was used in the form of cascades of transformers as represented in Fig. 14.3. This model representation allowed us however to simulate the voltages at the HV side of the EHV/HV transformers, which have been shown as interesting attributes (see §10.2 and below).

In addition to providing a reasonably detailed level of modelling, the above improvements allowed us also to gain significantly in flexibility. They provided quite satisfactory data bases; from a physical viewpoint they avoided major simplifications which could be misleading.

Figure 14.4 shows the general principle of the data base generation including (i) the random sampling of variants; (ii) the validation of the variants via the load flow computation, and the simulation of steady state secondary voltage control and automatic HV shunt compensation effects; (iii) the computation of prefault attributes (for use in preventive security assessment); (iv) the computation of the JAD states corresponding to a snapshot at a preselected time τ along the post-contingency trajectory; (v) the subsequent simulation of the mid-term dynamics until either a voltage collapse is diagnosed or MV voltages are restored within dead-bands around nominal values; (vi) the computation of an approximate post-contingency load power margin, which amounts here to simulating a sequence of steps of the load demand in the Brittany region, and to observing the resulting dynamics [MI 92].

Generation of pre-disturbance operating states

The one-line diagram of the EHV (225kV and 400kV) system in the study region is represented in Fig. 14.5. The random variations made in the data base generation concern the topology as well as the load level, pilot node voltage set-points and active generation schedule.

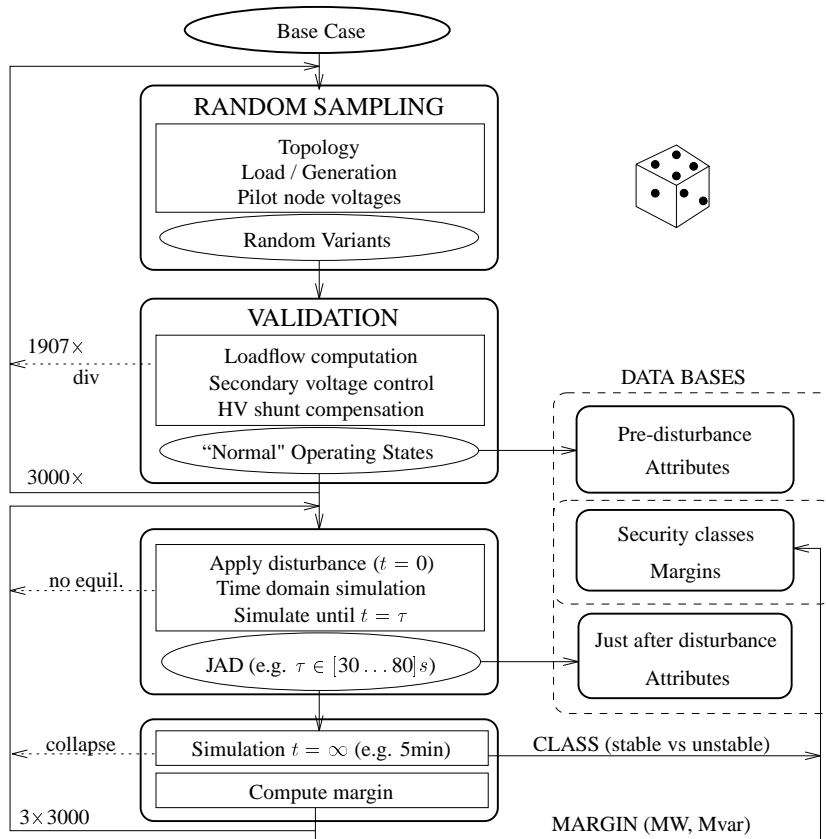


Figure 14.4 Principle of the data base generation

The topology variations consist of one (40%), two (50%), or three (10%) simultaneous outages of lines (mainly 400kV, and some 225kV) or transformers.

The load level is varied according to a uniform prior distribution in the interval [6000 . . . 9000] MW; as an illustration of the effect of load flow divergence, we show at Fig. 14.6 the a posteriori distribution of the load levels in the data base. Observe the effect of the filtering introduced by the load flow divergence, gradually increasing for increasing load levels.

The active generation scheduled within the region by the three power plants outlined in Fig. 14.5 was fixed by a random sampling of combinations of generation units in operation, so as to control the level of power imported from the remaining system (each unit in operation is supposed to operate at its nominal active power rating). The reactive generation within the region is fixed according to the secondary voltage regulation criterion, which essentially aims at controlling voltages at the pilot nodes,

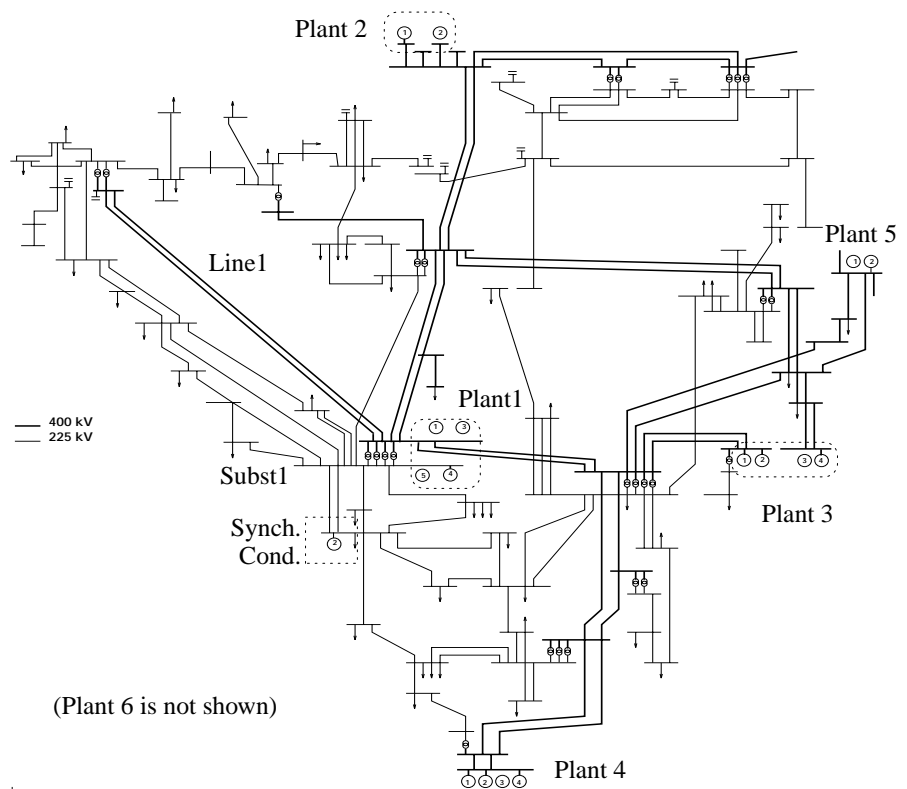


Figure 14.5 One-line diagram of the study region.

while maximizing the total reactive reserve. Pilot node voltage set-points are drawn randomly according to a Gaussian distribution around their usual values, to take into account tertiary voltage control by generating sufficiently diverse situations [MI 92].

The resulting distribution of the regional load level vs the level of active power import and vs the reactive reserve available in the three power plants, are depicted graphically in Fig. 14.7 for the pre-disturbance states contained in the data base. Each one of the strips in the left-most scatter plot corresponds to a particular combination of units in operation. In the right-most scatter plot we appreciate the effect of secondary voltage control, able to maintain a rather high reactive reserve, even for relatively high load levels.

Simulated disturbances and JAD states

Three disturbances have been studied, namely (see Fig. 14.5): (i) loss of one generating unit in operation in plant 1 (generating about 600MW); (ii) loss of one circuit of line

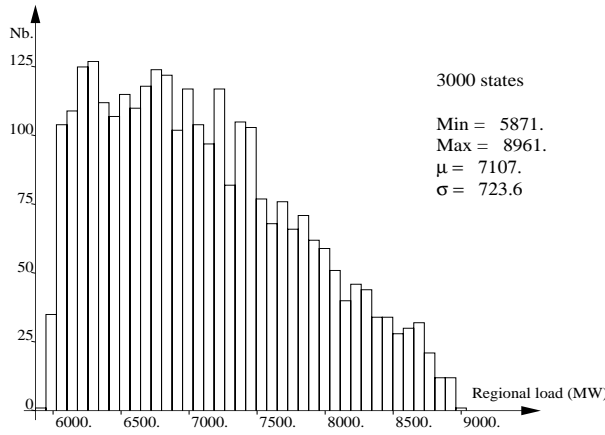


Figure 14.6 Histogram of the regional pre-disturbance MV load level

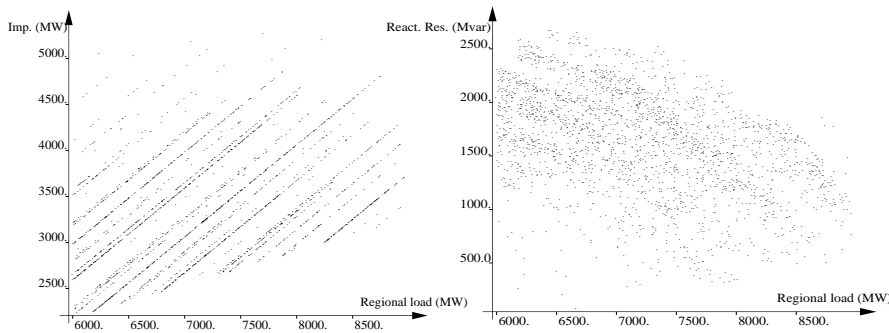


Figure 14.7 Power import and reactive reserve in the study region

1; (iii) 400kV busbar fault in substation 1 (leading to the loss of up to three 400kV lines entering the substation, one generation unit of 600MW, and two 400kV/225kV transformers).

For each disturbance the JAD states are normally considered at $\tau = 30s$ after the disturbance inception. At this time instant machine excitation limits are already active, and the first actions of the secondary voltage control have been applied. On the other hand, the OLTC delays are larger than 30s.

Attributes

The candidate attributes are computed from a system snapshot either in the pre-disturbance state or at a given time instant ($\tau = 30s, 45s$ or $80s$) in the post-disturbance state. The latter will be called in the sequel JAD or emergency mode attributes whereas

the former will be called normal or preventive mode attributes. There are also some attributes which take into account information from both the JAD state and the normal state, in order to quantify the impact of the disturbance on the electrical state of the system.

The first list (list 1) of candidate attributes contained the following 154 readily available real-time attributes.

HV voltages, at the HV side of the 39 EHV/HV transformers represented explicitly in our study.

EHV voltages, characterizing 29 important 225kV and 400kV buses in the study region.

Power flows, corresponding to the active and reactive flows in 30 EHV lines.

Topological indicators, of 12 lines which may be out of operation in the pre-fault situation.

Load, active and reactive MV load levels of the region.

Reactive reserves, of 8 individual and 4 combinations of power plants, corresponding to the difference between the reactive generation and its upper capability limit.

In preventive mode the preceding are contingency independent attributes computed in the normal state and will be denoted by “list 1a” in the sequel. On the other hand, in the JAD state they depend both on the contingency and on the time instant τ and will be denoted by “list 1b” (resp. c, d) for $\tau = 30s$ (resp. 45s, 80s).

In addition, the following two attributes were also used in some simulations, although they call for more complex computations.

Delta-Pc, the variation in the active MV load level due to the voltage sag caused by the disturbance.

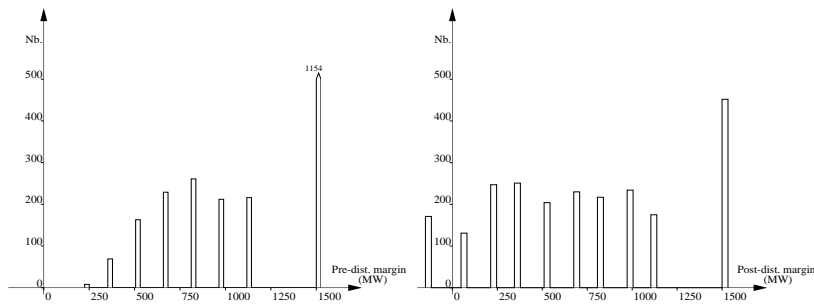
Marge-P-Pre, the pre-disturbance load power margin, i.e. the amount of additional regional load demand which may be delivered without the system becoming unstable.

Stability criteria, load power margins and classifications

A system evolution is considered to be unstable if either it cannot reach a solution of the short-term equilibrium equations or it reaches an unacceptable equilibrium. The latter may correspond to unacceptable EHV or HV voltages and/or unacceptable sensitivity coefficients, or to a situation where the post-contingency load power margin is below a pre-determined threshold Ma [MI 92]. A JAD state corresponding to an unstable future evolution is denoted *critical*, whereas a normal state is denoted *insecure* with respect to a contingency if the latter yields unstable behavior.

Table 14.2 *Proportion of unstable situations*

Contingency		Nb. of relevant JAD states	% of “unstable” states	
No.	Description		$Ma = 0\text{MW}$	$Ma = 300\text{MW}$
1	Loss of 600 MW	2312	7.4	23.7
2	Loss of 400 kV line	2000	1.6	8.1
3	Busbar fault	2000	20.0	33.5

**Figure 14.8** *Pre- and post-disturbance active load power margin distributions (the relevant 2312 states for disturbance number 1)*

In our simulations we have used either $Ma = 0\text{MW}$ or $Ma = 300\text{MW}$, to define two possible classifications. Table 14.2 shows the corresponding percentages of unstable scenarios among the relevant states of the data base. Figure 14.8 shows the normal state load power margin distribution and the corresponding post-disturbance distribution after the loss of 600MW of generation in plant1. The states which have a negative margin are arbitrarily set at -150MW; the states which correspond to a very large margin are arbitrarily set at 1500MW.

Considering the proportions of unstable states, we note that the normal disturbances corresponding to a single line or generator trip, lead to a very small proportion of unstable situations, even if the security criterion requires a post-disturbance margin of 300MW. This limits somewhat the representativity of the unstable states of the data base, and provision was taken in the next generation data base discussed in §14.5, so as to obtain a sufficient number of unstable states. In spite of this limitation, many interesting aspects could be investigated on the basis of this data base and it allowed us to gain experience in the context of voltage security and in particular emergency state detection. We will further see that the obtained results are already very promising in comparison to the criteria presently in use at EDF.

14.4.2 Overview of obtained results

Single-contingency trees have been built for classifying the JAD states of the three above contingencies, for various candidate attributes and two different security classifications. In addition, multicontingency trees were grown for the *union* of the three corresponding data bases of JAD states. Finally, a complementary investigation was carried out with the first disturbance, to analyze more systematically candidate attributes and quality improvements.

In this section we briefly report on the general results obtained with the three contingencies. In the next section we will focus on the more in-depth analysis of the first contingency.

Single contingency trees

Table 14.3 summarizes the performances of the decision trees obtained for the three contingencies, two classifications and two lists of candidate attributes. All trees have been constructed on the basis of half the data base states and tested on the remaining half. The pruning parameter of $\alpha = 10^{-4}$ was used in the stop-splitting criterion. For each decision tree we indicate its specifications (the candidate attributes and the classification) together with its main characteristics (number of nodes, of selected attributes; accuracy assessment). The last column indicates the type of information of the most salient attributes selected by a tree and the amount of its information as a percentage of the total tree information.

Note that the dangerous errors (DE) correspond to non-detections (ND) which are unstable with a margin less or equal to zero. For the trees built with respect to a post-disturbance margin of 300MW, the false alarms (FA) are states classified insecure by the tree although their margin is larger than 300MW; the non-detections are states classified secure although their margin is smaller than 300MW, and include the dangerous errors. On the other hand, for the trees built with respect to a post-disturbance margin of 0MW, the false alarms are states classified insecure while they have a margin larger than 0MW, whereas the non-detections reduce to the dangerous errors.

The following tendencies may be observed.

- Without margin (DTs no. 1 and 5), the trees select HV voltages as the most interesting attributes.¹ This behavior was also observed in the academic example. It may be “explained” by the fact that the HV voltage sag observed in the JAD state reflects at the same time the strength of the disturbance and the amount of load which must be restored by the action of the tap changers. A further analysis of the scores provided by the individual HV voltages at each tree node shows that the

¹For contingency no. 2 no meaningful tree could be built due to the small number of unstable states.

Table 14.3 *Single-contingency decision tree performances*

Dt.	Specifications		Decision tree characteristics						
No.	Cand. Atts.	Ma	$\#N$	$\#A$	$P_e\%$	$P_{FA}\%$	$P_{ND}\%$	$P_{DE}\%$	Types of Atts.
Loss of 600MW in plant 1									
1	List1b	0	15	7	4.20	1.47	2.77	2.77	HV volt. (81.5%)
2	List1b	300	21	9	8.48	3.81	4.67	0.17	Reac. res. (76.5%)
3	+ Delta-Pc	300	19	8	7.09	4.33	2.77	0.00	Delta-Pc (77.6%)
Loss of one circuit of 400kV line 1									
4	List1b	300	17	8	4.60	2.80	1.80	0.00	Reac. res. (60.9%)
Busbar fault in substation 1									
5	List1b	0	25	12	7.30	2.30	5.00	5.00	HV volt. (56.8%)
6	List1b	300	29	12	11.01	4.80	6.30	1.40	Reac. res. (61.8%)
7	+ Delta-Pc	300	23	9	7.10	3.90	3.20	0.10	Delta-Pc (77.6%)

method hesitates among various more or less equivalent HV voltages, due to the high correlation among them. A more robust approach discussed in §5.3, would consist of using appropriate mean values based on the identification of voltage coherent regions. The latter may be achieved by using an appropriate clustering method to exploit the statistical information contained in a data base of JAD states.

- Including a margin of 300MW in the security criterion (DTs no. 2, 4 and 6) causes the reactive reserve attributes to be selected in preference to HV voltages. A possible explanation lies in the fact that this security criterion is more “preventive like”. The trees become slightly more complex, which merely reflects the higher number of unstable states in their learning sets, and almost all the dangerous errors are removed.
- The use of the “Delta-Pc” attribute further improves significantly the trees by reducing their complexity and number of errors. This attribute is a weighted mean of the MV voltage sag, taking into account the sensitivity of the load to MV voltage variations as well as the amount of load connected to each EHV bus. This “clever” combined attribute was suggested by the analysis of previous trees, as a possible robust combination of more elementary attributes.

To provide a further ground for appraising the accuracy of the above trees, we have applied to the JAD states of each disturbance the criterion presently in use at EDF, which merely consists of blocking the OLTCs as soon as the EHV voltage at a particular given node is below a pre-determined threshold. Its comparison with the reference classifications defined above is indicated in Table 14.4, where the proportions of its various types of errors are indicated. They are defined in similar fashion to the classes of errors of the corresponding trees so as to allow a straightforward comparison.

Comparing these figures with those of the decision trees given in Table 14.3, we

Table 14.4 Presently used criterion

Ma	P_e %	P_{FA} %	P_{ND} %	P_{DE} %
Loss of 600MW in plant 1				
0	16.91	16.39	0.52	0.52
300	13.62	6.57	7.05	0.52
Loss of one circuit of line 1				
0	8.60	7.60	1.00	1.00
300	11.45	5.80	5.65	1.00
Busbar fault in substation 1				
0	29.75	29.15	0.60	0.60
300	23.65	19.40	4.25	0.60
Three contingencies (weighted means)				
0	18.35	17.65	0.70	0.70
300	16.11	10.39	5.72	0.70

conclude that the decision trees built with respect to $Ma = 300MW$ are able to make fewer dangerous errors and at the same time significantly less false alarms than the presently used criterion. On the other hand, the decision trees constructed with $Ma = 0MW$ make slightly more non-detections, but their low false alarm rates suggest that it would be possible to further improve the trees by using an intermediate margin threshold $0 < Ma < 300MW$.

Multicontingency trees

One of our initial objectives was to identify the risk of voltage collapse on the basis of information acquired from *available* system measurements in the JAD state. In particular, the criteria should not rely on information concerning the past (i.e. pre-disturbance) system states nor on the disturbance identification. Thus, although only three disturbances have been analyzed it was deemed interesting to consider a multicontingency decision tree by merging the three data bases, and to analyze its characteristics.

A learning set of 3156 states was obtained by merging the 3 learning sets, and a test set of 3156 states by merging the 3 test sets. As before, a tree was first built on the basis of the classification with $Ma = 0MW$, then with $Ma = 300MW$. In these simulations the basic list of elementary candidate attributes was used (List1b) so as to obtain trees with the desired real-time features. In Table 14.5, the multicontingency trees are summarized and compared with the corresponding values obtained by the single contingency trees². (Note that, for comparison purposes we have used for the loss of line1 in the case of $Ma = 0MW$ a single-contingency “default” tree which

²i.e. the *total* number of different test attributes, the total number of nodes and the weighted mean values of the various error rates

Table 14.5 *Multicontingency tree performances*

Specifications		Decision tree characteristics						
Cand. Atts.	Ma	#N	#A	$P_e\%$	$P_{FA}\%$	$P_{ND}\%$	$P_{DE}\%$	Types of Atts.
Multicontingency trees								
List1b	0	35	17	4.02	1.52	2.50	2.50	Reac. res. (50.7%)
List1b	300	51	21	7.95	5.26	2.69	0.22	Reac. res. (65.9%)
Weighted mean of single-contingency trees								
List1b	0	41	19	4.36	1.27	3.10	3.10	HV volt. (70.04%)
List1b	300	57	24	8.08	3.80	4.28	0.51	Reac. res. (66.90%)
Presently used criterion for three contingencies (weighted means)								
	0			18.35	17.65	0.70	0.70	
	300			16.11	10.39	5.72	0.70	

corresponds to a single node tree classifying all states as secure.)

Comparing the multicontingency with the single-contingency trees we may observe that the number of nodes of the former is slightly smaller than the total number of nodes of the latter. The multicontingency trees are also slightly more accurate in the mean. They have thus been able to exploit similarities among unstable JAD states. Further, we may see that the HV voltage attributes disappear from the multicontingency trees, which suggests (and confirms) that they are rather contingency specific. This is also confirmed by the fact that the multicontingency trees are rather robust; in particular, they show the ability to classify JAD states corresponding to disturbances not used in their learning set without important performance degradation. They are also able to detect unstable states for the weaker contingencies which cannot be covered as well by single contingency trees. However, the single-contingency trees are easier to interpret since they are less complex and correspond to a more elementary physical problem.

Finally, comparing the above results with the mean results corresponding to the presently used criterion, we may see again that the trees constructed by incorporating a load power margin in the criterion are much more efficient than the admittedly very conservative EHV voltage criterion, presently used. They are better in terms of their ability to identify the critical situations and at the same time are able to reduce, by a factor of two, the proportion of false alarms. Further, the multicontingency trees built without using the margin are naturally less effective in terms of identifying unstable states, but they are able to reduce the proportion of false alarms by a further factor of two.

Thus, although the decision trees obtained here for emergency voltage insecurity detection are of lower reliability than trees obtained for transient stability assessment, their potential advantages with respect to present day practice appears clearly from the preceding analysis. We will further illustrate below how the machine learning

methodology offers a flexible framework for the systematic analysis of JAD states and of alternative security criteria.

14.4.3 Further investigations on contingency number 1

Although the third, more severe contingency leads to a higher number of unstable states, it was not considered to be representative of the “usual” disturbances; thus complementary investigations were rather carried out on the first contingency, corresponding to the loss of 600MW of generation in plant 1. For this contingency various trees were constructed, all with respect to the second classification taking into account a margin of 300MW. The latter was indeed deemed to be more representative of the conservative criteria sought in practice. In these investigations we first analyzed various effects related to the candidate attributes and then applied additional techniques likely to improve the quality of the detection, in particular the hybrid DT-ANN approach.

Effect of candidate attributes

We first analyze the influence of the time instants τ corresponding to the JAD state, then consider decision trees using attributes computed in the pre-disturbance state, leading to preventive security assessment criteria, similar to those discussed in 14.3.1.

A. Various measurement instants τ

We first analyse the effect of the OLTC driven dynamics on the pattern of insecure states.

For this purpose, in addition to the tree no 2 described in Table 14.3 (corresponding to $\tau = 30s$), we built two other trees on the basis of JAD attributes determined respectively at $\tau = 45s$ and $\tau = 80s$. The three corresponding trees are represented in Fig. 14.9, where the notation “Qr” is used for reactive reserve attributes of various combinations of generation plants, “Ln” for active or reactive power flows, and “EHV” (resp. “HV”) for EHV (resp. HV) voltage magnitudes, the latter being expressed in p.u..

Considering the three trees, we observe first that they are of similar complexity. On the other hand, we note that increasing the measurement delay leads to a significant increase in reliability, in particular in terms of non-detections. Further, the EHV voltage attributes appear gradually in the trees, providing about 70% of their information. This reflects the physical fact that after the initial delay of 30s, the EHV/HV transformers start increasing their ratios, which tends to decrease the EHV voltages in the attempt to restore the HV voltages (see Figs. 8.6 and 8.7). Again, the fact that the EHV voltages are correlated is reflected by the similar scores they obtain at the various nodes of the tree, and this tends to make the selection of a particular voltage depend strongly on the random nature of the learning sets.

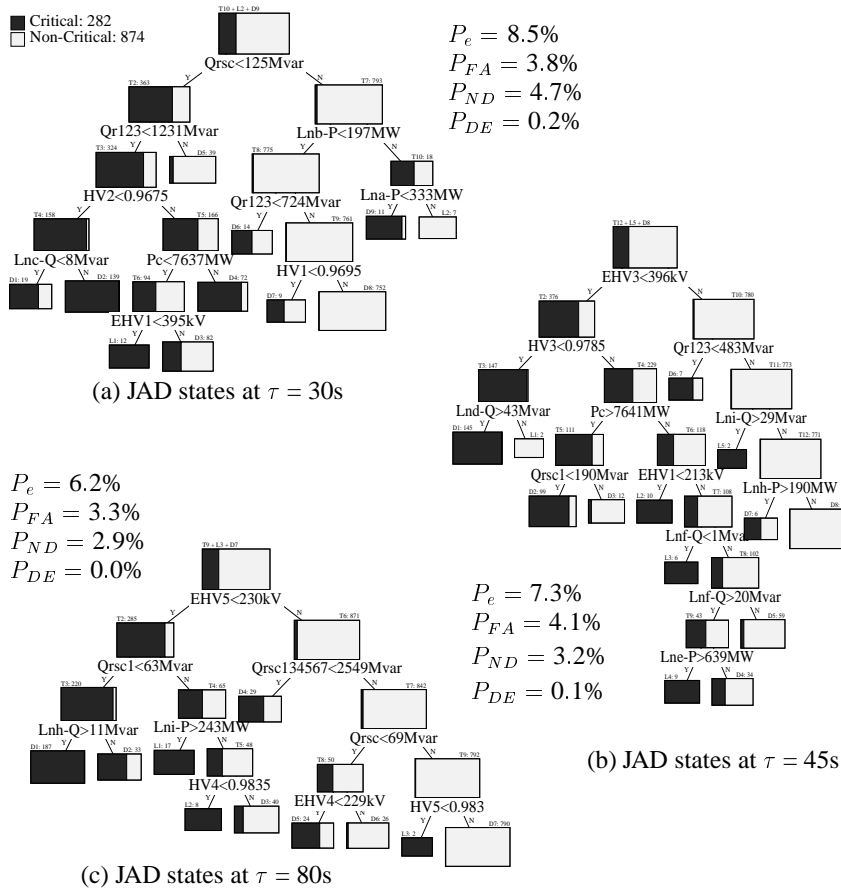


Figure 14.9 Emergency mode detection criteria for various measurement instants

It is also interesting to assess the robustness of the criteria with respect to uncertainties in the measurement instant, by using decision trees built for a given τ to classify JAD states corresponding to different τ 's. For instance, the tree of Fig. 14.9a built for $\tau = 30s$ yields an error rate of 14.3% (corresponding to $P_{FA} = 8.1\%$ and $P_{ND} = 6.2\%$) if applied to classify JAD states obtained at $\tau = 80s$. On the other hand, the tree of Fig. 14.9b, built for $\tau = 45s$, yields an error rate of 8.5% (corresponding to $P_{FA} = 3.0\%$ and $P_{ND} = 5.4\%$) when used to classify the above JAD states at $\tau = 80s$.

Thus, the action of the OLTCs changes significantly the outlook of the critical states. Before their action, low HV voltages are a symptom of insecurity, but as soon as the tap changers start acting these voltages start increasing, which makes the JAD states look “less insecure”. On the other hand, EHV voltages, as well as reactive reserves, have a more monotonic behavior, since the action of the tap changers will make the

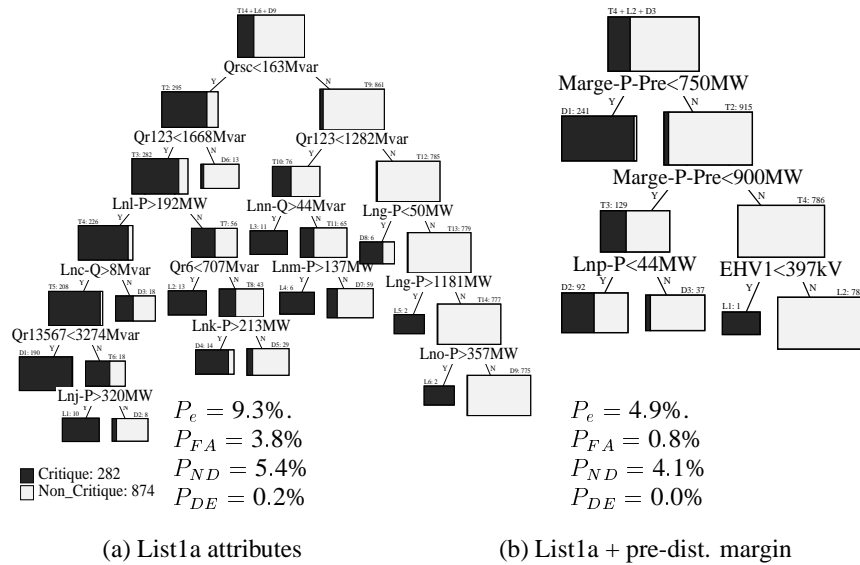


Figure 14.10 Preventive mode decision trees built with pre-disturbance attributes

weak situations look even weaker later on. Thus, criteria formulated in terms of the latter kind of attributes are probably more effective in practice, even though they may be seem to be less effective in the very first time instants following a disturbance.

B. Pre-disturbance attributes

A further comparison was made by constructing a tree for the above disturbance and taking into account a margin of 300MW, on the basis of the “List1a” of candidate attributes evaluated in the pre-disturbance state, yielding thus a preventive voltage security criterion. This tree is represented in Fig. 14.10a; we notice its similarity with the emergency mode tree of Fig. 14.9a : the same attribute is selected at the root node, corresponding to the reactive reserve available from the synchronous condenser feeding the study region (see Fig. 14.5).

It is also interesting to compare this tree with the one represented in Fig. 14.1, which corresponds to the same disturbance but to a different stability criterion (in particular not taking into account a load power margin of 300MW) and a slightly different base case condition and random generation of the data base. In spite of these main differences, similarities may be observed : the two trees exploit mainly reactive reserve attributes; the EHV voltages are used only very marginally. This expresses the fact that in the highly compensated systems the EHV voltage profiles are rather flat, independent of the distance to insecurity. This is even more apparent in the tree of Fig. 14.10a. A possible explanation of this is that in the latter case the effect of secondary voltage control was modelled, which leads to almost constant pre-disturbance EHV voltages,

while it was neglected in the preliminary data base used to construct the tree of Fig. 14.1.

The fact that the preventive mode tree is less (although only slightly) accurate and at the same time more complex indicates that the post-disturbance attributes are more discriminating than the pre-disturbance ones. However, in preventive-mode security assessment it is possible to exploit more sophisticated attributes. For example, the tree represented at Fig. 14.10b was constructed by including in the list of candidate attributes the *pre-disturbance* load power margin (denoted in the tree by “Marge-P-Pre”). Admittedly, this simplifies very significantly the resulting tree structure, while strongly improving its accuracy. It is worth mentioning that 96.5% of the information quantity provided by the tree is provided by “Marge-P-Pre” and only 3.5% by the two other attributes. Note also that the rather coarse determination of the load power margin (with steps of 150MW) certainly reduces its discriminating power.

The simplicity of the tree enables straightforward interpretation, as follows.

if the *pre-disturbance load power margin* is smaller than 750MW,
then the *post-disturbance load power margin* is smaller than 300MW;
otherwise if the *pre-disturbance load power margin* is larger than 900MW,
then the *post-disturbance load power margin* is larger than 300MW;
otherwise a more refined analysis should be made to determine the security.

This example highlights how decision tree building may provide interesting information about the relationship between values assumed by long-term preventive mode security margins and the system capability to withstand disturbances. Moreover, the above attribute is a contingency independent security index, characteristic of the overall system robustness. It may be available in many control rooms, determined with standard on-line security assessment tools such as the one described in [LE 90a]. Another complementary possibility explored in [WE 94c] would consist of approximating the value of the post-disturbance margin, for a given disturbance, in terms of the parameters characterizing the pre-disturbance state. Its pre-disturbance load power margin could be one, among others.

Quality improvements

The purpose of the investigations reported below was twofold : (i) to assess to what extent a very simple criterion, exploiting only two or three attributes selected by a tree, may be used to reliably identify critical situations; (ii) to evaluate the capability of the hybrid DT-ANN approach to provide accurate criteria for emergency voltage insecurity detection.

A simple two-level tree structure, as shown in Fig. 14.11, was used for this purpose,

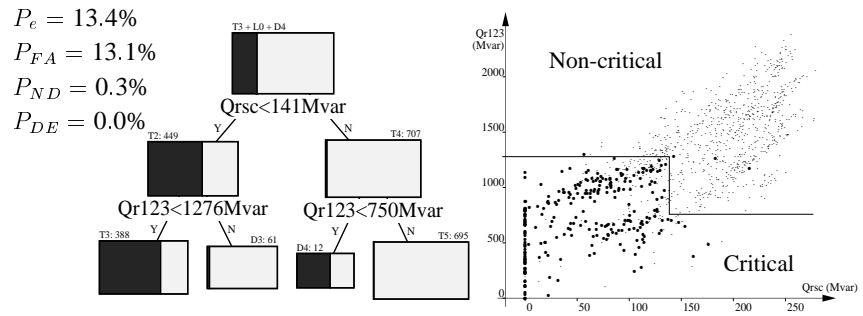


Figure 14.11 Simplified two level tree structure and its security regions

exploiting the two main attributes selected by the tree of Fig. 14.9a. Various techniques were compared to determine appropriate threshold values, in order to reduce as much as possible the number of non-detections, while minimizing the false alarms.

The particular combination of thresholds shown in Fig. 14.11 was determined by an iterative minimization of the mean non-detection cost (determined in the learning set). In order to force the method to give priority to the reduction of non-detection errors, various compromises were tested. The one corresponding to a non-detection cost 10 times higher for the insecure states than for the secure states was considered to be appropriate. The thresholds were adjusted so as to maximize the non-detection cost, by using a cyclic unidimensional search procedure, adapting each threshold in turn, until a local maximum was reached. Although, this technique was appropriate thanks to the small number of thresholds used in this problem, its application to optimize more complex trees, with a larger number of thresholds, would need a more sophisticated search algorithm (e.g. the genetic algorithm described in §3.5.3).

In the right-hand part of Fig. 14.11 we show the security region determined by the tree together with the 1156 independent test states. The overall error rate of the tree is of 13.4%, which corresponds to $P_{FA} = 13.1\%$, $P_{ND} = 0.3\%$ and $P_{DE} = 0.0\%$. Notice that the non-detections correspond to 4 states whose margin belongs to $[150 \dots 300] MW$. Again, it is interesting to compare these figures with the performance of the presently used criterion, in Table 14.4 : for a similar overall error rate, the modified decision tree allows, in spite of its simplicity, to virtually detect all critical situations, whereas the EHV voltage criterion leads to a non-detection rate of 7.1% including 0.5% of dangerous errors.

The above tree has also been tested on the JAD states corresponding to $\tau = 80s$, leading to an error rate $P_e = 21.0\%$, out of which 0.2% (2 states) are non-detections. This confirms the monotonic behavior of the reactive reserve attributes, and suggests various possible compromises. On the one hand, using rather high thresholds as in Fig. 14.11 will lead to an early detection of the critical situations and the higher the thresholds,

Table 14.6 *K – NN results for disturbance 1*

K	1	3	5	7	9	11	13	15	17	19
$P_e\%$	8.13	7.70	8.91	8.30	7.96	7.61	7.61	7.70	7.79	8.04

the higher the margins with respect to a possible fast load build up. On the other hand, using lower thresholds will allow us to reduce the non-detection rates at the expense of a less anticipative detection.

The preceding discussions show the multitudinous potential uses of the decision tree approach. The latter is indeed able to derive in a very flexible manner security criteria of appropriate characteristics. The quantitative comparison of the derived criteria with those presently in use, shows also that on the basis of a rather small data base it is nevertheless possible to derive interesting and useful security criteria, in spite of the complexity of the considered phenomena. In particular, we note that the decision trees told us to use reactive reserve attributes in order to define robust and efficient emergency state detection criteria.

14.4.4 Hybrid approaches

In addition to the preceding simulations various other learning methods have also been applied to the above data base, in the context of the same disturbance and classification.

Nearest neighbor

Table 14.6 reproduces results of the nearest neighbor method obtained using the 9 attributes selected by the tree of Fig. 14.9a, appropriately pre-whitened. The results suggest that the proper choice of a distance in the tree test attribute space might allow us to improve the classification performances of the trees. They show also that the “optimal” value of K lies around 13.

Multilayer perceptron

A more in-depth investigation was carried out to evaluate the capabilities of the multilayer perceptrons, and in particular of the hybrid DT-ANN approach described in chapter 6. In these simulations, reported in [WE92a], various hybrid and standard multilayer perceptrons were built. In Table 14.7 we report the main information leading to important conclusions. The columns of the table indicate the type of MLP approach, the attributes used as input variables, the type of structure (number of neurons in successive layers), the type of output information provided at the learning stage and finally the error rate obtained in the independent test set, when using the MLP to

Table 14.7 *Multilayer perceptrons built for disturbance 1*

Type of MLP	Attributes	Structure	Type of info.	P_e %
Direct	154 candidate	154-5-2	Classes	5.6
Hybrid	9 test attributes	9-10-11-2	Classes	8.2
Hybrid	9 test attributes	9-10-11-1	Margin	6.8
DT	9 test attributes		Classes	8.5

predict the classification of unknown states. For the ease of comparison we recall also the characteristics of the corresponding decision tree.

First of all, we notice the significant improvement of the error rates obtained by the MLPs using all 154 candidate attributes. This observation supports our previous impression that emergency voltage insecurity detection information is diffused among many different attributes.

Concerning the hybrid MLP, we observe that while without exploiting a security margin it hardly improves the accuracy of its corresponding tree, it improves it significantly when exploiting the post-contingency security margin during the learning stage. Noting again that in the present study the margins were determined rather coarsely, mainly to reduce computation times, it is expected that the hybrid approach should perform better on the basis of the richer data bases discussed below.

14.5 MULTICONTINGENCY STUDY

In this section we will briefly deal with the next generation data base, constructed so as to improve some of the shortcomings noted above. We describe first the main modifications made in the software and models used and then we comment on the range of situations and contingencies which have been screened.

14.5.1 Data base generation adaptations

One of the objectives of the present research collaboration is to exploit the preceding experience so as to specify and implement a flexible prototype data base generation software, in particular able to generate systematically large-scale data bases corresponding to a large number of operating states and contingencies.

A second requirement was to evaluate the feasibility of using a more elaborate power system model, representing in detail the HV subtransmission system in the study region. Another important improvement concerned the development of a more reliable voltage stability criterion and a more elaborate load power margin computation, allowing us in

particular to minimize discretization and other computation errors [VA 93b].

Finally, in order to take into account existing uncertainties about the load behavior, it was deemed necessary to randomize the steady state distribution of the load and compensation levels, as well as the sensitivity coefficients of the active and reactive load power to the MV voltage variations. From a practical point of view this should lead to more robust voltage security criteria, in particular for emergency state detection. From a methodological point of view it will allow us to illustrate and assess the ability of the machine learning approach to account for the effect of modelling uncertainties.

In addition to the above main modifications, some adaptations have also been made for the organization of the data base generation. In particular, on the basis of the experience acquired in the Hydro-Québec project (see Fig. 13.17, §13.4), it was decided to trace the random variants generated in an a priori data base, so as to enable the analysis of possible causes of load flow divergence problems, which may become a practical obstacle to the generation of representative data bases.

According to the above objectives a new data base generation software was developed and applied to evaluate voltage stability of the Brittany region, in a very broad multicontingency study, considering in parallel preventive security assessment and emergency state detection. In the next section we will describe the latter data base briefly and illustrate some of its information.

14.5.2 Summary of generated data bases

In order to make the study more easily accessible and appealing for the power system engineers in charge of the operation of the Brittany system, it was decided to take into account their expertise from the beginning. Thus, the scope of the data base generation, the disturbances and the candidate attributes, were decided in collaboration with the operators [WE 93g].

Random generation specifications

With respect to the preceding data bases the main concern was to generate a more diverse set of situations, while at the same time increasing the representativity of important classes of configurations.

In terms of topology, this led us in particular to determine a more adapted set of simple and double line pre-disturbance outages, taking into account information provided by the expert on “interesting” classes of topology. Further, the “radial” operation under high load conditions of the 225kV system, as well as changes in substation configurations were taken into account.

From the load level point of view the main changes consisted in randomizing the proportion of MV load at individual HV busses, around their usual values as well as their power factors.

On the other hand, major modifications were made concerning the active power generation schedule, including, in particular, situations where the generation units may operate at intermediate or low active power levels. In addition, 50% of the states were generated with gas turbines in operation in the pre-disturbance situation, so as to evaluate their quantitative impact on voltage security limits. Similarly, the possibility of having one or two of the region's synchronous condensers out of operation was considered.

A total number of 5000 normal pre-disturbance states were thus generated. Taking into account load flow divergence and random sampling specifications, a total number of 13513 variants were required. Further analyses were carried out on the a priori data base, so as to determine the physical or algorithmic reasons for this high percentage (62%) of divergences.

Disturbances

The interest in carrying out systematic multicontingency studies was shown in the transient stability context of the preceding chapter. To enable similar investigations, a rather broad set of 26 different disturbances was considered in the present study.

They correspond to the following types of contingencies.

Generation unit trippings, of 1, 2 or all units in operation in a regional power plant, or among the synchronous condensers (9 disturbances).

Busbar faults, in any of 5 important EHV substations of the study region.

Line trippings, of one or two circuits of 6 important EHV lines.

For each of the 26 disturbances, and for each of the 5000 pre-disturbance states a mid-term voltage security simulation of about 5 minutes was carried out, and for the stable scenarios a post-disturbance load power margin was determined, leading to 140,000 simulations and about 115,000 margin computations. The model used for these simulations has been specified in §12.1.1; it is described in detail in [JA 93].

Attributes

In terms of candidate attributes the experts proposed some key variables which are often used to monitor the system state, such as important EHV power flows, some representative 400kV voltages, the number of units in operation in plant 1 (see Fig.

14.5), the total load demand and the reactive shunt compensation reserve in the study region.

In addition to these, a set of complementary attributes were also included, such as the reactive generation reserves and additional power flows and voltages, as well as topological indicators.

Other variables, like HV voltages, various (EHV, zonal) active and reactive load levels, and EHV transformer power flows were also computed. In addition, the pre-disturbance load power margin was computed for each one of the 5000 states, providing a contingency independent security index.

Generated data bases

All in all, 28 data bases were constructed, containing information about (i) the 5000 pre-disturbance states; (ii) the 26×5000 JAD states; (iii) the pre-disturbance, and 26 post-disturbance margins and security classifications. 460 Mbytes of data were thus generated, stored in readily accessible ASCII files and made accessible to the analysis on the basis of the statistical and graphical tools developed within the TDIDT software. In particular, each one of the 5000 states may be selected and disturbances be resimulated in a very flexible and efficient way, and key parameters may be analyzed at variable time instants.

Also, histograms, scatter plots and correlation analyses may be systematically generated so as to appraise the multitudinous information contained in these data bases. This is illustrated below, where we merely show without discussion some of the security margin types of information.

14.5.3 Illustrations of load power margins

To illustrate the improvement of the security information available we have displayed in Fig. 14.12 the value of the pre-disturbance load power margin and its value in the post-disturbance state of disturbance 1, studied above. Comparing these histograms with those shown in Fig. 14.8 suggests that the new margin determination provides more precise information, in particular because of the *continuous* spectrum of its values. It is expected that this information may be exploited in various ways to improve the reliability of derived security criteria.

To suggest a possible interesting use of these margins we have represented in Fig. 14.13 two scatter plots which show the correlation between the pre-disturbance load power margin and the post-disturbance load power margins for two different disturbances.

On the one hand, the scatter plot of Fig. 14.13a illustrates a quite mild disturbance,

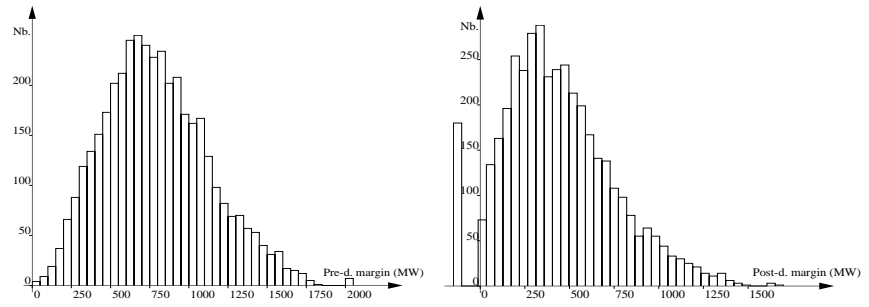


Figure 14.12 Pre- and post-disturbance active load power margin distributions (the relevant 4041 states for disturbance number 1)

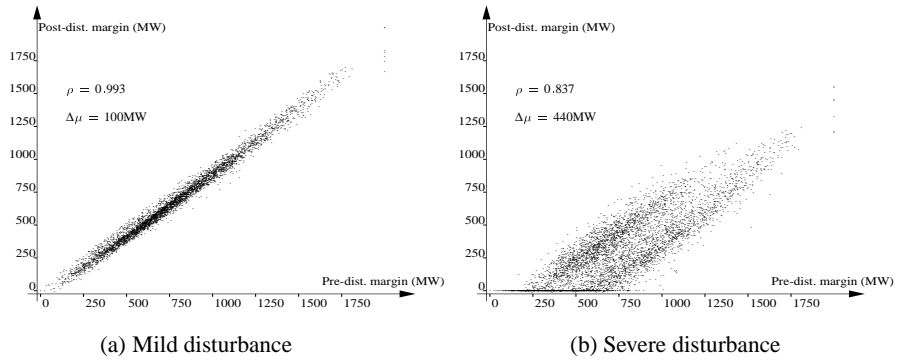


Figure 14.13 Correlation of pre- and post-disturbance load power margins

corresponding to the tripping of one synchronous condenser. Of course, this scatter plot considers only the 4527 relevant situations with respect to the contingency of concern. They correspond to the states, among the 5000 states of the data base, where there is actually a synchronous condenser in operation. As indicated in Fig. 14.13a, the mean difference between the pre-disturbance and post-disturbance margins is of about 100MW. Further, as we see from the scatter plot and from the high correlation coefficient $\rho = 0.993^3$, the post-disturbance margin is strongly related to the pre-disturbance one.

On the other hand, the scatter plot of Fig. 14.13b illustrates a much severer disturbance, corresponding to the busbar fault in substation 1. Here, the mean difference between the pre-disturbance and post-disturbance margins is about 440MW. Further, the relatively low correlation coefficient ($\rho = 0.837$) suggests that other factors influence the value of this post-disturbance load power margin. This is also confirmed by the diffuse and multi-modal shape of the scatter plot. In fact, the severity of this disturbance will

³ ρ denotes the correlation coefficient of the pre-disturbance and post-disturbance margins, computed according to eqn. (2.11) on the basis of the relevant data base states.

depend for example on the number of EHV lines connected to the faulted busbar, which varies from one operating state to another.

14.6 FUTURE PERSPECTIVES

During the last four years several breakthroughs have been made in the context of machine learning approaches, as well as in terms of reliable and efficient stability simulation software and hardware and in terms of methodologies and software for data base generation and management.

Both voltage security and transient stability applications of the machine learning framework will benefit in the future from these advances, and will hopefully lead to practical implementation. The preceding two chapters aimed among other things at suggesting the practicality of this approach, by providing in-depth discussions of past and future research projects.

In the context of voltage security, we believe that the data base generation has by now reached maturity, and the existing tools may be considered as prototypes of future practical software packages. In particular, we believe that the main considerations, relating to the choice of independent parameters for random sampling, and to the randomization of the unknown “hidden” parameters such as load modelling, have been solved. On the other hand, the generation of large enough data bases and the simulation of the relevant disturbances, rely strongly on the existence of reliable and at the same time efficient simulation techniques, which have been developed in parallel with the present research project. In particular, the validation of the security criteria and of the margin computations could be advantageously achieved on the basis of the very large diversity of situations contained in our data bases. Finally, the used power system model is certainly sufficiently detailed for making realistic voltage security studies.

In short, the feasibility of generating very large data bases for voltage security studies has been demonstrated. On the other hand, the potential of decision trees was shown in the preceding studies.

In our future research on voltage security, we will first come back to our early concern of preventive wise assessment so as to appraise security information in terms of usual prefault operating parameters. This will allow us to compare this information with existing expertise and gain further confidence. In this context, we believe that the proper exploitation of load power margins may provide very rich and powerful security criteria [WE94c], and allow multitudinous multicontingency analyses.

The next research stage will then be to compare systematically the criteria obtained in the context of emergency state detection with the preventive mode. A main issue will be to determine how far variable time delays and uncertain load behavior may lead

to a fuzzification of the emergency state security boundaries. These considerations should be taken into account in the modelling process, so as to avoid overestimating the derived criteria. Let us therefore recall that the possibility of taking such modelling uncertainties into account while designing the security criteria is a very unique feature of our machine learning approach.

15

Conclusions

In this thesis we have attempted to survey potentials of *machine learning approaches* for *power system security assessment*.

In the first part we have described machine learning and related statistical and neural network methods. Our purpose was to provide insight into the possible complementary uses of these various methods. We have therefore put the emphasis on illustrations and discussions of issues related to their practical use, rather than on extensive theoretical presentations already available in the specialized literature.

Being driven by the requirements of security problems more than by the features of a particular subclass of learning methods, we have provided in the second part a synthetic discussion of security problems and of the computer based learning framework to solve them. Here also, we have purposely avoided the restatement of information about modelling and simulation techniques already available in the power system literature.

To render credible machine learning approaches to power system security and to appraise the current advancement of research, we have reported in the third part our results obtained with extensive experimentations. Although the most interesting results correspond to the real large-scale system applications, we found it interesting to recall our early attempts with academic type systems. This has illustrated the successive phases of research which have gradually led us to formulate the methodology.

One of the messages of this thesis is that to make learning methods really successful it is important to include the human expert in the process of deriving security information. For example, to guide the security studies it is necessary to exploit his prior expertise and then to allow him to criticize, assimilate and accept the new information. The results must therefore be provided in a form compatible with his own way of thinking. In the general class of computer based learning approaches, the machine learning approaches are presently the only ones able to meet this requirement. They are therefore a key element in our framework.

Clearly, machine learning as well as other learning methods can produce interesting security information only when they exploit representative data bases. The data base generation approaches that we have used, discussed and illustrated in our research essentially rely on a pragmatic trial and error procedure. We believe that this methodology has reached some maturity and we note that while the initial investment, when applying it to a new security problem is quite important, the subsequent data base generation takes full advantage of the previous ones.

At the present stage of development, we believe that the credibility and the practical feasibility of the proposed approach and its usefulness have already been shown. There are however some aspects calling for additional research.

From the methodological viewpoint, there is a need for more systematic ways to control the “false alarm vs non-detection” compromise of the derived security criteria, so as to meet the different requirements of planning, normal operation and emergency control. Some promising approaches have been identified and explored but need further developments. In particular, we mention a decision tree threshold shifting algorithm and hybrid DT-ANN or DT-NN techniques, which allow exploitation of the information contained in security margins.

In the context of data base generation, on the other hand, parallel simulation environments should be developed to exploit available computing powers, by enabling a transparent allocation of simulations on virtual machines composed of large numbers of elementary workstations connected by local or wide area networks.

Such computing environments would allow us to progress further in terms of practical validations and assessments of the methodology within various security contexts. For example, various compromises could be studied between very broad long-term studies covering many different system configurations, and the determination of security limits for a more restricted range of situations one day or one hour ahead.

After eight years of research, we deem that machine learning methods are indeed able to provide interesting security information for various physical problems and practical contexts. Actually, in their philosophy they are quite similar to existing practices in power system security studies, where limits are derived from simulations, though in a manual fashion. But machine learning approaches are more systematic, easier to handle and master, in short more reliable and powerful.

Meanwhile, available computing powers have increased sufficiently to run with acceptable response times the large amounts of simulation required by statistical machine learning methods. We can go even further, by stating that the very rapidly growing computing powers can no longer be satisfactorily exploited via manual approaches used traditionally in security studies. In this respect, the presented methodology provides a fully flexible way to exploit systematically parallelism. As we have indicated, with presently or soon to be available computing environments, it indeed becomes possible

to run hundreds of thousands of realistic security simulations within response times as small as some days to some weeks.

These possibilities open up new perspectives to power system engineers to respond to the challenge of planning and operating future power systems with an acceptable level of security, in spite of increasing uncertainties (e.g. due to the deregulation of transmission systems and fast technological changes) and increasing economical and environmental pressures.

Appendix - Uncertainty measures

A.1 MOTIVATION

The objective of this appendix is to provide a deeper insight into the uncertainty or information criteria used in the context of decision tree induction, and more generally of learning conditional class probability models. Our intention is to show the high degree of similarity among two main families of criteria based respectively on the logarithmic *SHANNON* entropy function and the quadratic *GINI* index.

We start by introducing a general family of entropy functions and then discuss some of the interesting particular cases mentioned in chapter 2 or 3.

A.2 GENERALIZED INFORMATION FUNCTIONS

The concept of generalized information functions of type β was first introduced by Daróczy [DA 70] and its use for pattern recognition problems was discussed by Devijver [DE 76].

The entropy of type β (β positive and different from 1) of a discrete probability distribution (p_1, \dots, p_m) is defined by

$$H^\beta(p_1, \dots, p_m) \triangleq \sum_{i=1}^m p_i u^\beta(p_i), \quad (\text{A.1})$$

where $u^\beta(p_i)$ denotes the uncertainty measure (of type β) of class c_i and is defined by

$$u^\beta(p_i) \triangleq \frac{2^{\beta-1}}{2^{\beta-1} - 1} (1 - p_i^{\beta-1}). \quad (\text{A.2})$$

The uncertainty measure u^β is a strictly decreasing function of p_i .

A.2.1 Properties of H^β

H^β satisfies the following properties [DA 70, DE 76, WE 90a].

1. $H^\beta(p_1, \dots, p_m) = \frac{2^{\beta-1}}{2^{\beta-1}-1} \left[1 - \sum_{i=1}^m p_i^\beta \right]$;
2. $H^\beta(p_1, \dots, p_m)$ is invariant w.r.t. the permutation of its arguments;
3. $H^\beta(p_1, \dots, p_m) = H^\beta(p_1, \dots, p_m, 0)$;
4. $H^\beta(1) = H^\beta(0, \dots, 0, 1, 0, \dots, 0) = 0$ and $H^\beta(\frac{1}{2}, \frac{1}{2}) = 1$;
5. $H^\beta(p_1, \dots, p_{m-1}, p_m) = H^\beta(p_1, \dots, p_{m-1} + p_m) + (p_{m-1} + p_m)^\beta H^\beta(p_{m-1}/(p_{m-1} + p_m), p_m/(p_{m-1} + p_m))$ (pseudo-additivity);
6. $0 \leq H^\beta(p_1, \dots, p_m) \leq H^\beta(\frac{1}{m}, \dots, \frac{1}{m})$, i.e. the maximal expected uncertainty corresponds to uniform distribution;
7. $H^\beta(p_1, \dots, p_m)$ is a concave (\cap) function on the convex set of probability distributions, defined by the constraints $p_i \geq 0$ et $\sum_{i=1}^m p_i = 1$:

$$\begin{aligned} & \forall \lambda_j \geq 0, p_{ij} \geq 0, i = 1, \dots, m; j = 1, \dots, k, | \\ & \sum_{j=1}^k \lambda_j = 1 \text{ et } \forall j : \sum_{i=1}^m p_{ij} = 1 : \\ & H^\beta\left(\sum_{j=1}^k \lambda_j p_{1j}, \dots, \sum_{j=1}^k \lambda_j p_{mj}\right) \geq \sum_{j=1}^k \lambda_j H^\beta(p_{1j}, \dots, p_{mj}). \end{aligned}$$

The interested reader may refer to [WE 90a] for the proofs of the above properties, not given here to save space.

Daróczy shows that properties 2, 4 and 5 provide a characterization of the entropy functions of type β . In particular, if we impose simple *additivity* of entropies of independent variables, or equivalently

$$\begin{aligned} H^\beta(p_1, \dots, p_{m-1}, p_m) &= H^\beta(p_1, \dots, p_{m-1} + p_m) + \\ & (p_{m-1} + p_m) H^\beta\left(\frac{p_{m-1}}{(p_{m-1} + p_m)}, \frac{p_m}{(p_{m-1} + p_m)}\right), \end{aligned}$$

it is necessary to let β converge towards 1, yielding the classical logarithmic entropy used in thermodynamics and information theory. This is further discussed below.

To fix ideas about the effect of β on the shape of the entropy functions we have reproduced in Fig. A.1 the graphs of these functions, in the two-class case ($p_1 = p; p_2 = 1 - p$), for various values of β . In particular, it is interesting to notice the relatively small difference between the logarithmic ($\beta \rightarrow 1$) and the β type entropies, for $\beta \in]1 \dots 3]$. Thus, considering these curves the logarithmic and the quadratic entropies discussed further below appear to be quite similar.

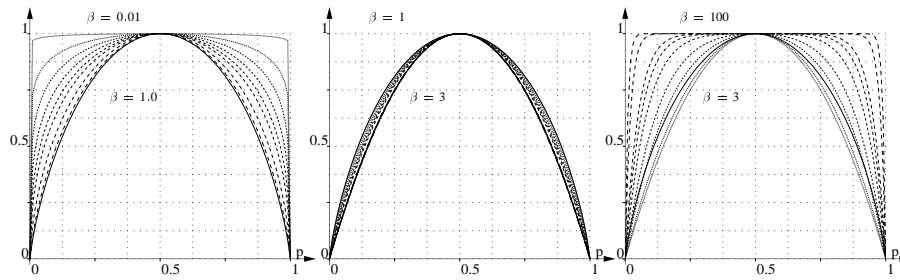


Figure A.1 Entropy functions for $\beta \in [0.01 \dots 100.0]$

A.2.2 Conditional entropies

Let t and c denote two discrete random variables (e.g. a test at a tree node, and a classification) of respective probability distribution $(p(t_1), \dots, p(t_k))$ and $(p(c_1), \dots, p(c_m))$. We denote by

$$H_C^\beta \triangleq H^\beta(p(c_1), \dots, p(c_m)), \tag{A.3}$$

the prior classification entropy of type β and the conditional type β entropy is defined by

$$H_{C|t_j}^\beta \triangleq H_C^\beta(p(c_1 | t_j), \dots, p(c_m | t_j)), \tag{A.4}$$

and the mean conditional type β entropy by

$$H_{C|T}^\beta \triangleq \sum_{j=1}^k p(t_j) H_{C|t_j}^\beta. \tag{A.5}$$

The concave nature of H^β implies the following fundamental monotonicity property (see [WE 90a])

$$H_{C|T}^\beta \leq H_C^\beta. \tag{A.6}$$

Furthermore, due to the strictness of the concavity the following equality holds true

$$H_{C|T}^\beta = H_C^\beta \Leftrightarrow p(c_i | t_j) = p(c_i), \quad \forall i, j; \tag{A.7}$$

i.e. if and only if the class variable c and t are statistically *independent*.

The conditional entropy is a measure of the mean residual uncertainty of the classes, given full information about the random variable t .

A.3 SHANNON ENTROPY

For $\beta = 1$ the above uncertainty measure is not defined anymore, but in the limit, for $\beta \rightarrow 1$, we obtain the logarithmic or SHANNON entropy defined in chapter 2.

$$H \triangleq \lim_{\beta \rightarrow 1} H^\beta = - \sum_{i=1}^m p_i \log_2 p_i, \quad (\text{A.8})$$

where by continuity we take $0 \log_2 0 = 0$.

It may be easily checked that the properties of the β type entropies hold also for the logarithmic entropy function. A fundamental property of this entropy function is its additivity, expressing the fact that the uncertainty of two independent events is equal to the sum of their respective uncertainties. In the context of probabilistic modelling this leads to an interesting interpretation of the information provided by a model in terms of the posterior likelihood of this model [WE 90a, RI 91, WE 94a]. It is not our intention to discuss these interpretations here, but we merely note that they are certainly among the main reasons of the high popularity of this particular uncertainty measure [GU 93].

A.3.1 Conditional entropies and information

The mean conditional entropy becomes the following

$$H_{C|T} = - \sum_{j=1}^k \sum_{i=1}^m p(c_i, t_j) \log_2 p(c_i | t_j). \quad (\text{A.9})$$

The following quantities of interest are also defined.

- The entropy of t ,

$$H_T = - \sum_{j=1}^k p(t_j) \log_2 p(t_j) \quad (\text{A.10})$$

- The mean conditional entropy of t given c

$$H_{T|C} = - \sum_{i=1}^m \sum_{j=1}^k p(c_i, t_j) \log_2 p(t_j | c_i). \quad (\text{A.11})$$

- The joint entropy of t and c

$$H_{C,T} = - \sum_{i=1}^m \sum_{j=1}^k p(c_i, t_j) \log_2 p(c_i, t_j). \quad (\text{A.12})$$

- The mutual informations

$$I_C^T \triangleq H_C - H_{C|T}, \quad (\text{A.13})$$

$$= - \sum_{i=1}^m \sum_{j=1}^k p(c_i, t_j) \log_2 \frac{p(c_i)}{p(c_i|t_j)}, \quad (\text{A.14})$$

$$I_T^C \triangleq H_T - H_{T|C}, \quad (\text{A.15})$$

$$= - \sum_{i=1}^m \sum_{j=1}^k p(c_i, t_j) \log_2 \frac{p(t_j)}{p(t_j|c_i)}. \quad (\text{A.16})$$

The following relationships are satisfied.

- Additivity of entropies

$$H_{C,T} = H_C + H_{T|C} = H_T + H_{C|T} = H_{T,C}. \quad (\text{A.17})$$

- And consequently reciprocity of the mutual information

$$I_C^T = H_C - H_{C|T} = H_T + H_C - H_{T,C} = H_T - H_{T|C} = I_T^C. \quad (\text{A.18})$$

- Thus,

$$I_T^C = - \sum_{i=1}^m \sum_{j=1}^k p(c_i, t_j) \log_2 \frac{p(c_i)p(t_j)}{p(c_i, t_j)}. \quad (\text{A.19})$$

- Inequalities

$$H_{T|C} \leq H_T ; H_{C|T} \leq H_C ; I_C^T \leq H_C ; I_T^C \leq H_T ; I_C^T \leq H_{C,T} ; I_C^T \geq 0. \quad (\text{A.20})$$

Further, under the necessary and sufficient condition of strict association between t and c (i.e. $p(c_i, t_j)$ diagonalized by permutation of columns or lines) the following equalities hold.

$$I_C^T = H_T = H_C = H_{C,T} ; H_{T|C} = H_{C|T} = 0. \quad (\text{A.21})$$

Finally, under the necessary and sufficient condition of statistical independence the following equalities hold.

$$H_T = H_{T|C} ; H_C = H_{C|T} ; H_{C,T} = H_C + H_T ; I_C^T = 0. \quad (\text{A.22})$$

A.3.2 Normalizations

The information I_C^T measures the reduction of the uncertainty of one of the variables t or c , given the knowledge of the other one. In the context of decision tree induction it is useful as an evaluation function of alternative tests at a tree node, in order to select the one reducing most significantly the uncertainty about the unknown classification. More generally, in the context of statistical modelling this measure may be used to assess the information provided by alternative models, e.g. alternative sets of parameters of a neural network.

Within this context, the fact that the information quantity is upper bounded by the prior entropy H_C renders the interpretation of its values difficult. The upper bound, and thus the observed values of candidate models may indeed be highly variable according to the number and distribution of classes.

Another frequently mentioned difficulty in the context of decision tree induction concerns the bias of the information quantity which tends to favor tests at a tree node with a larger number of outcomes [QU 86b, WE 90a, LO 91].

To provide an improved “score” measure, various ways of normalizing the information quantity have thus been proposed in the literature [QU 86b, KV 87, MI 89b, LO 91]. We will present some of them briefly below and provide an illustration on the basis of data related to our transient stability example.

Normalization by H_C

We denote this score measure by

$$A_C^T \triangleq \frac{I_C^T}{H_C}. \quad (\text{A.23})$$

In the context of decision tree building, at a given tree node H_C is constant. Thus the ranking provided by A_C^T and I_C^T are equivalent and the normalization has no effect at all on the resulting tree. We have used it rather than I_C^T , merely for comparison purposes, its values being closer to the values obtained by the other three measures described below.

It is worth mentioning that I_C^T and consequently A_C^T presents at least two interesting properties which do not hold necessarily for the other measures presented below.

The first property concerns the location of optimal thresholds for ordered attributes. One may indeed show that for ordered attributes, the optimal thresholds maximizing I_C^T must lie at so-called cut-points, i.e. values where the class probabilities are not stationary. (In the finite sample case, this excludes in particular all thresholds lying

between states of identical classes.) Exploiting this property allows in general to reduce significantly the computational burden of searching for the optimal thresholds.

The second property concerns the search for an optimal binary partition for a qualitative attribute [BR 84, CH 91]. It allows to reduce the search from $2^{L-1} - 1$ to L candidate partitions (where L denotes the number of different values assumed by the qualitative attribute).

Normalization by H_T

In order to reduce the bias towards many-valued splits, Quinlan introduced the so-called “gain ratio”, which we denote by

$$B_C^T \triangleq \frac{I_C^T}{H_T}. \quad (\text{A.24})$$

The division by H_T allows of course to compensate the higher bias of I_C^T for tests with a higher number of successors, which correspond generally to a higher value of H_T .

However, a possible problem with this measure lies in the fact that it may overestimate the value of splits with very low H_T values, in particular splits corresponding to uneven decompositions of a learning set into subsets. Thus, for ordered attributes the optimal values of B_C^T often tend to be located closer to its extreme values; this is known in the literature as the “end-cut” preference of the “gain ratio” criterion.

Normalization by $\frac{1}{2}(H_C + H_T)$

The preceding normalizations yield asymmetrical “score” measures. While it has been suggested that asymmetrical measures are natural in the context of pattern recognition applications, because the learning objective privileges the classification variable [DE 76], we believe that symmetrical measures are more appropriate. Indeed, in the context of decision tree building a main objective is interpretation of correlation among attributes and classifications, and also among various attributes. There is no reason that the correlation of two attributes should depend on their order.

Thus, sharing the opinion of Kvålseth [KV 87], we preferred to use the following measure [WE 89b].

$$C_C^T \triangleq \frac{2I_C^T}{H_C + H_T}, \quad (\text{A.25})$$

which is symmetrical in C and T .

Kvålseth shows that if $I_C^T > 0$, the sampling estimate \hat{C}_C^T is asymptotically normally distributed with mean C_C^T and thus is unbiased. One of its main practical advantages is that Kvålseth provides an explicit formulation of its variance (see eqn. (3.21)).

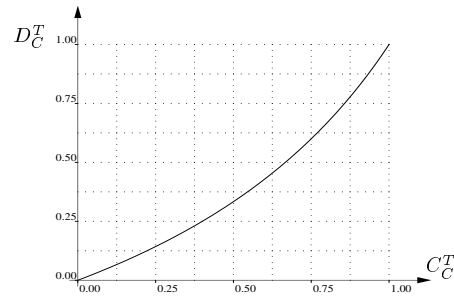


Figure A.2 Relationship between D_C^T and C_C^T

This allows one to appraise the uncertainty of the sample estimate of the uncertainty measure, thus the significance of score differences among various candidate partitions may be assessed.

Normalization by $H_{C,T}$

Another symmetrical and normalized measure recently proposed by López de Mántaras is defined by [LO91]

$$D_C^T \triangleq \frac{I_C^T}{H_{C,T}}. \quad (\text{A.26})$$

This author shows formally that D_C^T is not biased towards many-valued splits, and suggests also that it tends to provide simpler trees than the gain ratio measure. He shows also that $1 - D_C^T$ is a proper distance measure of two probability distributions $(p(c_1), \dots, p(c_m))$ and $(p(t_1), \dots, p(t_k))$, which satisfies the triangular inequality.

Let us show the equivalence of the last two measures C_C^T and D_C^T .

Noting that $H_{C,T} = H_C + H_T - I_C^T$ we find that

$$D_C^T = \frac{I_C^T}{H_C + H_T - I_C^T}, \quad (\text{A.27})$$

or equivalently that

$$D_C^T = \frac{1}{\frac{H_C + H_T}{I_C^T} - 1} \quad (\text{A.28})$$

Thus

$$D_C^T = \frac{1}{\frac{2}{C_C^T} - 1} \quad (\text{A.29})$$

and the two measures are a monotonic transformation of each other, as shown in Fig. A.2. Thus the two measures are equivalent as far as the *ranking* of candidate tests is concerned and the formal property of no bias towards multiple-valued splits of D_C^T holds also for C_C^T .

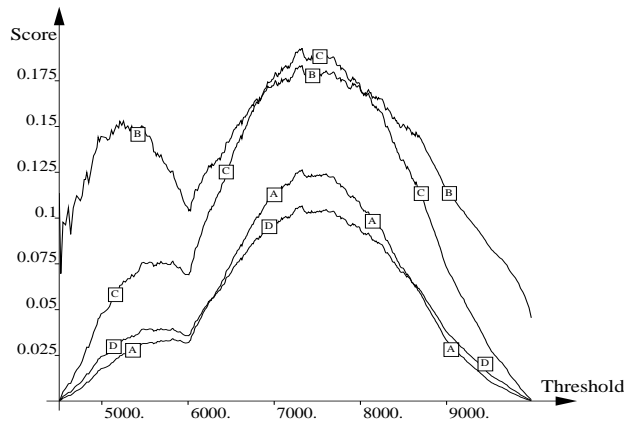


Figure A.3 Variation of various scores for test $TRBJ < THRESHOLD$

Comparison

First of all we recall that in the context of experimental studies the predictive classification reliability of decision trees appears to be not much affected by the type of attribute selection criteria used. We noted this when reporting the results of the Statlog project, and this was observed by many researchers [BR 84, MI 89b, LO 91] including ourselves.

However, the complexity of the trees and hence their interpretability which is one of their main attractive features, does depend much more strongly on the type of measure used. Further, since the complexity of the tree will influence the size of the learning samples at the terminal nodes, it will influence strongly the accuracy of their class probability estimates. Information about the tree complexity is however not so often reported in experimental studies and the value of simplicity may depend on pragmatic considerations which are difficult to take into account in systematic comparisons.

For the purpose of our illustration, Fig. A.3 represents the variation of the above four measures as a function of the test threshold for the problem described in §3.4.3. To minimize the effect of sampling, we have determined the scores on the basis of the complete data base of 12497 states.

From the observation of these curves we make the following comments. First of all, all four measures present two salient local maxima, one below 6000MW and one around 7300MW, which is also the global maximum. Actually, this translates the two different statistical populations from which the data base samples were drawn [WE 93c]. In addition to these dominant tendencies, there are small high frequency oscillations translating the effect of the sampling of the probability distributions of classes. They vanish however above 8700MW, where all four curves start decreasing monotonically.

This is merely the consequence of the fact that above this threshold value all the states of the data base belong to the same class, as is confirmed by Fig. 3.9.

Comparing the curve related to measure A with the three others, we observe that the normalization of B, C and D taking into account H_T , enhances indeed the scores nearby the upper and lower bound of the threshold interval. In particular, the value of the local maximum nearby 5700MW is enhanced, and pulled towards the lower values. This effect is stronger for measure B_C^T then for measures C_C^T and D_C^T . Incidentally, we note that the latter two measures are indeed equivalent, in terms of the location of all the local maxima of their curve.

Finally, we may observe in this present example the odd behavior of measure B_C^T near the extreme values of the threshold interval, where $H_T \approx 0$. In particular its limit value is not equal to zero.

A.3.3 Hypothesis testing

We merely recall the already mentioned fact that under the hypothesis of statistical independence the finite sample estimate $2N \ln 2 \hat{I}_C^T$ is distributed according to a χ -square law of $(m-1) \times (k-1)$ degrees of freedom [KV 87].

Thus the expected value of \hat{I}_C^T will assume the following value

$$E\{\hat{I}_C^T\} = \frac{(m-1)(k-1)}{2N \ln 2}. \quad (\text{A.30})$$

This confirms¹ the fact that I_C^T is biased, and the higher the number of successors and classes, the higher the bias. On the other hand, the bias decreases towards zero when the sample size N increases.

A.4 QUADRATIC ENTROPY

The quadratic entropy is the β type entropy, for $\beta = 2$.

$$H^2 = 2 \left[1 - \sum_{i=1}^m p_i^2 \right] \quad (\text{A.31})$$

$$= 4 \sum_{i \neq j} p_i p_j \quad (\text{A.32})$$

$$= 2 \sum_{i=1}^m p_i (1 - p_i), \quad (\text{A.33})$$

¹strictly speaking only under the independence hypothesis

This is identical to the so-called ‘‘Gini’’ index [BR 84], which may be interpreted in the following way. Let us suppose that an object is classified randomly into c_i , with a probability equal to p_i , in order to mimic the observed random behavior of the classification. Then the probability of misclassifying the object will be equal to $1 - p(c_i)$ and the expected misclassification probability is

$$P_e = \sum_{i=1}^m p(c_i)(1 - p(c_i)) = \frac{H_C^2}{2}. \quad (\text{A.34})$$

Thus reducing the Gini index amounts to reducing the misclassification error associated with a randomized classification. The Gini index is also equal to the variance of the class-indicator regression variable (defined by $y_i(o) = 1$ if $c(o) = c_i$, and $y_i(o) = 0$ otherwise). Thus, reducing the Gini index consists also of reducing the residual variance of class indicator variables.

From the preceding discussion it follows also that the expected value of the quadratic entropy conditioned on the attribute values is identical to the asymptotic error rate of the nearest neighbor rule.

A.4.1 Conditional entropies and information

As above, the conditional quadratic classification entropy is defined by

$$H_{C|T}^2 \triangleq \sum_{j=1}^k p(t_j) H_{C|t_j}^2, \quad (\text{A.35})$$

$$= 1 - \sum_{i=1}^m \sum_{j=1}^k \frac{p^2(c_i, t_j)}{p(t_j)}, \quad (\text{A.36})$$

and the quadratic information provided by t on c is defined by

$$I_C^{2T} \triangleq H_C^2 - H_{C|T}^2. \quad (\text{A.37})$$

Similarly, one may define

$$H_{C|T}^2 \triangleq \sum_{i=1}^m p(c_i) H_{T|c_i}^2, \quad (\text{A.38})$$

$$= 1 - \sum_{i=1}^m \sum_{j=1}^k \frac{p^2(c_i, t_j)}{p(c_i)}, \quad (\text{A.39})$$

and the quadratic information provided by c on t is defined by

$$I_T^{2C} \triangleq H_T^2 - H_{T|C}^2. \quad (\text{A.40})$$

It is worth noting that in general $I_C^{2T} \neq I_T^{2C}$.

In the CART method, Breiman et al. use I_C^T as an attribute selection criterion [BR 84]. Given the very similar behavior of quadratic and logarithmic entropies, this criterion must admittedly suffer from similar difficulties than the logarithmic information criterion of §A.3.2. In particular, it suffers from bias towards many-valued splits and makes the comparison of scores for different values of the prior entropy difficult.

A.4.2 Normalizations

We are not surprised that the same normalization “medicine” has been applied to derive from the quadratic entropy an appropriate optimal splitting criterion. We will merely indicate the definition of the resulting *symmetrical* τ measure proposed by [ZH 91],

$$\tau \triangleq \frac{I_C^T + I_T^C}{H_T^2 + H_C^2}, \quad (\text{A.41})$$

which is the exact equivalent of our own C_C^T measure.

Of course the advantages of the latter measure are the same than those of C_C^T , no more no less.

A.4.3 Hypothesis testing

In the second part of their paper the authors of [ZH 91] present the use of an associated χ -square hypothesis test. They note indeed that the quantities

$$(N-1)(m-1) \frac{I_C^T}{H_C^2} \text{ or } (N-1)(k-1) \frac{I_T^C}{H_T^2} \quad (\text{A.42})$$

are distributed according to a χ -square law with $(m-1)(k-1)$ degrees of freedom.

A.5 OTHER LOSS AND DISTANCE FUNCTIONS

Many other criteria have been proposed and are used in various decision tree induction algorithms. Not all use an as uniform approach as the two preceding ones, exploiting the same measure for the selection and pruning criteria. A very interesting discussion of general divergence measures and their algorithmic properties is given in [CH 91]. Another approach to avoid overestimating the capabilities of multiple-valued tests (and thus also of over complex trees) consists of “deconvexifying” the used information measures by using modified estimates of relative frequencies such as

$$\hat{p}_i = \frac{n_i + \lambda}{n + m\lambda} \quad \forall i = 1, \dots, m. \quad (\text{A.43})$$

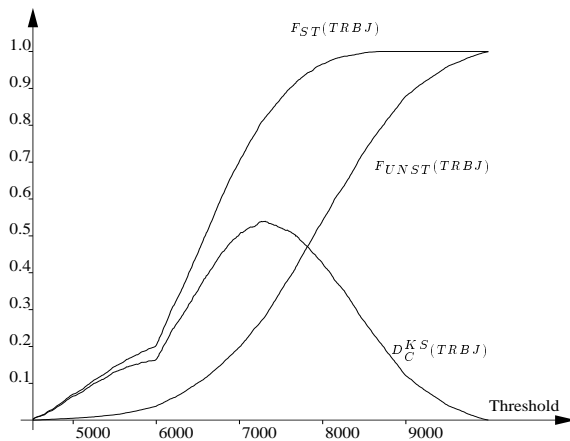


Figure A.4 Kolmogorov-Smirnoff distance as a function of $TRBJ$.

These have been proposed by various authors on the basis of various arguments and for various purposes [QU 87b, BU 90, ZI 92].

Below we will merely describe the Kolmogorov-Smirnoff criterion, which has been proposed very early by Friedman [FR 77] as an attribute selection criterion, and was extended afterwards by Rounds to provide also a stop-splitting criterion [RO 80], on the basis of a similar hypothesis testing approach than above.

A.5.1 Kolmogorov-Smirnoff distance

The basic method is restricted to the two-class case and to ordered (e.g. numerically continuous) attributes.

Let us denote by $F_{c_1}(a_i)$ the (cumulative) probability distribution of an attribute, conditioned to class c_1 and $F_{c_2}(a_i)$ the same distribution in class c_2 . The Kolmogorov-Smirnoff distance is defined as

$$D_C^{KS}(a_i^*) = \max_{a_i} |F_{c_1}(a_i) - F_{c_2}(a_i)|. \tag{A.44}$$

The sampling distribution of this quantity has been determined by Kolmogorov, under the hypothesis of independence, i.e. if the probability distribution of the attribute a_i is independent of its class. Its nice property is that it is independent of the distribution $F(a_i)$, and thus yields a non-parametric hypothesis test of the independence of a_i and c .

Note that the sampling distribution (and thus the levels of significance) depends on the sample sizes of each class which are however constant at a given tree node and

independent of the considered attribute. Thus the ranking of $D_C^{KS}(a^*)$ is equivalent to the ranking of the significance levels, and the optimal splitting rule derived by Friedman consists of splitting a node by the attribute a_* corresponding to the maximum Kolmogorov-Smirnoff distance,

$$D_C^{KS}(a_*) = \max_i D_C^{KS}(a_i^*), \quad (\text{A.45})$$

together with its optimal threshold a_*^* .

The stop-splitting rule associated with this criterion by Rounds consists merely of checking that the significance level $1 - \alpha$ corresponding to $D(a_*^*)$ is larger than a given a priori fixed threshold. It is a wonder to us that Friedman did not propose this rule himself.

To appraise this criterion, we have applied it to the same problem corresponding to Fig. A.3. The corresponding variation of the sample values of $F_{ST}(TRBJ)$, $F_{UNST}(TRBJ)$ and $D_C^{KS}(TRBJ)$ are illustrated in Fig. A.4.

We note that the overall shape of the D_C^{KS} curve is quite similar to the shape of the curves in Fig. A.3. It reaches its maximum value at 7310.5MW, which is very close to the 7308.5MW where the maximum of curves C_C^T and D_C^T of Fig. A.3 is located. It is interesting to observe that the behavior of D_C^{KS} is smoother than that of the latter two measures, which suggests that its optimum threshold may be less sensitive to sampling noise.

List of Figures

1.1	Machine learning framework for security assessment	1
1.2	Operating states and transitions. Adapted from [FI 78]	4
1.3	Hypothetical decision tree	17
1.4	A soft linear threshold unit	19
1.5	Feed forward multi-layer perceptron	19
3.1	Graphs, trees and directed trees	50
3.2	Example tree and attribute space representation	52
3.3	Partitioning of qualitative vs hierarchical attributes.	55
3.4	Illustration of overfitting	58
3.5	Characteristics of pruned trees for increasing β	61
3.6	Difficult examples for the standard TDIDT approach	64
3.7	Example of trellis structure resulting from node merging	66
3.8	One-line diagram of 735kV Hydro-Québec system	69
3.9	Empirical distribution of TRBJ : total James' Bay power flow	71
3.10	Variation of the score of the test $TRBJ < THRESHOLD$	79
3.11	Random variations of optimal thresholds and scores	80
3.12	Illustration of linear combination attribute	83

3.13	Quality variation : growing and pruning (adapted from [WE 93h]) . . .	86
3.14	Test set error of pruned trees and “1 standard error rule”	87
3.15	Pruning sequences for a transient stability assessment tree	88
3.16	Decision tree : $N = 8000$, $M = 2497$, $\alpha = 5 * 10^{-5}$, $P_e = 4.2\%$. . .	89
3.17	Illustration of crossover and mutation operators	99
4.1	Difference between Fisher and optimal linear discriminant	108
4.2	Nearest neighbor, editing and condensing	114
4.3	Nearest neighbor ambiguity and distance rejection	115
4.4	Graphical representation of the projection pursuit model	117
4.5	Various kernel functions and smoothing parameters	119
4.6	Example two-dimensional histograms	121
4.7	Classification corresponding to the histograms of Fig. 4.6	121
4.8	Hierarchical attribute clustering example	125
5.1	Basic linear threshold unit	135
5.2	Soft threshold unit for the linear combination of <i>TRBJ</i> and <i>NB_COMP</i> 138	
5.3	Comparison of various linear combinations	139
5.4	Variation of MSE during steepest descent iterations	140
5.5	Feed-forward multi-layer perceptron	142
5.6	General feed-forward network	143
5.7	Back-propagation of errors	143
5.8	Explanation of the chain rule differentiation	145
5.9	Convergence of the BFGS algorithm for the transient stability example	150
5.10	Convergence of the steepest descent algorithm	151
5.11	Abnormal extrapolations due to overfitting	152

5.12	The “hyperplane-box-region” model and the “prototype” model	155
5.13	Two-dimensional Kohonen feature map	157
5.14	Kohonen map for the voltage security example. Adapted from [TA 94]	159
5.15	Voltage coherency SOM	161
6.1	Hybrid DT-ANN approach	170
6.2	Illustration of distance computations in the attribute space	172
6.3	Uses of distance computations in the attribute space	173
7.1	Different classes of learning methods	183
8.1	Learning approach to power system security assessment	190
8.2	Transient stability behavior : stable vs unstable	192
8.3	Typical marginally stable and unstable swing curves	194
8.4	Equal-area criterion applied to the critical machines of Fig.8.3	195
8.5	Time scales for voltage stability simulations. Adapted from [VA 93b]	197
8.6	Typical EHV PV transmission characteristic	199
8.7	Typical evolution of consumer voltages	200
8.8	Three level decomposition for security studies	202
10.1	Preventive transient stability assessment of a power plant	220
10.2	Automatic off-line construction of a data base	222
10.3	Global decision tree covering 14 contingencies. Adapted from [WE 93e]	224
10.4	Single contingency decision tree for a double-line fault. Adapted from [WE 93e]	226
10.5	Output normalization for the hybrid MLP CCT approximation	227
10.6	Voltage emergency state detection in a weak region. Adapted from [VA 91b]	229
10.7	Construction of a data base of JAD states	231

10.8	Emergency state detection tree. Adapted from [VA 91b]	232
10.9	Critical vs noncritical regions of the DT of Fig. 10.8. Adapted from [WE 93a]	233
10.10	Multilayer perceptron derived from the DT of Fig. 10.8. Adapted from [WE 93a]	234
11.1	Effect of loadflow divergence on the distribution of a power flow	241
11.2	Overview of the learning based security assessment approach	243
12.1	Deriving operating margins	253
13.1	OMIB system	261
13.2	Tree features and number N of learning states. Adapted from [WE 91a]	266
13.3	Tree features and pruning parameter α . Adapted from [WE 91a]	267
13.4	One-line diagram of the EDF system	271
13.5	One-line diagram of the study plant substation. Adapted from [WE 93d]	272
13.6	Statistics relative to the study plant. Adapted from [WE 93d]	274
13.7	3-class DT. Adapted from [WE 93d]	278
13.8	DT1 of Table 13.3 subtree for node D1. Adapted from [WE 93d]	279
13.9	CCT distribution of errors of DT26. Adapted from [WE 93d]	281
13.10	Global decision tree for all 17 faults	290
13.11	Partial view of a contingency dependent tree. Adapted from [WE 93d]	292
13.12	Frequency diagram of the number of simultaneously unstable faults	293
13.13	Contingency ranking via a global DT. Adapted from [PA 93]	294
13.14	Main transmission corridors of the Hydro-Québec system	304
13.15	Convergence diagram of Manic-Québec power flow (6 base case files)	306
13.16	Convergence diagram of Manic-Québec power flow (12 base case files)	306
13.17	Data base generation procedure	307

13.18	Groupings of generators or lines defining stability limits used for the global stability assessment	309
13.19	Partial view of decision tree built with 67 attributes : $N = 8,000$ $M = 2497$	312
13.20	Decision tree built for the 22-North configurations : $N = 2746$ $M = 657$	314
13.21	Improved DT built for the 22-North configurations : $N = 2746$ $M = 657$	316
14.1	Preventive voltage security DT. Adapted from [WE91c]	323
14.2	Distribution of 2000 random states in the (Qatcor,Res-Comb) space. Adapted from [VA93a]	324
14.3	Compound OLTC - Load - Compensation model	326
14.4	Principle of the data base generation	327
14.5	One-line diagram of the study region.	328
14.6	Histogram of the regional pre-disturbance MV load level	329
14.7	Power import and reactive reserve in the study region	329
14.8	Pre- and post-disturbance active load power margin distributions (the relevant 2312 states for disturbance number 1)	331
14.9	Emergency mode detection criteria for various measurement instants	337
14.10	Preventive mode decision trees built with pre-disturbance attributes	338
14.11	Simplified two level tree structure and its security regions	340
14.12	Pre- and post-disturbance active load power margin distributions (the relevant 4041 states for disturbance number 1)	346
14.13	Correlation of pre- and post-disturbance load power margins	346
A.1	Entropy functions for $\beta \in [0.01 \dots 100.0]$	355
A.2	Relationship between D_C^T and C_C^T	360
A.3	Variation of various scores for test $TRBJ < THRESHOLD$	361
A.4	Kolmogorov-Smirnoff distance as a function of $TRBJ$	365

List of Tables

1.1	Security assessment environments. Adapted from [WE 93i]	7
3.1	Rules corresponding to the tree of Fig. 3.2	53
3.2	Hill-climbing tree growing algorithm	57
3.3	Hypothesis testing approach to pruning	59
3.4	Tree post-pruning algorithm	60
3.5	Pruned tree selection algorithm	61
3.6	Weighted object propagation	65
3.7	SIPINA algorithm. Adapted from [ZI 92]	67
3.8	Deriving classification from class probabilities	71
3.9	Optimal threshold identification	75
3.10	Linear combination search	76
3.11	Splitting of the data base by a test	78
3.12	Detailed information about attribute scores and correlations	82
3.13	Percentage of $N * I_C^T$ provided by each test attribute	90
3.14	The CN2 induction algorithm. Adapted from [CL 89]	93
3.15	Iterative adaptation of object weights	97

4.1	Fisher vs logistic linear discriminant. Adapted from [TA 94]	109
4.2	Error rates (%) of $K - NN$ classifiers	115
5.1	Perceptron learning algorithm	136
5.2	Effect of criteria and algorithms on CPU time and quality assessment	141
5.3	Kohonen self-organizing map learning algorithm	158
7.1	Synthetic characterization of supervised learning methods (see text for explanation)	183
13.1	Tree features and number of classes. Adapted from [WE 91a]	265
13.2	Distribution of errors of a 4-class tree. Adapted from [WE 93d]	281
13.3	Effect of the types of candidate attributes. Adapted from [WE 93d]	283
13.4	Quality improvement of DT3 of Table 13.3	286
13.5	Quality improvement of subtree of DT1 of Fig. 13.8	287
13.6	DTs for collective stability assessment	289
13.7	Contingency ranking	295
13.8	CCT approximation via MLPs	296
13.9	CPU times on a 28MIPS Sparc2 SUN workstation	297
13.10	Results obtained in the Statlog project. Adapted from [TA 94]	299
13.11	Tree characteristics for various learning set sizes	311
13.12	Effect of improved attributes on tree characteristics,	313
13.13	$K - NN$ results for the Hydro-Québec system	317
14.1	Results obtained in the Statlog project. Adapted from [TA 94]	320
14.2	Proportion of unstable situations	331
14.3	Single-contingency decision tree performances	333
14.4	Presently used criterion	334
14.5	Multicontingency tree performances	335

14.6 $K - NN$ results for disturbance 1 341

14.7 Multilayer perceptrons built for disturbance 1 342

Bibliography

The numbers in italics between parentheses appearing at the end of a reference identify the pages where this reference is cited.

- [AH 91] D. W. Aha, D. Kibler, and M. K. Albert, *Instance-based learning algorithms*, Machine Learning **6** (1991), 37–66. (*48, 94*)
- [AK 93] V.B. Akella, L. Wehenkel, M. Pavella, M. Trotignon, A. Duchamp, and B. Heilbronn, *Multicontingency decision trees for transient stability assessment*, Procs. of the 11th Power Systems Computation Conference, Aug-Sept 1993, pp. 113–119. (*207, 220, 223, 270, 288*)
- [AN 92] M. Anthony and N. Biggs, *Computational learning theory*, Cambridge University Press, 1992. (*15*)
- [AR 92] R. Araya and P. Gigeon, *Segmentation trees : a new help building expert systems and neural networks*, Procs. of Stats., 1992, pp. 119–124. (*169*)
- [AT 90] L. Atlas, R. Cole, Y. Muthusamy, A. Lippman, J. Connor, D. Park, and M. El-Sharkawi, *A performance comparison of trained multilayer perceptrons and trained classification trees*, Proceedings of the IEEE **78** (1990), no. 10, 1614–1617. (*185*)
- [BE 85] J. O. Berger, *Statistical decision theory and Bayesian analysis*, Springer Verlag, 1985. (*15*)
- [BE 91a] D. Beaulieu, J. Gauthier, and R. Mailhot, *Transits maximums du réseau Baie James à quatre liens avec et sans rejet de production*, Tech. report, Hydro-Québec - Dir. RTI - SR, December 1991, In French. (*304, 314*)
- [BE 91b] R. Belhomme and M. Pavella, *A composite electromechanical distance approach to transient stability*, IEEE Trans. on Power Syst. **PWRS-6** (1991), no. 2, 622–631. (*271*)

- [BI 79] P. Billingsley, *Probability and measure*, John Wiley and Sons, 1979. (39)
- [BL 87] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, *Occam's razor*, *Information Processing Letters* **24** (1987), 377–380. (48)
- [BO 93] B. Bouchon-Meunier, L. Valverde, and R. Yager (eds.), *Uncertainty in intelligent systems*, North-Holland, 1993. (48)
- [BR 84] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*, Wadsworth International (California), 1984. (17, 47, 48, 54, 59, 60, 62, 63, 168, 359, 361, 362, 363)
- [BR 88] I. Bratko and I. Kononenko, *Learning diagnostic rules from incomplete and noisy data*, *AI and Statistics* (B. Phelps, ed.), Technical Press, 1988. (47)
- [BR 93] V. Brandwajn, *Localization concepts in (in)-security analysis*, Proc. of IEEE-NTUA Joint Int. Power Conf. Athens Power Tech, September 1993, pp. 10–15. (13)
- [BU 89] W. L. Buntine, *A critique of the Valiant model*, Procs. of the IJCAI-89, 1989, pp. 837–842. (48)
- [BU 90] W. L. Buntine, *A theory of learning classification rules*, Ph.D. thesis, School of Computing Science, Sidney University of Technology, February 1990. (35, 48, 62, 364)
- [BU 91] W. L. Buntine and A. S. Weigend, *Bayesian back-propagation*, *Complex Systems* **5** (1991), 603–643. (48, 164)
- [BU 92] W. L. Buntine, *Learning classification trees*, *Statistics and Computing* **2** (1992), 63–73. (47, 67, 68)
- [CA 84] J. Carpentier, *Voltage collapse proximity indicators computed from an optimal power flow*, Procs. of the 8th Power Systems Computation Conference, 1984, pp. 671–678. (201, 252)
- [CA 87] C. Carter and J. Catlett, *Assessing credit card applications using machine learning*, *IEEE Expert* **Fall** (1987), 71–79. (66, 181)
- [CA 93a] J. Carpentier, *Static security assessment and control : a short survey*, Proc. of IEEE-NTUA Joint Int. Power Conf. Athens Power Tech, September 1993, pp. 1–9. (13)
- [CA 93b] P. Caseau, *Keynote address*, Procs. of the 11th Power Systems Computation Conference, August 1993. (211)
- [CE 93] V. Centeno, J. De La Ree, A. G. Phadke, G. Michel, J. Murphy, and R. Burnett, *Adaptive out-of-step relaying using phasor measurement techniques*, *IEEE Computer Applications in Power* (1993), no. 4, 12–17. (228)

- [CH 85] P. Cheeseman, *In defense of probability*, Proc. of the IJCAI-85, 1985, pp. 1002–1009. (39, 48)
- [CH 88a] P. Cheeseman, M. Self, J. Kelly, and J. Stutz, *Bayesian classification*, Proc. of the 7th AAAI Conf., 1988, pp. 607–611. (37, 48, 126)
- [CH 88b] P. A. Chou, *Application of information theory to pattern recognition and the design of decision trees and trellises*, Ph.D. thesis, Stanford University, June 1988. (66)
- [CH 91] P. A. Chou, *Optimal partitioning for classification and regression trees*, IEEE Trans. on Pattern Analysis and Machine Intelligence **PAMI-13** (1991), no. 14, 340–354. (43, 54, 359, 364)
- [CI 92] K. J. Cios and N. Liu, *A machine learning method for generation of a neural network architecture : a continuous ID3*, IEEE Transactions on neural networks **3** (1992), no. 2, 280–291. (168)
- [CL 89] P. Clark and T. Niblett, *The CN2 induction algorithm*, Machine Learning **3** (1989), 261–283. (48, 92, 93, 94, 395)
- [CO 91] S. Cost and S. Salzberg, *A weighted nearest neighbor algorithm for learning with symbolic features*, Tech. report, Dept. of Computer Science, John Hopkins University, 1991. (48, 94, 95, 96, 97)
- [DA 70] Z. Daróczy, *Generalized information functions*, Information and Control **16** (1970), 36–51. (76, 353, 354)
- [DE 76] P. A. Devijver, *Entropie quadratique et reconnaissance de formes*, NATO ASI Series, Computer Oriented Learning Processes (JC. Simon, ed.), Noordhoff, Leyden, 1976. (353, 354, 359)
- [DE 82] P. A. Devijver and J. Kittler, *Pattern recognition : A statistical approach*, Prentice-Hall International, 1982. (16, 20, 44, 45, 105, 113, 114, 119, 123, 126, 128, 156)
- [DE 90] K. A. Dejong, *Genetic algorithm based learning*, Machine Learning III (Y. Kodratoff and R. Michalski, eds.), Morgan Kaufman, 1990, pp. 611–638. (98, 100)
- [DE 92] F.P. de Mello, J. W. Feltes, T. F. Laskwski, and L. J. Opper, *Simulating fast and slow dynamic effects in power systems*, IEEE Computer applications in power **5** (1992), no. 3. (9)
- [DO 86] J. C. Dodu and A. Merlin, *New probabilistic approach taking into account reliability and operation security in EHV power system planning at EDF*, IEEE Trans. on Power Syst. **PWRS-1** (1986), 175–181. (210)

- [DU 73] R. O. Duda and P. E. Hart, *Pattern classification and scene analysis*, John Wiley and Sons, 1973. (15, 16, 20, 37, 63, 104, 105, 106, 119, 122, 124, 126, 158)
- [DY 67] T. E. DyLiacco, *The adaptive reliability control system*, IEEE Trans. on power apparatus and systems **PAS-86** (1967), no. 5, 517–531. (3)
- [DY 68] T. E. DyLiacco, *Control of power systems via the multi-level concept*, Ph.D. thesis, Sys. Res. Center, Case Western Reserve Univ., 1968, Rep. SRC-68-19. (2, 104)
- [DY 93] T. E. DyLiacco, *On the open road to enhancing the value of control centers to system operation and the utility enterprise*, Procs. of APSCOM-93, IEE Int. conf. on advances in power system Control, Operation and Management (Invited), December 1993, pp. 24–29. (216)
- [ED 70] Edison Electric Institute, *On-line stability analysis study*, Tech. report, North American Rockwell Information System Company, 1970. (2)
- [EL 89] M. A. El-Sharkawi, R. J. Marks, M. E. Aggoune, D. C. Park, M. J. Damborg, and L. E. Atlas, *Dynamic security assessment of power systems using back error propagation artificial neural networks*, Procs. of the 2nd Symposium on Expert Systems Application to power systems, 1989, pp. 366–370. (236, 238)
- [EU 92] E. Euxibie, M. Goubin, B. Heilbronn, L. Wehenkel, Y. Xue, T. Van Cutsem, and M. Pavella, *Prospects of application to the French system of fast methods for transient stability and voltage security assessment*, CIGRE Report 38-208, Paris, Aug.-Sept. 1992. (8, 276)
- [FA 90] F.C. Fahlman and C. Lebière, *The cascaded-correlation learning architecture*, Advances in Neural Information Processing Systems II (D. S. Touretzky, ed.), Morgan Kaufmann, 1990, pp. 524–532. (118)
- [FI 78] L. H. Fink and K. Carlsen, *Operating under stress and strain*, IEEE Spectrum **15** (1978), no. 3, 48–53. (4, 389)
- [FI 89] D. H. Fisher and K. B. McKusick, *An empirical comparison of ID3 and back-propagation*, Procs. of the IJCAI-89, 1989, pp. 788–793. (185)
- [FO 92] M. Fombellida and J. Destiné, *Méthodes heuristiques et méthodes d'optimisation non contraintes pour l'apprentissage des perceptrons multicouches*, Procs. of NEURO-NIMES 92, Fifth International Conference on Neural Networks and their Applications, 1992. (148)
- [FR 77] J. H. Friedman, *A recursive partitioning decision rule for nonparametric classification*, IEEE Trans. on Computers **C-26** (1977), 404–408. (17, 58, 63, 364)

- [FR 81] J. H. Friedman and W. Stuetzle, *Projection pursuit regression*, Jour. of the Am. Stat. Ass. **76** (1981), no. 376, 817–823. (16, 118, 164)
- [FR 84] J. H. Friedman, W. Stuetzle, and A. Schroeder, *Projection pursuit density estimation*, Jour. of the Am. Stat. Ass. **79** (1984), no. 387, 599–608. (16)
- [FR 87] J. H. Friedman, *Exploratory projection pursuit*, Jour. of the Am. Stat. Ass. **82** (1987), no. 397, 249–266. (16)
- [GA 89] Q. Gao and M. Li, *The minimum description length principle and its application to on-line learning of handprinted characters*, Procs. of the IJCAI-89, 1989, pp. 843–848. (48)
- [GE 93a] S. Geeves and TF 38-02-09, *Assessment of practical fast transient stability methods : state of the art report*, Tech. report, CIGRE, to appear in 1993. (10)
- [GE 93b] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manecek, and V. Sunderam, *PVM 3 user's guide and reference manual*, Tech. Report ORNL/TM-12187, Oak Ridge National Laboratory, 1993. (217)
- [GL 93] H. Glavitsch, *Power system security enhanced by post-contingency switching and rescheduling*, Proc. of IEEE-NTUA Joint Int. Power Conf. Athens Power Tech, September 1993, pp. 16–21. (6)
- [GO 89a] D. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, 1989. (98)
- [GO 89b] M. Goubin, *Cadre d'une étude pour évaluer la possibilité d'utiliser les arbres de décisions pour la détection des états critiques en tension*, Tech. Report HR-46/833, EDF - DER, 1989. (319)
- [GU 93] S. Guiasu, *A unitary treatment of several known measures of uncertainty induced by probability, possibility, fuzziness, plausibility and belief*, Uncertainty in Intelligent Systems (B. Bouchon-Meunier, L. Valverde, and R.R. Yager, eds.), Elsevier - North Holland, 1993, pp. 355–365. (356)
- [HA 81] D. J. Hand, *Discrimination and classification*, John Wiley and Sons, 1981. (16, 20, 105, 119, 120, 126)
- [HA 90] Y. Harmand, M. Trotignon, J. F. Lesigne, J. M. Tesson, C. Lemaître, and F. Bourgin, *Analyse d'un cas d'écroulement en tension et proposition d'une philosophie de parades fondées sur des horizons temporels différents*, CIGRE Report 38/39-02, Paris, August 1990. (6, 319, 322)
- [HA 92] H. Hakim, *Application of pattern recognition in transient stability assessment*, Electric Machines and Power Systems **20** (1992), 1–15. (104)

- [HE 69] E. G. Henrichon and K. S. Fu, *A non-parametric partitioning procedure for pattern classification*, IEEE Trans. on Computers (1969), no. 7, 614–624. (49, 58)
- [HE 91] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the theory of neural computation*, Addison Wesley, 1991. (18, 134, 147, 153, 156, 159)
- [HO 75] J. H. Holland, *Adaptation in natural and artificial systems*, Michigan Press, 1975. (97)
- [HU 66] E. B. Hunt, J. Marin, and P. J. Stone, *Experiments in induction*, Wiley, 1966. (47, 49)
- [HW 93] J. N. Hwang, S. S. You, S. R. Lay, and I. C. Jou, *What's wrong with a cascaded correlation learning network : a projection pursuit learning perspective*, Tech. report, Info. Proc. Lab., Dep.t of Elec. Eng., University of Washington, September 1993. (118, 164)
- [IE 90] IEEE System Dynamic Performance Subcommittee of the power system engineering committee of the PES, *Voltage stability of power systems : concepts, analytical tools, and industry experience*, Tech. Report 90TH0358-2-PWR, IEEE, 1990. (196)
- [IE 92a] IEEE PES Power System Eng. Ctte. Power System Restoration Working Group, *New approaches in power system restoration*, IEEE Trans. on Power Syst. **7** (1992), no. 4, 1428–1434. (4)
- [IE 92b] IEEE Task Force on Load Representation for Dynamic Performance, *Load representation for dynamic performance analysis*, Paper # 92 WM 126–3–PWRD. (250)
- [JA 93] Y. Jacquemart, L. Wehenkel, and T. Van Cutsem, *Analyse de la sécurité de tension par la méthode des arbres de décision. Présentation du modèle de réseau et de la génération de situations de conduite*, Tech. report, University of Liège, December 1993, Report of contract EDF/CIRC No. R46L14/ER178. (344)
- [KO 84] I. Kononenko, I. Bratko, and E. Roskar, *Experiments in automatic learning of medical diagnosis rules*, Tech. report, Jozef Stefan Institute, 1984. (47, 59)
- [KO 90] T. Kohonen, *The self-organizing map*, Proceedings of the IEEE **78** (1990), no. 9, 1464–1480. (18, 20, 156, 157, 163)
- [KV 87] T. O. Kvålseth, *Entropy and correlation: some comments*, IEEE Trans. on Systems, Man and Cybernetics **SMC-17** (1987), no. 3, 517–519. (77, 79, 84, 262, 358, 359, 362)

- [LA 89] P. Lagonotte, J. C. Sabonnadière, J. Y. Léost, and J. P. Paul, *Structural analysis of the electrical system : application to the secondary voltage control in France*, IEEE Trans. on Power Syst. **PWRS-4** (1989), no. 4, 479–486. (23)
- [LE 72] S.T.Y. Lee, *Transient stability equivalents for power system planning*, Ph.D. thesis, Massachusetts Institute of Technology, 1972. (263)
- [LE 90a] C. Lemaître, J. P. Paul, J. M. Tesson, Y. Harmand, and Y. S. Zhao, *An indicator of the risk of voltage profile instability for real-time control applications*, IEEE Trans. on Power Syst. **PWRS-5** (1990), no. 1, 148–161. (12, 198, 339)
- [LE 90b] J. F. Lesigne, *Organisation d'une étude de stabilité dans l'environnement de l'exploitation*, Tech. Report D7061/SET/89/JFL/AR/No51, Electricité de France - SME - CNME, February 1990. (237, 243)
- [LE 90c] E. Levin, N. Tishby, and S. A. Solla, *A statistical approach to learning and generalization in layered neural networks*, Proceedings of the IEEE **78** (1990), no. 10, 1568–1574. (164)
- [LI 89] C.C. Liu, Shing-Ming Wang, H.Y. Marathe L. Wong, and M.G. Lauby, *A self learning expert system for voltage control of power systems*, Proc. 2nd Symp. on Expert Systems Application to Power Systems, 1989, pp. 462–468. (55, 321)
- [LI 91] C.C. Liu and Shing-Ming Wang, *Development of expert systems and their learning capability for power system applications*, Academic Press Series on Advances in Control and Dynamic Systems, Academic Press, 1991. (321)
- [LO 80] J. Lorigny, *Théorie des questionnaires et reconnaissance des intitulés - premier bilan : le chiffrement du code profession*, Tech. report, Institut Nat. de la Stat. et des Et. Econ., 1980, In French. (47)
- [LO 91] R. López de Mántaras, *A distance-based attributes selection measure for decision tree induction*, Machine Learning **6** (1991), 81–92, Technical Note. (358, 360, 361)
- [MA 90] M. A. Maria, C. Tang, and J. Kim, *Hybrid transient stability analysis*, IEEE Trans. on Power Syst. (1990), no. 2, 384–393. (11)
- [MC 43] W. S. McCulloch and W. Pitts, *A logical calculus of ideas immanent in nervous activity*, Bulletin of Mathematical Biophysics **5** (1943), 115–133. (133)

- [MC 52] P. McCullagh and J.A. Nelder, *Generalized linear models*, Chapman and Hall, 1982. (109)
- [ME 89] C. J. Metheus and L. A. Rendell, *Constructive induction on decision trees*, Procs. of the IJCAI-89, 1989, pp. 645–650. (14)
- [ME 92] B. Meyer and M. Stubbe, *EUROSTAG, a single tool for power system simulation*, Transmission and Distribution International **3** (1992), no. 1, 47–52. (9)
- [MI 69] M. L. Minsky and S. A. Papert, *Perceptrons*, MIT Press, 1969. (133)
- [MI 81] T. A. Mikolinnas and B. F. Wollenberg, *An advanced contingency selection algorithm*, IEEE Trans. on Power App. and Syst. **PAS-100** (1981), no. 2, 608–617. (13)
- [MI 83] R. S. Michalski, *A theory and methodology of inductive learning*, Artificial Intelligence **20** (1983), 111–161. (47, 92)
- [MI 84] R. S. Michalski, J. G. Carbonell, and T. M. Mitchel (eds.), *Machine learning : an artificial intelligence approach*, Springer Verlag, 1984. (20)
- [MI 86] R. S. Michalski, J. G. Carbonell, and T. M. Mitchel (eds.), *Machine learning II*, Morgan Kaufmann, 1986. (20)
- [MI 89a] J. Mingers, *An empirical comparison of pruning methods for decision tree induction*, Machine Learning **4** (1989), 227–243. (60, 62)
- [MI 89b] J. Mingers, *An empirical comparison of selection measures for decision tree induction*, Machine Learning **3** (1989), 319–342. (62, 358, 361)
- [MI 92] E. Miconnet, T. Van Cutsem, and L. Wehenkel, *Application the la méthode des arbres de décision à la détection des états critiques en tension*, Tech. report, University of Liège, October 1992, Final report of contract EDF/CIRC No. R46L14. (319, 326, 328, 330)
- [MO 63] J. N. Morgan and J. A. Sonquist, *Problems in the analysis of survey data, and a proposal*, J. of the Amer. Stat. Ass. **58** (1963), 415–434. (17, 49)
- [MO 89] R. Mooney, J. Shavlik, G. Towell, and A. Gove, *An experimental comparison of symbolic and connectionist learning algorithms*, Procs. of the IJCAI-89, 1989, pp. 775–780. (185)
- [MO 90] D. J. Montana, *Empirical learning using rule threshold optimization for detection of events in synthetic images*, Machine Learning **5** (1990), 427–450. (100)

- [MO 91] H. Mori and Y. Tamura, *An artificial neural-net based approach to power system voltage stability*, Procs. of the 2nd Int. Workshop on Bulk Power System Voltage Phenomena - Voltage Stability and Security, August 1991, pp. 347–358. (159)
- [MU 93] S. Murthy, S. Kasif, S. Salzberg, and R. Beigel, *OCI : randomized induction of oblique trees*, Procs. of the AAAI-93, 1993. (63)
- [NI 91] D. Niebur and A. Germond, *Power system static security assessment using the Kohonen neural network classifier*, Procs. of the IEEE Power Industry Computer Application Conference, May 1991, pp. 270–277. (159)
- [NO 91] North American Electricity Reliability Council, *Survey of the voltage collapse phenomenon - Summary of the Interconnection Task Force*, Tech. report, NERC, 1991. (6, 196)
- [OH 86] Y. Ohura, K. Matsuzawa, H. Ohtsuka, N. Nagai, T. Gouda, H. Oshida, S. Takeda, and S. Nishida, *Development of a generator tripping system for transient stability augmentation based on the energy function method*, IEEE Trans. on Power Delivery **PWRD-1** (1986), no. 3, 17–24. (194)
- [OS 91] D. R. Ostojic and G. T. Heydt, *Transient stability assessment by pattern recognition in the frequency domain*, IEEE Trans. on Power Syst. **PWRS-6** (1991), no. 1, 231–237. (228)
- [PA 82] Y. H. Pao, *Feasibility of using associative memories for static security assessment of power system overloads*, Tech. Report EPRI EL-2343, Electric Power Research Institute, 1982. (2)
- [PA 85] Y. H. Pao, T. E. DyLiacco, and I. Bozma, *Acquiring a qualitative understanding of system behavior through AI inductive inference*, Procs. of the IFAC Symp. on Electric Energy Systems, 1985, pp. 35–41. (2)
- [PA 87] D. B. Parker, *Optimal algorithms for adaptive networks : second order back propagation, second order direct propagation, second order Hebbian learning*, Procs. of IEEE First Int. Conf. on Neural Networks, 1987, pp. 593–600. (148)
- [PA 89a] K. R. Padiyar and K. K. Ghosh, *Direct stability evaluation of power systems with detailed generator models using structure preserving energy functions*, Int. J. of Elec. Power and Energy Syst. **11** (1989), no. 1, 47–56. (11)
- [PA 89b] Y. H. Pao, *Adaptive pattern recognition and neural networks*, Addison-Wesley, 1989. (134, 156)
- [PA 93] M. Pavella and P. G. Murthy, *Transient stability of power systems; theory and practice*, John Wiley, 1993. (191, 195, 261, 294, 392)

- [PE 88] J. Pearl, *Probabilistic reasoning in intelligent systems - networks of plausible inference*, Morgan-Kaufman, 1988. (39, 48)
- [PE 92] M. V. F. Pereira, M. E. P. Maceira, G. C. Oliveira, and L. M. V. G. Pinto, *Combining analytical models and Monte-Carlo techniques in probabilistic power system analysis*, IEEE Trans. on Power Syst. **PWRS-7** (1992), 265–272. (244)
- [PO 72] R. Poncelet, *Contribution à la conduite et à la protection des réseaux électriques par calculateurs numériques*, Ph.D. thesis, Université Libre de Bruxelles, 1972, In French. (2)
- [QU 83] J. R. Quinlan, *Learning efficient classification procedures and their application to chess endgames.*, Machine Learning : An artificial intelligence approach. (R. S. Michalski, J. Carbonell, and T. Mitchell, eds.), Morgan Kaufman, 1983, pp. 463–482. (17, 47, 56, 57)
- [QU 86a] J. R. Quinlan, *The effect of noise on concept learning.*, Machine Learning II. (R. S. Michalski, J. Carbonell, and T. Mitchell, eds.), Morgan Kaufman, 1986, pp. 149–166. (59, 83)
- [QU 86b] J. R. Quinlan, *Induction of decision trees*, Machine Learning **1** (1986), 81–106. (62, 66, 358)
- [QU 87a] J. R. Quinlan, *Generating production rules from decision trees*, Procs. of the 10th Int. Joint Conf. on Artificial Intelligence, 1987, pp. 304–307. (92)
- [QU 87b] J. R. Quinlan, *Simplifying decision trees*, Int. J. of Man-Mach. Studies **27** (1987), 221–234. (60, 364)
- [QU 89] J.R. Quinlan and R.L. Rivest, *Inferring decision trees using the minimum description length principle*, Information and Computation **80** (1989), 227–248. (48)
- [QU 90] J. R. Quinlan, *Learning logical definitions from relations*, Machine Learning **5** (1990), no. 3, 229–266. (91)
- [QU 91] J. R. Quinlan, *Knowledge acquisition from structured data - using determinate literals to assist search*, IEEE Expert **6** (1991), no. 6, 32–37. (91)
- [RA 91] F. A. Rahimi, *Evaluation of transient energy function method software for dynamic security analysis*, Tech. Report EPRI EL-7357 Project 4000-18, Electric Power Research Institute, 1991. (11)
- [RE 92] D. Reichelt and H. Glavitsch, *Features of a hybrid expert system for security enhancement*, IEEE Trans. on Power Syst. **7** (1992), no. 2, 907–913. (13)

- [RE 93] N. D. Reppen, R. R. Austria, J. A. Uhrin, M. C. Patel, and A. Galatic, *Performance of methods for ranking and evaluation of voltage collapse contingencies applied to large-scale network*, Proc. of IEEE-NTUA Joint Int. Power Conf. Athens Power Tech, September 1993, pp. 337–343. (12)
- [RI 78] J. Rissanen, *Modelling by shortest data description*, Automatica **14** (1978), 465–471. (48)
- [RI 83] J. Rissanen, *A universal prior for integers and estimation by minimum description length*, Ann. of Statistics **11** (1983), 416–431. (48)
- [RI 90] L. Riverin, *Activité établissement des limites de transits du réseau Hydro-Québec*, Personal communication, 1990. (8, 239, 243, 301)
- [RI 91] M. D. Richard and R. P. Lippmann, *Neural network classifiers estimate Bayesian a posteriori probabilities*, Neural Computation **3** (1991), 461–483. (146, 164, 356)
- [RI 93] B. D. Ripley, *Statistical aspects of neural networks*, Proc. of SemStat, Chapman & Hall, January 1993. (164)
- [RO 63] F. Rosenblatt, *Principles of neurodynamics*, Spartan, 1963. (133)
- [RO 80] E. M. Rounds, *A combined nonparametric approach to feature selection and binary decision tree design*, Pattern recognition **12** (1980), 313–317. (59, 364)
- [RO 93] S. Rovnyak, S. Kretsinger, J. Thorp, and D. Brown, *Decision trees for real-time transient stability prediction*, Paper # 93 SM 530–6–PWRS. (8, 228)
- [RU 86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning representations by back-propagating errors*, Nature **323** (1986), 533–536. (19, 133)
- [SA 91a] S. R. Safavian and D. Landgrebe, *A survey of decision tree classifier methodology*, IEEE Trans. on Syst., Man and Cybernetics **21** (1991), no. 3, 660–674. (17, 62)
- [SA 91b] S. Salzberg, *A nearest hyperrectangle learning method*, Machine Learning **6** (1991), 251–276. (48, 94, 97)
- [SA 91c] T. D. Sanger, *A tree-structured algorithm for reducing computation in networks with separable basis functions*, Neural Computation **3** (1991), 67–78. (168)
- [SC 93] C. Schaffer, *Overfitting as bias*, Machine Learning **10** (1993), 153–178. (35)

- [SE 85] J. Segen, *Learning concept descriptions from examples with errors*, Procs. of the IJCAI-85, 1985, pp. 634–636. (48)
- [SE 90] I.K. Sethi, *Entropy nets : from decision trees to neural networks*, Proceedings of the IEEE **78** (1990), no. 10, 1605–1613. (169)
- [SH 91] J. Shavlik, R. Mooney, and G. Towell, *Symbolic and neural learning algorithms : an experimental comparison*, Machine Learning **6** (1991), 111–143. (185)
- [SO 83] R. Sorkin, *A quantitative Occam's razor*, Int. J. of Theoretical Physics **22** (1983), 1091–1113. (48)
- [ST 86] C. Stanfill and D. Waltz, *Toward memory-based reasoning*, Communications of the ACM **29** (1986), no. 12, 1213–1228. (48, 94, 95)
- [ST 92] P. Stoa, S. N. Talukdar, R. D. Christie, L. Hou, and N. Papanikolopoulos, *Environments for security assessment and enhancement*, Int. J. of Elec. Power and Energy Syst. **14** (1992), no. 2/3, 249–255. (13)
- [ST 93] M. Stubbe, A. Bihain, and J. Deuse, *Simulation of voltage collapse*, Int. J. of Elec. Power and Energy Syst. **15** (1993), no. 4, 239–244. (11)
- [TA 83] Y. Tamura, H. Mori, and S. Iwamoto, *Relationship between voltage instability and multiple load flow solutions in electric power systems*, IEEE Trans. on Power App. and Syst. **PAS-102** (1983), no. 5, 1115–1125. (252)
- [TA 94] C. Taylor (ed.), *Machine learning, neural and statistical classification*, Ellis Horwood, To appear in 1994, Final rep. of ESPRIT project 5170 - Statlog. (15, 48, 62, 92, 109, 153, 159, 160, 163, 185, 298, 299, 320, 391, 396)
- [TO 74] G. T. Toussaint, *Bibliography on estimation of misclassification*, IEEE Trans. on Information Theory **IT-20** (1974), no. 4, 472–479. (44)
- [TO 93] G. G. Towell and J. W. Shavlik, *Extracting refined rules from knowledge-based neural networks*, Machine Learning **13** (1993), 71–101. (154)
- [UT 88] P.E. Utgoff, *Perceptron trees : a case study in hybrid concept representation*, AAAI-88. Procs. of the 7th Nat. Conf. on Artificial Intelligence, Morgan Kaufman, 1988, pp. 601–606. (63, 168)
- [UT 89] P. E. Utgoff, *Incremental induction of decision trees*, Machine Learning **4** (1989). (65)
- [VA 84] L. G. Valiant, *A theory of the learnable*, Communications of the ACM **27** (1984), no. 11, 1134–1142. (48)

- [VA 91a] T. Van Cutsem, *A method to compute reactive power margins with respect to voltage collapse*, IEEE Trans. on Power Syst. **PWRS-6** (1991), no. 2, 145–156. (12, 198, 252)
- [VA 91b] T. Van Cutsem, L. Wehenkel, M. Pavella, B. Heilbronn, and M. Goubin, *Decision trees for detecting emergency voltage conditions*, Proc. of the 2nd Int. NSF Workshop on Bulk Power System Voltage Phenomena - Voltage Stability and Security, Deep Creek Lake, Ma, August 1991, pp. 229–240. (201, 229, 230, 232, 391, 392)
- [VA 93a] T. Van Cutsem, L. Wehenkel, M. Pavella, B. Heilbronn, and M. Goubin, *Decision tree approaches for voltage security assessment*, IEE Proceedings - Part C. **140** (1993), no. 3, 189–198. (319, 324, 393)
- [VA 93b] T. Van Cutsem, *Analysis of emergency voltage situations*, Proc. of the 11th Power Systems Computation Conference, Aug-Sept 1993, pp. 323–330. (12, 197, 201, 247, 326, 342, 391)
- [VA 93c] T. Van Cutsem, *An approach to corrective control of voltage instability using simulation and sensitivity*, Proc. of IEEE-NTUA Joint Int. Power Conf. Athens Power Tech, September 1993, pp. 460–470. (12)
- [VE 92] V. Venkatasubramanian, H. Schättler, and J. Zaborszky, *A stability theory of large differential algebraic systems - A taxonomy*, Tech. Report SSM 9201 - Part I, Dept. of System Science and Math., Washington University, 1992. (9)
- [VI 86] M. Vincelette and D. Landry, *Stability limit selection of the Hydro-Québec power system : a new software philosophy*, Procs. of the 2nd Int. IEE Conf. on Power Syst. Monitoring and Control, 1986, pp. 367–371. (69, 301)
- [WA 87] R.L. Watrous, *Learning algorithms for connectionist networks : applied gradient methods of nonlinear optimization*, Procs. of IEEE First Int. Conf. on Neural Networks, 1987, pp. 619–627. (148)
- [WE 74] P. J. Werbos, *Beyond regression : new tools for prediction and analysis in the behavioral sciences*, Ph.D. thesis, Harvard University, 1974. (19)
- [WE 86] L. Wehenkel, T. Van Cutsem, and M. Ribbens-Pavella, *Artificial intelligence applied to on-line transient stability assessment of electric power systems (short paper)*, Proc. of the 25th IEEE Conf. on Decision and Control (CDC), December 1986, pp. 649–650. (68, 83, 236)
- [WE 87a] L. Wehenkel, Y. Xue, T. Van Cutsem, and M. Ribbens-Pavella, *Machine learning applied to power systems transient security functions*, Proc. of the IMACS Int. Symp. on AI, Experts Systems and Languages in Modelling and Simulation, June 1987, pp. 243–248. (195)

- [WE 87b] L. Wehenkel, T. Van Cutsem, and M. Ribbens-Pavella, *Artificial intelligence applied to on-line transient stability assessment of electric power systems*, Proc. of the 10th IFAC World Congress, July 1987, pp. 308–313. (83)
- [WE 88] L. Wehenkel, T. Van Cutsem, and M. Ribbens-Pavella, *Decision trees applied to on-line transient stability assessment of electric power systems*, Procs. of the IEEE Int. Symposium on Circuits and Systems, June 1988, pp. 1887–1890. (171, 172, 181, 268)
- [WE 89a] L. Wehenkel, T. Van Cutsem, and M. Ribbens-Pavella, *Inductive inference applied to on-line transient stability assessment of electric power systems*, Automatica **25** (1989), no. 3, 445–451. (261)
- [WE 89b] L. Wehenkel, T. Van Cutsem, and M. Ribbens-Pavella, *An artificial intelligence framework for on-line transient stability assessment of power systems*, IEEE Trans. on Power Syst. **PWRS-4** (1989), 789–800. (59, 68, 83, 359)
- [WE 89c] S. Weiss and I. Kapouleas, *An empirical comparison of pattern recognition, neural net, and machine learning classification methods*, Procs. of the IJCAI-89, 1989, pp. 781–787. (186)
- [WE 90a] L. Wehenkel, *Une approche de l'intelligence artificielle appliquée à l'évaluation de la stabilité transitoire des réseaux électriques*, Ph.D. thesis, University of Liège - Belgium, May 1990, In French. (2, 35, 73, 76, 84, 174, 195, 261, 263, 268, 354, 355, 356, 358)
- [WE 90b] L. Wehenkel, *Evaluation de la stabilité transitoire. Calcul des indicateurs fournis par la méthode DTTS. - Rapport de la phase C-1990*, Tech. report, University of Liège, December 1990, In French. (220, 223, 270)
- [WE 90c] S. M. Weiss, R. S. Galen, and P. V. Tadepalli, *Maximizing the predictive value of production rules*, Artificial Intelligence **45** (1990), 47–71. (92)
- [WE 91a] L. Wehenkel and M. Pavella, *Decision trees and transient stability of electric power systems*, Automatica **27** (1991), no. 1, 115–134. (68, 261, 264, 265, 266, 267, 283, 392, 396)
- [WE 91b] L. Wehenkel, *Etude de la stabilité du plan de tension au niveau d'une région. Exploitation des ensembles d'apprentissage fournis par le LAIH de Valenciennes*, Tech. report, University of Liège, April 1991, in French. (319, 324)
- [WE 91c] L. Wehenkel, T. Van Cutsem, M. Gilliard, M. Pavella, B. Heilbronn, and M. Goubin, *Decision trees for preventive voltage stability assessment*, Procs. of the 2nd Int. NSF Workshop on Bulk Power System Voltage Phenomena - Voltage Stability and Security, Deep Creek Lake, Ma, August 1991, pp. 217–228. (200, 319, 321, 323, 393)

- [WE 91d] L. Wehenkel, *Evaluation de la stabilité transitoire. Calcul des indicateurs fournis par la méthode DTTS. Investigations relatives aux attributs candidats - Rapport de la phase 1-1991*, Tech. report, University of Liège, September 1991, In French. (220, 223, 270)
- [WE 91e] L. Wehenkel, *Evaluation de la stabilité transitoire. calcul des indicateurs fournis par la méthode DTTS - Investigations relatives à l'amélioration de la qualité des arbres de décision - Rapport de la phase 2-1991*, Tech. report, University of Liège, December 1991, In French. (220, 223, 270)
- [WE 91f] S.M. Weiss and C.A. Kulikowski, *Computer systems that learn*, Morgan Kaufmann, USA, 1991. (15, 44, 46, 48)
- [WE 92a] L. Wehenkel, *Application de la méthode des arbres de décision à la détection des états critiques en tension - Compléments - Prolongements*, Tech. report, University of Liège, October 1992, Final report of contract EDF/CIRC No. R46L14. (319, 341)
- [WE 92b] L. Wehenkel, *An information quality based decision tree pruning method*, Procs. of the 4th Int. Congr. on Information Processing and Management of Uncertainty in Knowledge based Systems - IPMU'92, July 1992, pp. 581–584. (76, 267)
- [WE 93a] L. Wehenkel and V.B. Akella, *A hybrid decision tree - neural network approach for power system dynamic security assessment*, Procs. of the 4th Int. Symp. on Expert Systems Application to Power Systems, Melbourne, Australia, January 1993, pp. 285–291. (134, 169, 185, 233, 234, 270, 276, 296, 392)
- [WE 93b] L. Wehenkel and M. Pavella, *Decision tree approach to power system security assessment*, Int. J. of Elec. Power and Energy Syst. **15** (1993), no. 1, 13–36. (68)
- [WE 93c] L. Wehenkel, *Construction automatique d'arbres de décision pour la détermination de limites de transits du réseau Hydro-Québec - Spécification de la base de données*, Tech. report, University of Liège, January 1993, In French. (304, 305, 361)
- [WE 93d] L. Wehenkel, M. Pavella, E. Euxibie, and B. Heilbronn, *Decision tree based transient stability assessment - a case study*, Paper # 93 WM 235–2 PWRs. (220, 223, 270, 272, 275, 278, 279, 281, 283, 292, 392, 396)
- [WE 93e] L. Wehenkel, *Evaluation de la sécurité en temps réel : approche par arbres de décision*, Actes de la journée d'études SEE, *Intégration des techniques de l'intelligence artificielle dans la conduite et la gestion des réseaux électriques*, March 1993, pp. 11–20. (224, 226, 391)

- [WE 93f] L. Wehenkel and I. Houben, *Construction automatique d'arbres de décision pour la détermination de limites de transits du réseau Hydro-Québec - 1992-1994 - Rapport d'activités des phases B et C*, Tech. report, University of Liège, October 1993, In French. (310, 314)
- [WE 93g] L. Wehenkel, T. Van Cutsem, and Y. Jacquemart, *Analyse de la sécurité de tension par la méthode des arbres de décision. Questions relatives à la génération d'une base de données*, Tech. report, University of Liège, June 1993, Report of contract EDF/CIRC No. R46L14/ER178. (343)
- [WE 93h] L. Wehenkel, *Decision tree pruning using an additive information quality measure*, Uncertainty in Intelligent Systems (B. Bouchon-Meunier, L. Valverde, and R.R. Yager, eds.), Elsevier - North Holland, 1993, pp. 397–411. (60, 68, 73, 76, 86, 267, 390)
- [WE 93i] L. Wehenkel and M. Pavella, *Advances in decision trees applied to power system security assessment*, Procs. of APSCOM-93, IEE Int. conf. on advances in power system Control, Operation and Management (Invited), December 1993, pp. 47–53. (7, 395)
- [WE 94a] L. Wehenkel, *A hybrid decision tree - artificial neural network approach for power system security assessment*, Tech. report, University of Liège - Belgium, 1994, Thèse annexe A à la thèse d'agrégation. (2, 35, 73, 76, 84, 174, 195, 261, 263, 268, 354, 355, 356, 358)
- [WE 94b] L. Wehenkel, *A quality measure of decision trees. Interpretations, justifications, extensions*, Tech. report, University of Liège - Belgium, 1994, Thèse annexe B à la thèse d'agrégation. (52, 54, 63, 64, 73, 128, 147, 168)
- [WE 94c] L. Wehenkel, *Margin regression techniques for voltage security assessment*, Tech. report, University of Liège - Belgium, 1994, Thèse annexe C à la thèse d'agrégation. (321, 339, 347)
- [WI 70] J. L. Willems, *Stability theory of dynamical systems*, Th. Nelson and Sons, 1970. (193)
- [WO 93] D. H. Wolpert, *On overfitting as bias*, Tech. Report SFI TR 92-03-5001, The Santa Fe Institute, March 1993. (35)
- [XU 88] Y. Xue, *Extended equal area criterion : a new method for transient stability assessment and preventive control of power systems*, Ph.D. thesis, University of Liège - Belgium, September 1988. (195, 264)
- [XU 89] Y. Xue, Th. Van Cutsem, and M. Ribbens-Pavella, *Extended equal area criterion : justifications, generalizations, applications*, IEEE Trans. on Power Syst. **PWRS-4** (1989), no. 1, 44–52. (264)

- [XU 92] Y. Xue, L. Wehenkel, R. Belhomme, P. Rousseaux, M. Pavella, E. Euxibie, B. Heilbronn, and J.F. Lesigne, *Extended equal area criterion revisited*, IEEE Trans. on Power Syst. **PWRS-7** (1992), 1012–1022. (194, 196, 276)
- [XU 93a] Y. Xue, P. Rousseaux, Z. Gao, L. Wehenkel, M. Pavella, R. Belhomme, E. Euxibie, and B. Heilbronn, *Dynamic extended equal area criterion - Part 1. Basic formulation*, Proc. of the Joint IEEE-NTUA International Power Conference APT, September 1993, pp. 889–895. (10, 276)
- [XU 93b] Y. Xue, Y. Zhang, Z. Gao, P. Rousseaux, L. Wehenkel, M. Pavella, M. Trotignon, A. Duchamp, and B. Heilbronn, *Dynamic extended equal area criterion - Part 2. Embedding fast valving and automatic voltage regulation*, Proc. of the Joint IEEE-NTUA International Power Conference APT, September 1993, pp. 896–900. (11, 276, 300)
- [XU 93c] Y. Xue, *An emergency control framework for transient stability of large power systems*, Proc. of the IEE Conf. on Power Systems, 1993. (194)
- [XU 93d] Y. Xue, Y. Zhang, P. Rousseaux, L. Wehenkel, M. Pavella, B. Garnier, P. Juston, J. N. Marquet, B. Meyer, M. Trotignon, *Advances in the extended equal-area criterion fast transient stability assessment*, Submitted for publication, December 1993. (300)
- [ZA 82] J. Zaborszky, K. Whang, G. M. Huang, L. Chiang, and S. Lin, *A clustered dynamic model for a class of linear autonomous systems using simple enumerative sorting*, IEEE Trans. on Circuits and Syst. **CAS-29** (1982), no. 11, 747–758. (23)
- [ZH 90] Y. S. Zhao, *Conception d'un système expert destiné à la caractérisation des états en tension des réseaux électriques*, Tech. report, EDF - DER, 1990, Final report of contract EDF/LAIH No. R46L08/1E7184. (319, 324)
- [ZH 91] X.J. Zhou and T. S. Dillon, *A statistical-heuristic feature selection criterion for decision tree induction*, IEEE Trans. Pattern Analysis and Machine Intelligence **PAMI-13** (1991), 834–841. (364)
- [ZI 92] A. Zighed, J.P. Auray, and G. Duru, *Sipina. Méthode et logiciel*, Alexandre Lacassagne - Lyon, 1992. (66, 67, 364, 395)
- [ZU 90] J. M. Zurada, *Introduction to artificial neural systems*, West Publishing, 1990. (18, 20, 156)

Index

- Accuracy, **34**
- Activation function, 137
- Artificial neural networks, 18, **133**
- Attributes, 13, **30**
 - candidate, 31
 - selected, 31
 - test, 31
- Back-propagation algorithm, 19, **142**
- Class probability tree, **51**
- Classes, **31**
- Classical model
 - for transient stability studies, 11, 261, 264
- Classification, **31**
 - rule, **33**
- Clustering, **37**
- Clustering methods, 20
- Complexity, **34**
- Conceptual clustering, 20
- Contingencies, 3
- Corrective
 - control, 3
- Correlation
 - coefficients, **38**
- Cost
 - of implementation, 34
 - of learning, 34
- Cross-validation
 - reliability estimate, **45**, 263
- Deadend : a terminal node corresponding to a pruned subtree, 89
- Decision
 - rule, **33**
 - tree, **51**
- Dendrograms, 125
- Deterministic, **32**
- Diagnostic, **32**
- Direct methods
 - for transient stability, 10, 195
- Distances, 95
 - between objects, **37**
- Disturbances, 3
- Emergency
 - control, 3
 - state detection, 3
- Entropy
 - criterion for back-propagation, 146
 - logarithmic, **41**
- Examples
 - learning and test, **34**
- Features, 13
- Histogram, **120**
 - non-parametric estimation, 16
- Hypothesis space, **33**
- Instance based learning, **93**
- ISODATA and K -means, 122
- Kernel density estimation, **119**, 298, 319
- Lateral fault, 261
- Leaf : a terminal node of sufficiently low apparent entropy, 89
- Learning
 - classes of methods, 15

- supervised, 13
 - unsupervised, 20
- Leave-one-out
 - reliability estimate, **46**, 263
- Linear discriminant
 - Fisher, 105, 141, 298, 319
 - generalized, 111
 - logistic, 108, 298, 319
 - perceptron, 135
- Load
 - dynamics, 6
 - voltage sensitivity, 6
- Lyapunov methods, 10, 195
- Machine learning
 - class of computer based learning methods, 47
 - framework for security assessment, 1
 - methods, 16
- Mode of instability, 10
- Monitoring, 8
- Multi-layer perceptrons, **141**
- Naive Bayes classification method, 121
- Nearest neighbor method, **112**, 171, 173, 298, 317, 319
- Non-deterministic, **32**
- On-line
 - operation, 8
- Operation
 - on-line, 8
- Operation planning, 7
- Overfitting, **58**
 - and tree pruning, 57
 - in Bayesian clustering, 127
 - in decision trees, 56
 - in histograms, 120
 - in kernel density estimation, 119
 - in multi-layer perceptrons, 146
 - in nearest neighbor classifier, 113
- Partitioning
 - tree, **50**
- Pattern recognition, 15
- Perceptron, 18, **135**
- Planning
 - operation, 7
 - system, 7
- Prediction, **32**
- Preventive
 - control, 3
 - security assessment, 3
- Prototypes, 20
- Quality measure, **33**
- Real-time, 6
 - monitoring, 8
- Regression, **35**
 - models, **36**
 - tree, **51**
- Reliability, **34**, 42
- Restoration, 4
- Resubstitution
 - reliability estimate, **45**, 262
- Security, 2
 - steady state, 6
 - voltage, 5, **196**
- Similarity
 - of attributes, **38**
- Stability
 - transient, 5, **191**
 - voltage, 5, **196**
- Static security
 - tools, 12
- Steady state security, 6
- Synchronism
 - loss of, 5
- System planning, 7
- Test set
 - reliability estimate, **45**
- Training, 9
- Transient stability, 5, **191**
 - tools, 9

Tree, **49**

growing, 56

pruning, 56

Universe of possible objects, **30**

Voltage stability, 5, **196**

tools, 11

Glossary

- $3\phi SC$: three-phase short-circuit, 261
 H_m : maximal value of the entropy of a leaf, 89
 $K-NN$: nearest neighbor method, 16, 112, 171, 173, 298, 317, 319
 M : number of test states, 35
 N : number of learning states, 34
 P_e : error probability (and its test set estimate), 43
 α : risk of not detecting the statistical independence in stop-splitting stop-splitting, 59
 \inf : lower bound of a set of numbers, 38
 τ :
 for transient stability, CCT threshold or fault clearing time, 285
 or voltage security, post-disturbance time of the JAD state, 326
 m : number of classes, 31
 n : number of attributes, 33
 r : number of regression variables, 36
 \mathcal{N}_i : an *interior* or test node, 49
 \mathcal{N}_t : a *terminal* tree node, 49
 \mathcal{N} : a tree node, 49
 $\#\mathcal{N}$: total number of nodes of a tree, 73
 LS : learning set, 34
 PS : pruning set, 60
 TS : test set, 35
 U : universe of possible objects, 30
 δ_{ij} : (Kronecker) δ_{ij} if $i = j$ and 0 if $i \neq j$, 42
#A : number of different DT test attributes, 265
AC : alternating current, 12
ANN : artificial neural networks, 18
AutoClass : clustering method, 126
AVR : automatic voltage regulator of synchronous machines, 5
BFGS : Broyden-Fletcher-Goldfarb-Shannon algorithm, 148, **148**, 149
CCT : critical clearing time, **193**
CPU : central processing unit of a computer, 10
DC : direct current, 5
DE : dangerous errors, 285
DT-ANN : Decision Tree - Artificial Neural Network, **169**, 225, 232
DTSA : decision tree based security assessment, 189
DTTS : decision tree based transient stability assessment, 269, 285
DT : decision tree, 17
EHV : extra high voltage, 9
EMS : energy management system, 6
FACTS : flexible alternative current transmission systems, 5
FA : false alarms, 285
HV : high voltage, 199
IBL : instance based learning, 48, 93
JAD : just after disturbance state, 23

- LTU : linear threshold unit, 18, **135**
- LVQ : learning vector quantization, **163**, 298, 319
- MLP : multi-layer perceptrons, 19, **141**, 298, 316, 319
- MSE :
- generalization, 146
 - mean square error function, 19
 - perceptron, 137
 - projection pursuit, 116
 - regularization, 146
- MV : medium voltage, 199
- ND : non-detections, 285
- NE : normal errors, 285
- OLTC : on load tap changer, 5
- OMIB : One-Machine-Infinite-Bus system, 260
- OPF : optimal power flow, 6
- OS : operating state, 274
- PI : performance index for contingency filtering, 12
- SBS : step-by-step time-domain simulation method, 9
- SMART : projection pursuit method, 118, 298, 319
- SMES : superconducting magnetic energy storage devices, 5
- SOM : self organizing feature map, **156**, 298, 319
- SVC : static var compensator, 5
- TDIDT : top down induction of decision trees, 17, 55
- TSA : transient stability assessment, 5
- UHV : ultra high voltage, 301
- ULTC : under load tap changer, 5
- VSA : voltage security assessment, 11