

FEATURE SELECTION IN THE CLASSIFICATION  
OF TIME-VARIANT PATTERNS

Khalid J. Siddiqui

A Thesis  
in  
The Department  
of  
Computer Science

Presented in Partial Fulfillment of the Requirements  
for the Degree of Doctor of Philosophy at  
Concordia University  
Montreal, Quebec, Canada

May 1994

© Khalid J. Siddiqui, 1994

## ABSTRACT

### Feature Selection in the Classification of Time-Variant Patterns

Khalid J. Siddiqui, Ph.D.  
Concordia University, 1994

A bottleneck in building and using the knowledge base in an intelligent system is combining the appropriate problem solving knowledge with physical observations. Another problem is to derive pertinent information that is subtly available in physical observations. These problems are resolved by using the information and knowledge processing techniques available in the fields of signal processing, pattern recognition and knowledge engineering. Methods are developed to automatically measure, recognize and interpret the parameters (features) from the physical observations. A Successive Feature Elimination Scheme involving multiple steps is developed to eliminate poorly performing features. Pseudo-Similarity method which uses inter-class dissimilarity is introduced for feature ranking. To minimize the problems of information explosion and redundancy the concept of Pattern Association Hierarchy (PAH) is introduced to structure and organize the features and pattern classes in the form of a knowledge tree. Several classifiers including the two new algorithms PAH classifier and entropy based decision tree classifier are also developed. Based on the nature of the training data a number of meta rules are developed to select the best knowledge organization and classification algorithms. All these components and concepts are used as a basis to propose a structure of an intelligent waveform recognition system. This unified approach will not only automate and accelerate the knowledge acquisition and organization process, but will also formalize and structure the decision-making process, and thus reduce the reliance on a human expert.

The performance of these components is successfully demonstrated on several time-variant signals from non-destructive testing (NDT), and non-invasive testing (NIT) generated from materials (NDT signals), chemical mixtures (PNA spectra), human brain (EEG signals), and genetic cells (CEL signals). On the testing set from NDT data with 10 classes an overall performance reaching 84% was achieved and up to 95% when it is treated as a four class problem. Up to 95.67% of the EEG signals with 3 classes were correctly recognized whereas a perfect score of 100% was obtained on PNA data with 20 classes. On the CEL data with 19 classes the recognition performance reached 88.34%.

### Acknowledgements

I am grateful to the almighty Allah for granting me the wisdom curiosity, dedication, and endurance in pursuit of the goals I set.

I am sincerely thankful to my advisors Drs. D. Robert Hay and Ching Y. Suen for their continuous support, encouragement and guidance throughout the span of this research. Their patience and understanding, both work-related and personal have been most valuable for the completion of this endeavor.

I would also like to thank Dr. D. Eastwood of Lockheed Engineering and Sciences Company, Las Vegas, for providing knowledge and sharing thoughts pertaining to Chemistry and reading the sections which are carrying the information about the PNA problem. Thanks are also due to Dr. D.J. MacCrimmon, McMaster University, Hamilton, Canada for providing data on the EEG problem. The help that Dr. Y.-H. Liu, University of Nebraska, Omaha has provided in formulating the feature optimization problem is unforgettable. Some typesetting help received from Iftikhar is gratefully acknowledged. Akhtar's help in making few diagrams is appreciated as well. The software development, mainly, was done using Borland's Turbo Pascal and C in DOS and UNIX environments, the wordprocessing was done using Wordperfect, and the diagrams were drawn using Harvard Graphics. These vendors are sincerely acknowledged. The computing facilities at Concordia, Creighton, and at the University of Nebraska are appreciated very much. The office space and supporting facilities provided by Drs. M. Sketch, and S. Mohiuddin of The Cardiac Center, Creighton University, are acknowledged as well.

The comments, suggestions and the supply of NDT related literature by Dr. J.R. Matthews, Defence Research Establishment



Atlantic, Dartmouth, Canada, and Dr. A. Fahr, National Research Council, Ottawa, Canada are appreciated very much.

During my long period of studies at Concordia, I received much help from Stephanie, Halina, Angie, and Irene, the secretaries of the Computer Science department. Helene was very helpful in looking after the administrative matters at the Dean's office. I wish to thank them very sincerely. Mrs. C. Hay, Maureen, and other staff at Tektrend are also acknowledged for their share of support and well wishes.

I wish to express my gratitude to my peers, colleagues and dear friends in Montreal and Omaha, in particular, Drs. T. Fancott, R. Shinghal, S. Mohiuddin, D. Malik, and Mr. B. Qureshi for their encouragement, moral support in times of acute uncertainties and setbacks. My life-long friends, Alam in Chicago, Anzar and Ijaz in Montreal, and, Ismail, and Tahsin in Toronto provided transitory stay several times during these years of travelling; their hospitality and kindness is unforgettable.

Last, but not least I wish to thank my parents, siblings, and family for giving me support and kept praying for my welfare. I am particularly thankful to my wife and children for bearing with me during all these years of pain and miseries. Their understanding during eight years of shuttling between Omaha and Montreal is appreciated very much. The amount of time and money spent on this entire exercise is phenomenal. None other than my family deserves the acknowledgement, as often they sacrificed the acute necessities for making time and funds available.

Alas! the remaining errors, omissions, ambiguities, and misleading comments, if any, are the sole responsibility of the author.

Dedicated to my  
Homeland and my  
Family

## **Table of Contents**

Abstract	iii
Acknowledgement	v
Dedications	vii
Table of Contents	viii
Lists of Figures	xiv
List of Tables	xvi
List of Notations and Symbols	xxi
Glossary of Terms	xxiv

### **Chapter One**

#### **Introduction**

1.1 Intelligent Waveform Sensing .....	1
1.2 Signal and a Signal Processing System .....	5
1.3 Knowledge Based Systems .....	10
1.4 NDT Waveform Processing Systems .....	13
1.5 Medical Diagnostic Systems (MDS) .....	14
1.6 The Need for a Stand-alone System .....	18
1.7 Integration of Information Processing Technologies .....	20
1.8 A General Intelligent Recognition System (IRS)....	22
1.9 The Research Contributions .....	27
1.10 Thesis Organization .....	30

### **Chapter Two**

#### **Common Elements of an Intelligent Recognition System**

2.1 Introduction .....	32
2.2 Signal Interpretation - Current Practices .....	32

2.3	Modeling the Waveform Indications .....	34
2.3.1	NDT Indications Model .....	35
2.3.2	EEG Indications Model .....	38
2.3.3	PNA Indications Model .....	40
2.4	Ideal Knowledge Requirements .....	41
2.5	The Knowledge - Our Perspective .....	48
2.6	The Design of IRS System .....	51
2.6.1	The Knowledge Acquisition, Representation and Organization (KARO) Subsystem .....	52
2.6.2	Inference Engine .....	54
2.7	Signal Conditioning and Treatment .....	54
2.8	Mapping and Parameterization of Waveforms .....	56
2.8.1	Mapping Space .....	56
2.8.2	Parameter Extraction .....	60

### **Chapter Three**

#### **Analytical Features and Pattern Association Hierarchy**

3.1	Introduction .....	62
3.2	Fact Gathering Phase .....	62
3.2.1	The Pattern Data .....	63
3.2.2	Pattern Measurement Problems .....	63
3.2.2.1	Analytical Feature Extraction ....	66
3.2.2.2	Homogenizing the Analytical Features .....	67
3.3	Pattern Association Hierarchy .....	72
3.4	Clustering Procedures .....	77
3.4.1	The Bottom-up Organization .....	78
3.4.2	Generalized Variations Clustering Procedure	84
3.4.3	The Top-down (Divisive) Organization .....	87
3.4.4	Clustering Algorithm Selection Criterion - Meta Knowledge .....	89
3.5	Knowledge Organization Strategy .....	91

## **Chapter Four**

### **Feature Selection, Empirical Knowledge, and Organization of Knowledge Base**

4.1	Introduction .....	95
4.2	Selection of Optimal Features (Analytical) .....	95
4.2.1	The Size Selection .....	96
4.2.2	Feature (Label) Selection .....	97
4.2.2.1	Successive Elimination Process ...	98
4.2.2.2	Back-end Feature Dimensionality Reduction .....	103
4.3	Optimal Feature Selection - One by One Criterion..	103
4.4	Optimal Feature Selection - Simultaneous Selection Criterion .....	105
4.4.1	Pseudo-Similarity Algorithm .....	106
4.5	Weight Allocation to Features .....	111
4.6	Empirical Knowledge .....	113
4.7	Knowledge Formalization, Representation and Organization .....	116
4.7.1	Knowledge Formalization and Representation.	116
4.7.2	Hierarchical Knowledge Organization .....	117

## **Chapter Five**

### **Inference Engine and Machine Learning**

5.1	Introduction .....	120
5.2	Components of the Inference Engine .....	121
5.3	Inference Mechanism .....	123
5.4	Machine Learning .....	126
5.4.1	Learning by the Discrimination System .....	127
5.4.2	Learning by the Cognition System .....	131
5.5	Discrimination System - The Process .....	133
5.6	Cognition System - The Process .....	134
5.7	Entropy-based Decision Tree (EDT) Algorithm .....	135

5.7.1	Design of the EDT Algorithm .....	137
5.7.2	Computational Complexity and Problems .....	142
5.7.3	Merits of the EDT - Classifier .....	143
5.8	Failure Control .....	144

## Chapter Six

### Discrimination Subsystem

6.1	Introduction .....	146
6.2	Discrimination Subsystem .....	146
6.3	Classification Methodologies .....	147
6.3.1	Decision Theoretic Approaches .....	148
6.3.1.1	Parametric Approaches .....	148
6.3.1.2	Non-Parametric Approaches .....	150
6.3.1.2.1	Direct Decision Functions .....	150
6.3.1.2.2	Adaptive Decision Functions .....	151
6.3.2	Information - Theoretic Methods .....	154
6.3.3	Syntactic Approaches .....	154
6.3.4	Other Decision Making Strategies .....	155
6.3.4.1	Graph Theoretic Approaches .....	155
6.3.4.2	Heuristic Approaches .....	157
6.4	Trends in Decision Making Process .....	157
6.5	Classification (Search) Strategies .....	158
6.6	Hierarchical Decision Approaches .....	160
6.6.1	The PAH - Classifier .....	161
6.6.2	Design of the PAH - Classifier .....	162
6.6.3	Computational Complexity .....	164
6.6.4	Merits of the PAH - Classifier .....	168
6.7	Nodal Classifiers .....	169
6.7.1	Empirical Bayesian Classifier .....	169
6.7.2	K-Nearest Neighbor Classifier .....	172
6.7.3	Minimum Distance Classifier .....	173
6.7.4	Linear Discriminant Classifier .....	175
6.7.5	Quadratic Discriminant Classifier .....	176
6.8	Classification Process .....	178
6.9	Parametric Selection of a Classifier .....	179

## Chapter Seven

### Classification Experiments and Results

7.1	Introduction .....	188
7.2	The Functional View of Recognition Components ....	189
7.2.1	The Function of The Discrimination Sub-system .....	190
7.2.2	The Function of The Cognition Subsystem ...	192
7.2.3	The Function of The Failure Control Sub-system .....	193
7.3	System's Training .....	193
7.4	Performance of the Recognition Components .....	194
7.4.1	Implementation of MDC .....	194
7.4.2	Implementation of KNN .....	195
7.4.3	Implementation of LDC .....	195
7.4.4	Implementation of QDC .....	196
7.4.5	Implementation of BYC .....	196
7.4.6	Implementation of PAH .....	197
7.4.7	Implementation of EDT .....	198
7.5	Performance on NDT Data .....	198
7.5.1	Experiment A - MDC .....	200
7.5.2	Experiment B - KNN .....	204
7.5.3	Experiment C - LDC .....	206
7.5.4	Experiment D - QDC .....	206
7.5.5	Experiment E - BYC .....	209
7.5.6	Experiment F - PAH .....	210
7.5.7	Experiment G - EDT .....	212
7.5.8	Comments: Performance on NDT Data.....	214
7.6	Performance on EEG Data .....	218
7.6.1	Experiment A - MDC .....	222
7.6.2	Experiment B - KNN .....	229
7.6.3	Experiment C - LDC .....	232
7.6.4	Experiment D - QDC .....	233
7.6.5	Experiment E - BYC .....	236
7.6.6	Experiment F - PAH .....	237
7.6.7	Comments: Performance on EEG Data.....	239
7.7	Performance on PNA Data .....	242
7.8	Performance on CEL Data .....	248
7.8.1	Experiment A - MDC .....	253

7.8.2	Experiment B - QDC .....	258
7.8.3	Comments: Performance on CEL Data.....	262
7.9	Discussion: Overall Recognition Performance .....	263

## Chapter Eight

### Performance Review, Directions for Further Research, and Conclusions

8.1	Introduction .....	268
8.2	Performance Review .....	268
8.2.1	System Concept Level .....	269
8.2.2	Feature Extraction and Selection Level .....	269
8.2.3	Knowledge and Knowledge Representation .....	270
8.2.4	Knowledge Organization Level .....	270
8.2.5	Classification Level .....	271
8.2.6	Integration and Automation Level .....	272
8.2.7	Application Level .....	273
8.2.8	Overall Efficiency and Effectiveness .....	273
8.2.9	Expansion and Growth .....	274
8.3	Directions for Further Research .....	274
8.3.1	Future: Knowledge Acquisition .....	275
8.3.2	Future: Knowledge Formalization and Organization .....	275
8.3.3	Future: Modeling the Pattern Classes .....	276
8.3.4	Future: Inference Engine .....	277
8.3.5	Future: Expert/User Interface .....	277
8.4	The Research Contributions .....	278
8.5	Conclusions .....	280
8.5.1	NDT Problem .....	281
8.5.2	Medical Diagnosis Problem .....	282
8.5.3	Exploration/Classification of Oil and Minerals .....	284
8.5.4	General Remarks .....	284
	References .....	287
	Appendix - A Feature Extraction Details .....	311
	Appendix - B Acquisition and Characteristics of Data Sets .....	324



## List of Figures

1.1	A General Signal Acquisition and Processing System ..	6
1.2	A Typical ECG Waveform and its Information Contents .	8
1.3	A General Signal Classification/Diagnosis System ....	9
1.4	Components of an Expert System .....	11
1.5	The Proposed design of an Intelligent Recognition System .....	23
1.6	Skeletal View of components and techniques used in Intelligent Recognition System .....	28
2.1	NDT Indications .....	37
2.2	EEG Indications .....	39
2.3	PNA Indications .....	42
2.4	The Knowledge Acquisition, Representation, and Orga- nization (KARO) Subsystem .....	53
2.5	Pattern Measurement System [SIDD-90a] .....	57
2.6	Geometrical Interpretation of Envelope and Features Derived Therefrom .....	61
3.1	A few typical samples from NDT signals .....	65
3.2	Hierarchical Clustering Procedure .....	81
3.3	Block Diagram of Tree Organization .....	82
3.4	Bottom-up Algorithm for organizing the analytic feat- ures and corresponding classes in a tree structure .	83
3.5	Lower triangle .....	84
3.6	Top-down Algorithm for organizing the analytic feat- ures and corresponding classes in a tree structure .	90
3.7	Decision Tree for the Selection of a Clustering Procedure .....	94
4.1	A typical organization of knowledge frame .....	119
5.1	Schematic Design of the Inference Engine .....	122

### List of Figures (Contd.)

5.2	Inference Mechanism and Control Strategy .....	124
5.3	Supervised Learning and procedure for the Inference Tree .....	129
5.4	Nodal Training Scheme - Discrimination System .....	132
5.5	Tree Classification Mechanism of the Discrimination System .....	135
5.6	Information-theoretic organization of knowledge and pattern classes .....	136
5.7	Procedural steps of EDT Algorithm .....	138
6.1	PAH Classifier Design .....	161
7.1	Variations between different NDT Pattern Classes ..	199
7.2	A few samples from EEG Signals .....	219
7.3	Recognition Performance vs Number of Features .....	226
7.4	A few sample spectra of Petroleum Oils (PNA's) ....	243
7.5	Schematic structure of a nerve cell .....	250
7.6	The time course of action potentials .....	250

## List of Tables

2.1	Popular NDT Methods, their Application and useful Method Dependent Parameters .....	45
2.2	Theme Rules Describing Problem Domain Information ...	46
2.3	A priori knowledge - Test Specimen .....	47
2.4	Shape Factors of Some Common Defects .....	47
2.5	Test Apparatus & Test Conditions .....	48
2.6	Analytical knowledge features used .....	61
3.1	Sizes of Defect Areas and Their Identification .....	64
3.2	Analytical Features extracted for NDT Signal data [HAYD-88] .....	68
3.3	Parametric values for different clustering algorithms	78
3.4	Pattern Association Hierarchy using Single Linkage Method.....	80
3.5	Pattern Association Hierarchy using Centroid Method .	80
3.6	Pattern Association Hierarchy using Group Average Method.....	81
3.7	Pattern Association Hierarchy using Generalized Variations Procedure .....	87
3.8	Rule Set for the Selection of Clustering Procedure and Similarity Index .....	92
4.1	Stationary Features (NDT-data) removed .....	99
4.2	Features (NDT-data) rejected by t-test .....	101
4.3	Features (NDT-data) deleted by Collinearity test ....	101
4.4	Features (NDT-data) merged by 2nd Collinearity test .	103
4.5	Features (NDT-data) ranked using Fisher Index .....	105
4.6	Features (NDT-data) ranked using Pseudo-Similarity Algorithm .....	111
4.7	Empirical and Statistical Decision Parameters .....	114

## List of Tables (Contd.)

6.1	Complexity of Unweighed Pattern Classifiers .....	167
6.2	Properties of Classification Algorithms .....	184
6.3	Set of Rules used for the selection of Classification Procedures and Feature Weighing Criterion .....	185
7.5.A1	Classification Results on NDT Data Using Linear Organization of Pattern Classes (MDC-Euclidean, Feat-A)	202
7.5.A2	Classification Results on NDT Data Using Linear Organization of Pattern Classes (MDC-Mahalanobis, Feat-A)	202
7.5.A3	Classification Results on NDT Data Using Linear Organization of Pattern Classes (MDC-Euclidean, Feat-F)	203
7.5.A4	Classification Results on NDT Data Using Linear Organization of Pattern Classes (MDC-Euclidean, Feat-S)	203
7.5.A5	Classification Results on NDT Data Using Linear Organization of Pattern Classes (MDC-Mahalanobis, Feat-F)	204
7.5.B1	Classification Results on NDT Data Using Linear Organization of Pattern Classes (3NN-Euclidean, Feat-A)	205
7.5.B2	Classification Results on NDT Data Using Linear Organization of Pattern Classes (3NN-Mahalanobis, Feat-A)	205
7.5.D1	Classification Results on NDT Data Using Linear Organization of Pattern Classes (QDC, Feat-A) .....	207
7.5.D2	Classification Results on NDT Data Using Linear Organization of Pattern Classes (QDC, Feat-A, wt=1/sd) ..	208
7.5.D3	Classification Results on NDT Data Using Linear Organization of Pattern Classes (QDC, Feat-F) .....	208
7.5.D4	Classification Results on NDT Data Using Linear Organization of Pattern Classes (QDC, Feat-S) .....	209
7.5.E1	Classification Results on NDT Data Using Linear Organization of Pattern Classes (BYC, Feat-A) .....	210
7.5.1	Hierarchical Organization of Pattern Classes .....	211
7.5.F1	Classification Results on NDT Data Using Hierarchical Organization of Pattern Classes (MDC-Mahalanobis) 10 Class Problem .....	212

## List of Tables (Contd.)

7.5.F2	Classification Results on NDT Data Using Hierarchical Organization of Pattern Classes (MDC-Mahalanobis) 4 Class Problem .....	212
7.5.G1	Classification Results on NDT Data Using EDT Algorithm (10 Class Problem) .....	213
7.5.G2	Classification Results on NDT Data Using EDT Algorithm (4 Class Problem) .....	214
7.6.1	Features used for EEG Problem .....	220
7.6.2	EEG Problem: Features Deleted .....	221
7.6.3	EEG Problem: Feat-F and Feat-S .....	222
7.6.A1	EEG Problem: Design Set, Classification Results (MDC-Euclidean, Feat-A) .....	224
7.6.A2	EEG Problem: Testing Set, Classification Results (MDC-Euclidean, Feat-A) .....	225
7.6.A3	EEG Problem: Design/Testing Set, Classification Results (MDC-M, Feat-A) .....	226
7.6.A4	EEG Problem: Testing Set, Classification Results (MDC-Euclidean, Feat-F) .....	227
7.6.A5	EEG Problem: Testing Set, Classification Results (MDC-Euclidean, Feat-S) .....	228
7.6.B1	Classification Results on EEG Data Using Linear Organization of Pattern Classes (3NN-Euclidean, Feat-F) Design Set .....	230
7.6.B2	Classification Results on EEG Data Using Linear Organization of Pattern Classes (3NN-Euclidean, Feat-F) Testing Set .....	231
7.6.B3	Classification Results on EEG Data Using Linear Organization of Pattern Classes (3NN-Mahalanobis, Feat-A) .....	232
7.6.C1	EEG Problem: Classification Results -LDC (Feat-A) ..	233
7.6.D1	Classification Results on EEG Data Using Linear Organization of Pattern Classes (QDC, Feat-A) .....	235
7.6.D2	Classification Results on EEG Data Using Linear Organization of Pattern Classes (QDC, Feat-F) .....	235

### List of Tables (Contd.)

7.6.D3 Classification Results on EEG Data Using Linear Organization of Pattern Classes (QDC, Feat-S) .....	236
7.6.E1 Classification Results on EEG Data Using Linear Organization of Pattern Classes (BYC, Feat-S) .....	237
7.6.4 Hierarchical Organization of Pattern Classes .....	238
7.6.F1 Classification Results on NDT Data Using Hierarchical Organization of Pattern Classes (MDC-Mahalanobis) .	238
7.6.F2 Classification Results on NDT Data Using Hierarchical Organization of Pattern Classes (MDC-Euclidean) Feature Sets: Feat-F & Feat-S .....	239
7.7.1 PNA Pattern Classes, their Labels .....	245
7.7.2 PNA Problem - Features Extracted .....	246
7.7.3 Pattern Association Hierarchy using Centroid Method	246
7.7.4 Classification Results .....	247
7.8.1 CEL Data Pattern Classes .....	251
7.8.2 CEL Problem - Features Deleted .....	252
7.8.3 EEG Problem: Feat-F and Feat-S .....	253
7.8.A1 CEL Problem: Design Set, Classification Results (MDC-Euclidean, Feat-A) .....	254
7.8.A2 CEL Problem: Design Set, Classification Results (MDC-Euclidean, Feat-A, wt = 1/sd) .....	255
7.8.A3 CEL Problem: Testing Set, Classification Results (MDC-E, Feat-A, wt = 1/sd) .....	256
7.8.A4 CEL Problem: Testing Set, Classification Results (MDC-M, Feat-A) .....	257
7.8.B1 CEL Problem: Testing Set, Classification Results (QDC, Feat-A) .....	259
7.8.B2 CEL Problem: Testing Set, Classification Results (QDC, Feat-F) .....	260

## List of Tables (Contd.)

7.8.B3 CEL Problem: Testing Set, Classification Results (QDC, Feat-S) .....	261
7.9.1 Sensitivity of Classifiers on Different Data Sets .	267

## List of Notations and Symbols

$\{ \}$	: Set symbol
$[ ]$	: Vector symbol
$  \dots  $	: Determinant
$\neg$	: Logical negation
$\in$	: Inclusion of an element
$\alpha$	: Level of significance for t-test
$\varsigma$	: Correction factor to adjust the feature's weight
$\sigma_{ij}$	: Covariance of features $i$ and $j$
$\mathfrak{S}$	: Feature Subset
$\psi$	: Ratio of pattern space to no. of features
$\Phi_j(X)$	: Characteristic function on a vector $X$
$U_x$	: Graph of pattern $X$
$a, b, c$	: Parameters of recursive distance function
$cor$	: Correlation threshold
$C$	: Label of a Pattern Class
$d(q, r)$	: Distance between classes (or groups) $q$ and $r$
$\bar{D}$	: Mean difference
$D_{ij}$	: Difference between feature
$D_j(X)$	: Discriminant function
$\delta$	: Expected value of feature difference
$G$	: Total number of cluster groups
$G_o, G_q, G_r$	: Group of Pattern Classes
$h, g, \text{ and } f$	: Three functions defining clustering environment: $h$ : maps input as vector $g$ : symbolic label of a class $f$ : clustering procedure
$H$	: Entropy function
$i, j, k$	: Indices for a feature, a sample, or a classes
$I_n$	: Entropy of input group
$I_o$	: Entropy of output group
$J$	: Optimization Criterion
$L$	: Level of the tree
$\ln$	: Natural logarithm
$M_i$	: Mean vector of class $i$
$M_o, M$	: Mean over $N$ classes
$m_i$	: Mean of feature $i$
$m_i$	: Mean of feature $i$ in reduced feature set
$MAT(ijr)$	: An element of the training matrix representing the conditional probabilities



Max	: A function to find maximum
Min	: A function to find minimum
$n'', n', n$	: Number of features at various stages of analysis
N	: Number of pattern classes
$P_i$	: Number of samples in class i
$P_o$	: Total number of samples in a problem space
$\rho_{ij}$	: Cell ij of the similarity matrix R
P	: Number of samples (equal) in each class
$P(i)$	: Probability of occurrence of class i
$P(\omega_j)$	: A priori probability of occurrence of class $\omega_j$
$P(X \omega_j)$	: Conditional probability of occurrence of X given that $\omega_j$ has occurred
r	: evaluated decision plane (function value) of a classification method
R	: Similarity matrix
$s^2$	: Pooled variance
$S_j$	: Standard deviation of feature j
S	: Variance-Covariance Matrix or pooled matrix
$S_w, S_b, S_t$	: Scatter matrices - within, between, and total
$S(q)$	: Number of comparison for a successful search in a binary tree
$U(q)$	: Number of comparison for an unsuccessful search in a binary tree
$\tau$	: Classification method dependent decision plane (threshold)
t	: Variable that holds the Student's t Distribution
$\tau'$	: range of feature values
$T^2$	: Distance (Hotelling's $T^2$ )
$t^2$	: Squared distance from the mean vector
$t'_\alpha$	: Tabulated value of t at significance level $\alpha$
$\theta_o, \theta_1, \theta_2, \theta_3$	: Threshold used successively to eliminate the features
$\theta_m$	: Majority vote threshold
$\theta_c, \theta_{min-c}$	: Threshold used for correlation
U	: Universe of a problem space
$v_i$	: Variance of feature i
$v'$	: Value of a feature
$\omega$	: A general pattern class
$w_i$	: Weight allocation to feature i
$W, W'$	: Weight vectors

$x_{ij}$	: Feature j of class i (transformed)
$x_{ij}'$	: Original value of feature j of class i
$x_{j^0}$	: j-th region of feature $x_j$
$x_{min}$	: Minimum value of $x_{ij}'$
$x_{max}$	: Maximum value of $x_{ij}'$
$\bar{x}$	: Maximum likelihood estimate for mean of a feature
$X$	: An arbitrary feature vector
$y_i$	: Measurement (feature) on a pattern
$Z$	: Design set

## Glossary of Terms

Analytical Knowledge	A set a parameters analytically derived from physical observations.
BYC	The Bayesian Probabilistic Discriminant Function/ Classifier
Class	One of many groups or divisions of a problem space
Classify	Arrange or sort in classes (during design phase), or assign to a class (during classification)
Classification	A procedure of deciding if a given input pattern belongs to a given class of patterns
Cluster	A group of patterns or feature vectors closely associated based on a given criterion
CEL	Cell Data
Clustering	The process of grouping data
Complex Heuristics	A composite of data, information, knowledge, and meta knowledge allowing or assisting to discover
Decision	A conclusion reached as to some procedural action to be taken in a pattern analysis system
Decision function	A scalar single-valued function that defines the statistical decision boundary
Decision boundary (surface)	An artificially created border between two or more classes or clusters
Deduction	A term used in the sense of classification
Decision theoretic	Mathematical theory of making optimal decision
Discrimination	The ability to distinguish between two or classes (or clusters) of patterns
Distribution	The arrangement of data in a feature set, both within class and between class

ECG	Electrocardiogram
EDT	Entropy based Decision Tree algorithm
EEG	Electroencephalogram
Empirical Knowledge	A set a parameters derived from analytical knowledge
Fact	Established knowledge about a problem domain
Feature	A distinctive characteristic measurement, or a structural component made on a pattern
Feature Extraction	The determination of a feature or feature vector from a pattern
Feature Selection	The selection of a smaller feature set to be used for classification, may also involve mapping the original pattern to a lower dimensional pattern
Feature space	For n features, the n-dimensional space
Heuristics	Problem solving knowledge explicitly derived from expert(s)
Homostat	A homogeneous pattern class or a cluster
Hybrid Machine	An embedded system that combines physical observations, rules of thumb, and heuristics
Indication	A signal pattern representing certain class
Induction	Inference of a general law from particular instances
Inference	The act of decision making - to deduce or conclude from facts
Information	The possible features a pattern may have
Interpret	Bring out the meaning of; explain or understand
KNN	K-nearest neighbor classifier

Knowledge Based Pattern Discrimination	A new problem solving approach that combines physical observations and heuristics to distinguish between patterns of different classes
Hierarchical Clustering	Process of hierarchically grouping patterns/classes
Learning/Training	Knowledge acquired by study; to gain knowledge by experience
LDC	Linear Discriminant Classifier
Likelihood estimates	Empirical probabilities of the parameters
Matching	To test the similarity of two or more entities in some essential respect
MDC-E	Minimum Distance Classifier - Euclidean
MDC-M	Minimum Distance Classifier - Mahalanobis
Meta Knowledge	A set of Rules derived from established theory of Statistics, Clustering methods and data characteristics
NDT	Non Destructive Testing
PAH	Pattern Association Hierarchy: A new concept proposed in [SIDD-93a] that organizes the knowledge components based on their proximity
PNA	Polynuclear Aromatic Hydrocarbons (Petroleum Oils)
Pattern	A numerical vector of parameters describing an event
Pattern Recognition	A process leading to, 1) classification, 2) detection, or 3) parameter estimation, of an input pattern
Piecewise linear boundary	Boundary constructed to approximate non-linear decision boundary
PR	Pattern Recognition
Proximity	A numerical measure of establishing association between two pattern vectors
QDC	Quadratic Discriminant Classifier

Sample	A single measurement of a pattern or signal
Segregates	Pattern class or cluster with significant intra-class/cluster variations
Signal	A function of one or more independent variables
Supervised	To oversee the execution of a machine pattern recognition task by providing labeled reference pattern vectors to train the system
Trainable Machine	An intelligent machine that is capable of learning as more knowledge becomes available
Unsupervised	Not supervised, or learning as more knowledge is gained

## Chapter 1

### INTRODUCTION

#### 1.1 Intelligent Waveform Sensing

A waveform which may be time-variant or time invariant, is an electronic signal and it is used to convey information in a wide range of sensing and recognition applications. In these applications a signal is generated by a physical, chemical, or a biological phenomenon and its form is governed by inherent characteristics of some phenomenon such as electrical activity of brain cells represented as EEG's. The waveform therefore carries information about the structure or the functioning of the source and/or about the path from the source to the receiver. In time-variant signals the waveform may change its structure over time whereas the time has no bearing in time-invariant waveforms. The information carried by the waveforms may be extracted directly, by known observations of the waveforms as usually done in manual inspection techniques; or indirectly, by applying appropriate synthesizing and analyzing tools to either the waveforms or the measurements derived from them as usually done in automatic signal processing systems.

A number of applications, particularly for time-variant signals from both physical and biological systems can be cited in this regard [CHEN-82, COHE-86a,b, NAGA-91]. Notable examples from the physical systems are non-destructive testing (NDT) or evaluation (NDE), and spectral analysis of polynuclear aromatic (PNA) compounds (petroleum oils). The applications from the biological systems are too many to enumerate. However, some of the examples of such systems include electrographs (EG's), such as Electrocardiographs (ECGs), Electroencephalographs (EEG's), and cell and tissue categorization.

It is obvious that the information carried by the signals is phenomenal and an ideal system for processing and interpretation of signals would include a comprehensive cause and effect analysis of all attributes of the signals. Different approaches and issues involved in building such systems are discussed in subsequent sections. A formal approach would be to model the signals using all measurable attributes comprising the source and phenomenon generating the signals, their structure, and variations between signals.

Although numerous attempts were made to exhaustively synthesize the structure of the signals using analytical and mathematical tools [KRAU-69, NJP-93, STAL-82, VARY-79], such a comprehensive model cannot be represented using a mathematical expression alone. Instead we need methods to determine and evaluate the information signals carry, and to process large volumes of information we need automated knowledge based tools. Emphasizing such need we propose, in this thesis, a system approach to solve the problems of this magnitude and developed several essential tools for information analysis and summarization, and organization and categorization to achieve the objectives. Modeling the information contents carried by the signals and utilizing methods and concepts from pattern recognition, statistical decision theory and knowledge engineering, we developed algorithms to, 1) eliminate redundant information, 2) rank and weigh the parameters based on their discrimination power and information contents, 3) structure and organize the knowledge, based on natural association among pattern classes, and 4) categorize and interpret the signal patterns using a suitable classification algorithm. In addition, we developed rules to select different algorithms and made efforts to minimize the subjective biases of an expert/user at each phase of processing. A number of existing tools and techniques, such as clustering algorithms, Fisher's feature ranking method, and several traditional classification



algorithms, wherever they found feasible, are used. However, appropriate adjustments, as found necessary, have been made in these tools and techniques to conform with the objectives. Wherever it become necessary, new methods and techniques are developed (see Section 1.9). These methods mainly include, feature elimination and ranking methods, a pattern organization scheme, and several classification methods. All these methods are essentially the contributions of this research.

The performance of all these tools and techniques are evaluated in two domains of applications, i.e., physical systems and biological systems. Two examples from the physical systems considered are non-destructive testing (NDT), and chemometric analysis and interpretation of oil spectra. One of the examples from the biological systems is non-invasive medical testing (NIT) as applied to the interpretation of electroencephalographs (EEG's). The other example is living-body tissue and cells classification. These applications are briefly introduced in the following paragraphs.

Obviously, any test which is "destructive" will prevent the tested object from functioning usefully after the test. A case in point questions whether a machine which is working well will continue to do so, or have defects, e.g., hidden cracks, corrosion, wear and tear, etc., which will likely lead to an early breakdown. Of course such defects cannot be determined by running a test under service conditions. Thus a "Non-Destructive Testing (NDT)" method is introduced to test the intended or actual performance of a component or a structure without impairing the usefulness of the structure or the component. Generally, any test method in which the test signal has no significantly measurable effect on the properties of a material can be considered as non-destructive. The aim of NDT is to obtain information on the performance of material component or structure. Generally, only some inter-

mediate effects can be measured, which in turn must be related to performance. For example, radiographic methods determine changes in density. These density changes are then interpreted or characterized in terms of defects, which in turn are attributed to actual performance criteria.

Analogous to NDT methods, medical diagnostic testing by non-invasive techniques (NIT) can be defined as any test method by which the performance of living organisms, generally human beings, can be determined without in any way harming the living substance or even causing pain.

The kind of problem we are dealing with in this study is that of classification and assumes an appropriate name according to the nomenclature of the problem area. If the signal source is an NDT application then the classification problem will be referred to as a recognition problem and if it is a biomedical application then, synonymously, the classification will be called a diagnostic problem. The analysis of spectra in chemistry can be referred to as a chemometric interpretation problem. The identification of genetic cells is a biological grouping problem. Thus NDT, NIT, chemometric interpretation, and cell identification methods pose a similar problem: classification or interpretation of signals, based on their structural characteristics. However, each application would require its own set of parametric measurements. The majority of currently available techniques for these typical problems are primarily based on visual inspection and thus are human operator-dependent. The transfer of such human knowledge into the computer provides a basis for use of an automatic knowledge-based system for the interpretation of waveforms.

Although knowledge-based systems have evolved over a period of two decades, their use in the classification of waveform signals is still scarce. First, in Section 1.2, a conventional

signal processing system is outlined and different tools constituting such a system are described. Later the field of knowledge-based systems is reviewed, in general, in Section 1.3. Those systems specially designed for NDT waveform processing are separately discussed in Section 1.4. Section 1.5 reviews the medical diagnostic systems. Section 1.6 establishes the need for an intelligent signal processing system whereas Section 1.7 presents a unifying scheme to build a multi-disciplinary integrated smart system. Section 1.8 outlines the details of each component of a generalized intelligent recognition system proposed in this thesis. The contributions this research offers are summarized in Section 1.9. Lastly, Section 1.10 presents the organization of this thesis.

## **1.2 Signal and a Signal Processing System**

A general signal measurement, classification, interpretation and diagnostic system is schematically shown in Fig. 1.1. Usually such a system consists of a transducer coupled with the information source and extracts the required information. The transducer, in fact, converts the information into an electrical signal (analog) which is conditioned and transmitted, digitized, and submitted to digital conditioning (salt and pepper noise removal), processing (transformation and manipulation) and/or classification (see Fig. 1.1). The transmitted signal may be corrupted with additive and multiplicative noise, and the information required may constitute only a part of the signal such that irrelevant portions are considered noise. In such situations signal conditioning techniques such as noise attenuation and cancellation techniques, or signal enhancement methods, can be applied in order to increase the signal-to-noise ratio and/or information contents. A variety of methods are available for the enhancement of the relevant information in a signal [COHE-86b, STEA-88].

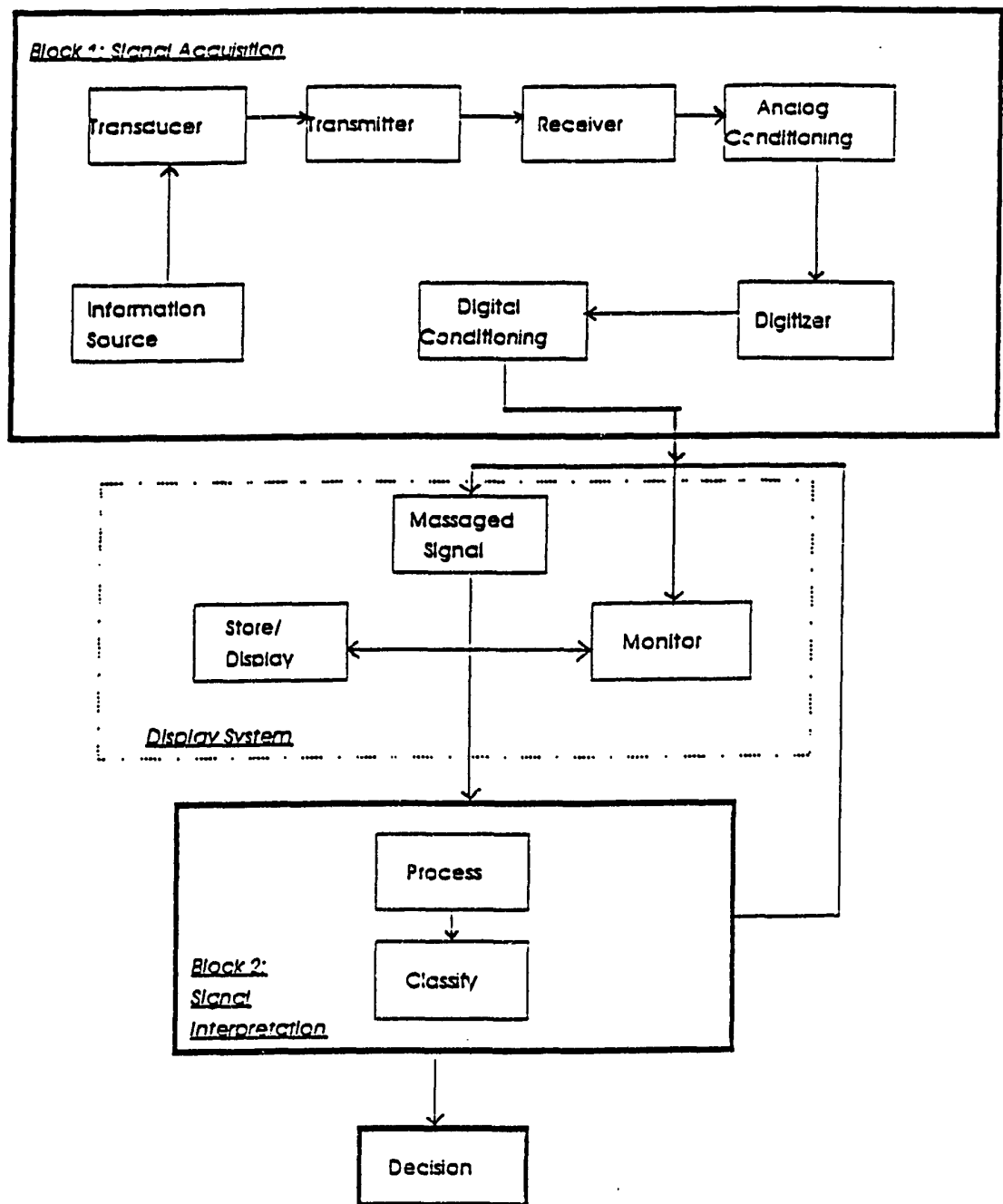


Fig. 1.1: A General Signal Acquisition and Processing System.

Sometimes the signals available may not yield the required information directly or the information conveyed by even a clean and massaged signal is not sufficient for high precision decision-making. In such cases the transformation of signal has been suggested in the literature [BRAC-86]. The most common transform applied in the majority of engineering applications is the Discrete Fourier Transform (DFT). The DFT is used to transform the signal, usually from the time domain into frequency domain, so that spectral information about the signal can be revealed explicitly. In addition to the frequency domain other properties of the transforms can also be used. In some other situations the signal may drastically change its properties with time, i.e., signal is time-variant. In such cases the observations and processes on the signal are performed only in a finite time window. The length and type of the window depends on the signal source and the processing objectives.

Very often only the general wave shape is known. A good example is the electrocardiographic signal, shown in Fig. 1.2, where the general shape of the wavelet, also called as PQRS complex in medical terminology, is known. In order to determine the patient's condition, the extraction of this wavelet present in the signal and its frequency of occurrence (defined in terms of number of cardiac cycles) is required [DICA-93, SIDD-93a,b, TRAH-89]. The wavelet can be extracted using segmentation or similar techniques. Other parameters of interest for this problem are identified in Fig. 1.2 [TRAH-89].

Once conditioning and/or preprocessing is done, the signal or the extracted wavelet is ready for processing (manipulation). Again not all the information conveyed by the signal is necessarily of interest. The signal itself may contain redundant information. When effective storing and transmission are required, or when the signal needs to be automatically classi-

fied, these redundancies have to be eliminated. The signal can be represented by a set of features that contain the required information. These features are subsequently used for storage, transmission and classification. Furthermore, such features are also needed for enhancement and reconstruction of the signal. The number and types of features used dictate, on one hand, the data reduction rate for efficient storage and transmission and, on the other hand, the error of reconstruction. For tasks such as classification and diagnosis, pattern recognition (PR) techniques are usually used. The processes shown in the heavy dotted block of Fig. 1.3 are some of the typical steps involved in PR systems.

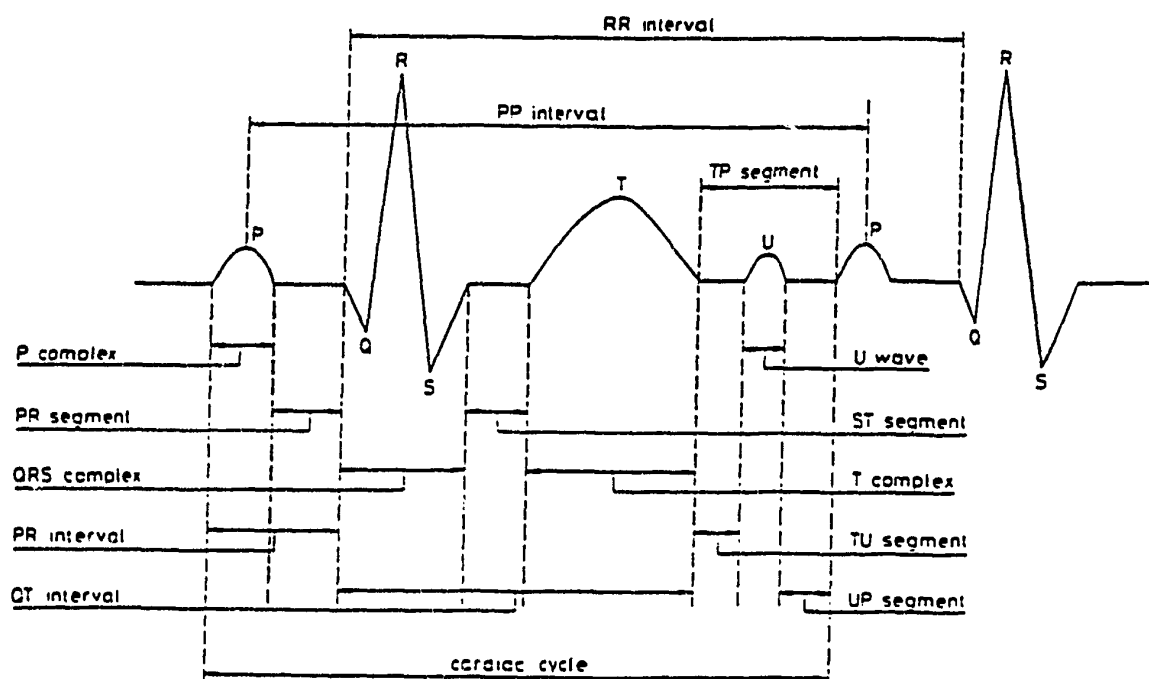


Fig. 1.2: A Typical ECG Waveform and its Information Contents.

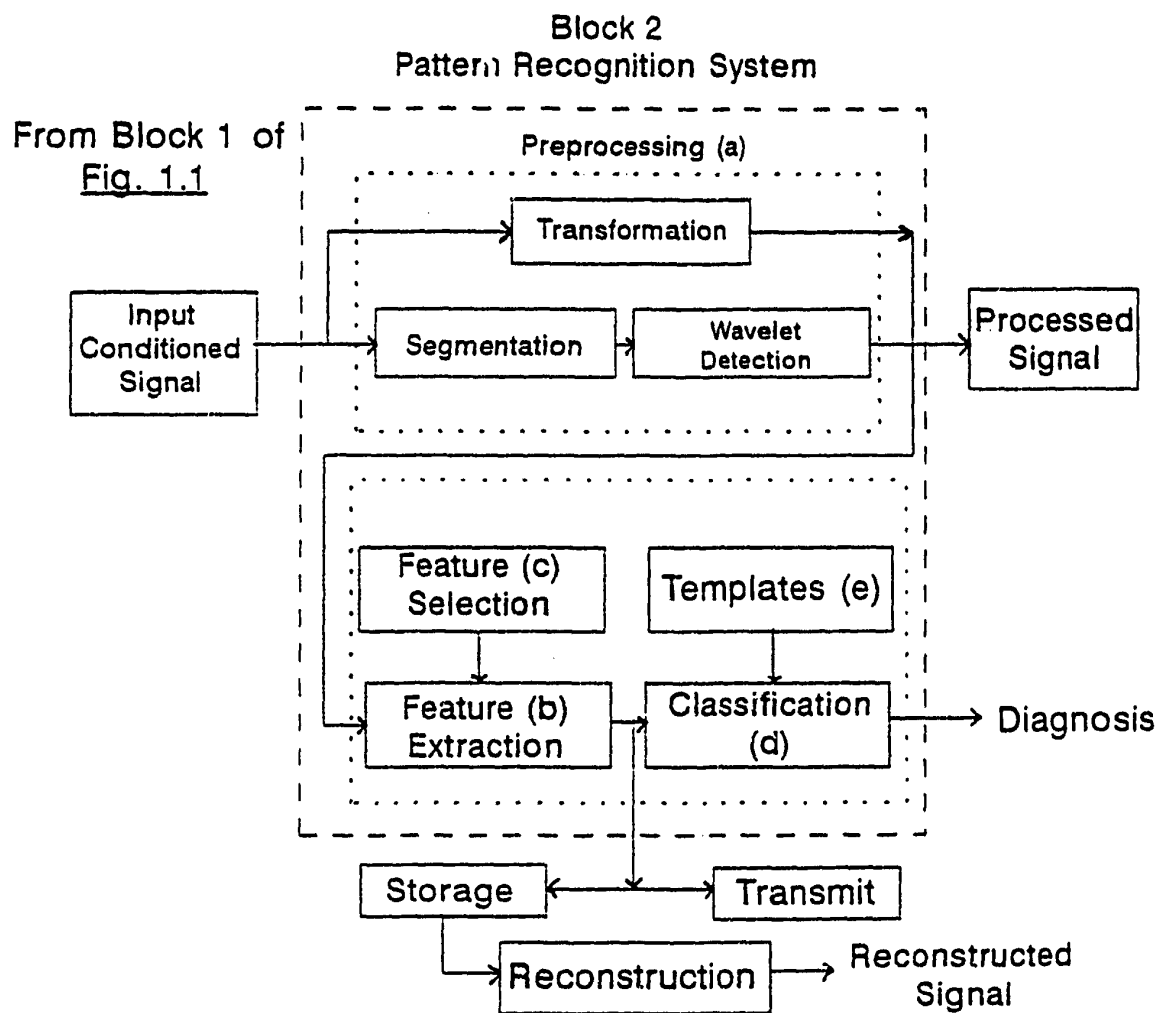


Fig. 1.3: A General Signal Classification/Diagnosis System.

The various functions thus needed in a signal processing and classification or interpretation/diagnosis system are depicted in the blocks of Fig. 1.3. The system shown in Fig. 1.3 is a general system. For specific applications, however, one may delete some of the components or add some other.

### **1.3 Knowledge Based Systems**

Scientists and engineers from every discipline are increasingly interested in including 'knowledge' among the materials from which they construct their artifacts. Knowledge based systems is a generic term and is used to characterize a general class of computer systems which incorporate knowledge as an integral component for decision making. When this knowledge is explicitly acquired from human expert(s), these systems are called expert systems.

Most knowledge-based systems that have been designed to date have been cited as expert systems. An expert system is a knowledge-intensive computer program that solves problems requiring human expertise so as to enable the computing system to perform convincingly as an advisory consultant or a decision maker. In some cases such human expertise is very expensive, rare, and occasionally, not replaceable. The scarcity of the experts and the need for an autonomous system has led the industry and the researchers to store the domain-dependent expertise into a data base and to computerize its subsequent utilization. Perhaps for this reason expert systems and knowledge based systems are growing in many areas, particularly in business applications. Industrial, medical (both clinical and electromagnetic diagnosis), and other scientific fields are beginning to find applications as well.

A typical knowledge based system is shown in Fig. 1.4. Such systems are usually composed of three principal components, a



knowledge base, an inference engine, and a user-interface. The knowledge base mostly contains declarative knowledge that is usually represented in the form of IF <premise> THEN <conclusion>, expressing a problem solving strategy that may be followed by an expert for the domain at hand. The inference engine is made up of decision rules that are used to control how the knowledge stored in the knowledge base is used or processed. The user-interface allows communication or interaction between the system and an end-user. A few examples will follow to illustrate the kinds of consultant services and

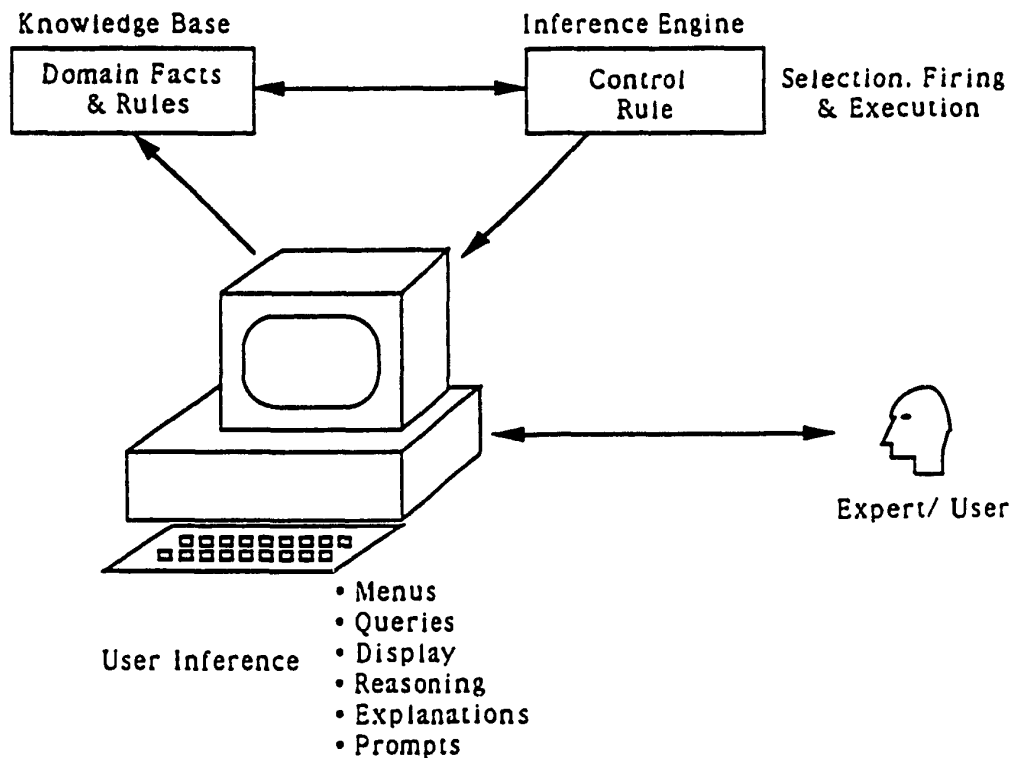


Fig. 1.4: Components of an Expert System.

skills which have been provided by such systems. MOLGEN [MARN-77,STEF-81] interactively aids molecular geneticists in the planning of DNA-manipulation experiments. SACON [BENN-78] guides engineers in the use of a large program which integrates structural analysis procedures. ITES [NAGA-91] deals with human images of feelings and translates them into ergonomic knowledge.

Well known examples of expert systems developed for industrial applications include PROSPECTOR [DUDA-78a,b] which has been successfully used to locate mineral deposits, Dipmeter Advisor, which is used for oil exploration [DAVI-81] and R1, which is used to configure computers [MCDE-82]. DENDRAL [BUCH-78, LIND-80] takes the pattern generated by subjecting an unknown organic chemical to a mass spectrometer and infers its molecular structure. SECS [WIPK-74] uses a 'knowledge base' of chemical transformations to propose schemes for synthesizing stated compounds. SES (Spectral Expert System) uses spectral information to characterize polynuclear aromatic compounds [SIDD-91a]. GUIDON is a knowledge-based tutoring system [CLAN-87]. End-game expert system deploys and discusses chessmaster 'knowledge and generates improved teaching texts [BRAT-78,BRAT-80]. Examples of other tutoring systems are [BRWN-74, BRWN-75, BURT-82, CARB-70].

A number of knowledge-based systems have also been developed for medical applications [ANBA-87, BETA-91, NAUD-83]. They will be reviewed separately in Section 1.5. The systems developed for applications in chemistry are reviewed by the author and others in [BROW-88, PAVE-86, SETT-87,SIDD-89a,SIDD-91a]. Further information on many of these systems and other applications is available in [BUCH-79,IRGO-90, LIEB-86, MOLD-87, NAUD-83, SIDD-88, SIDD-89a,b,c,d, SIDD-90c, SIDD-93a, SIDD-94a,b].

#### 1.4 NDT Waveform processing systems

Until the beginning of the 70's, non-destructive techniques (NDT) grew primarily as an experience-based technology. The field depended on skilled, experienced inspectors who progressed through certification programs based on knowledge of fundamentals and accumulated experience (expert knowledge) [MCGO-61]. While an underlying science base was developed through the 1960's and 70's, the qualified inspector remains the corner stone of the technology. Technological developments tended to be technique-based and explored new methods for interrogating materials. Since the 70's, there have been significant changes, there is not only a growing need for NDT technology, but there is also a broader perception and acceptance of its importance and applications.

A case in point concerns large engineering development projects, which characterize various new approaches to generate, transport, or use energy. These systems very often use materials close to their limits and the consequence of any failure can be catastrophic. The need for a timely and adequate inspection system is naturally obvious.

In recent years, because of the need for reliable and precise flaw classification techniques, the thrust of the research has been shifted from manual/analytical techniques to automatic multi-disciplinary techniques. Among the multi-disciplinary approaches, Shankar et al. [SHAN-78], Rose and Singh [ROSE-79a,b], Chan et al. [CHAN-85a,b], and Hay et al. [HAYD-84, HAYD-88] have demonstrated the feasibility of applying signal processing and pattern recognition techniques. Another approach combines signal processing and artificial intelligence [SING-83]. Other approaches combine two or more disciplines of image processing, holography, statistics, signal processing, artificial intelligence, and pattern recognition

[CHAN-80, CHAN-82, CHAN-85a,b, ELSL-83, HARR-80, HAYD-84, MAHL-85, ROSE-84, SIDD-86a, SIDD-88, SING-81].

As a result of these multi-disciplinary approaches, knowledge based systems and expert systems for NDT are beginning to emerge. For example, Mahalingam and Sharma [MAHL-85] reported a preliminary system called WELDEX for testing of welds and Siddiqui et al. [SIDD-87a] proposed the design of an expert system called KNOMES (Knowledge Monitoring Expert System). Primarily, the WELDEX system is derived from a medical system [GOME-81] and uses radiographic and other image processing techniques to diagnose the weld defects whereas the KNOMES system was based on signal processing and pattern recognition methods and was designed to classify NDT signals. None of these systems showed any experimental results. However, the work reported in this thesis uses the KNOMES' design approach [SIDD-87a] as the basis and provides experimental results on several application areas discussed above.

Another system of this type is ICEPAK (Intelligent Classifier Engineering Package). To the best of our knowledge this is one of the most effective and practically operating, and commercially available AI-based system which uses advanced signal processing and pattern recognition techniques for waveform characterization. This system has been successfully used on different kinds of waveforms from both material testing, and medical interpretation, to characterize flaw/defect and disease problems of 2 to 6 classes [CHAN-85a,b, HAYD-84, HAYD-88]. ICEPAK's design philosophy is partially used in the development of preprocessing and feature extraction modules.

### **1.5 Medical Diagnostic Systems (MDS)**

Physicians' diagnostic decisions are central to the treatment and care of patients. Often a physician's decision could mean

the difference between life and death. The question of how a physician should react to each new test result is persistent and pervasive. The information needed to answer a series of questions related to the outcome of such test results is based upon experience gained from training, and more immediately, information sources such as medical records, laboratory reports, textbooks, documented case histories, etc. In a majority of cases the answer requires simple list searches and comparisons, but the amount of information required to conduct the search, compare and remember is enormous. Human beings are not perfect. Given both the human limitations and extensive medical training of physicians, many medical errors may well be due to the limitations of endurance and instantaneous recall, rather than to remediable flaws in their funds of knowledge. There have been efforts made to reduce the chances of such errors [ANBA-87, BUCH-84, COOP-84, GOME-81, MILL-82, NATH-84, SHOR-76].

For example, WAMIS (German acronym for Vienna General Medical Information System) is an integrated knowledge-based medical system [ADLA-89] that combines medical information system with medical expert system CADIAG-2 [ADLA-86]. It links medical records, hospital testing facilities, pharmacy, patient's rooms, and outpatient clinics. By putting this information together, along with the factual and heuristic medical knowledge in its knowledge base, the knowledge system can interpret instrument readouts, suggest what the patient's illness might be, and advise on the proper drugs and treatment. Such consultant systems and several specialized disease-oriented systems have been in use since the middle 1970s. For example PUFF (PULmonary Function) system, developed at the Pacific Medical Center is a diagnostic system for pulmonary diseases [AIKI-83, KUNZ-78]. Ventilator Management (VM) [FAGA-78, FAGA-80] gives real-time advice on the management of intensive care patients' mechanical ventilation

at the same center. The Regenstrief Institute of Health Care (Indianapolis, Indiana) knowledge system called CARE, and the LDS Hospital (Salt Lake City, Utah) knowledge system named HELP, are both consultants that handle a more comprehensive range of physicians' chores [NATH-84, PRYO-83]. HELP has been shown to reduce healthcare costs. Some of the patients are treated even at remote locations. Barney Clark, the recipient of the first artificial heart was among such patients treated by HELP system. MYCIN [SHOR-74] and INTERNIST [POPL-77] have outperformed clinical consultants within the bounds of the systems' stored expertise. Foetos [BETA-91] is one of the most recent expert system developed for the assessment of fetus in a high-risk pregnancy. There are other computer programs such as, CADIAG-2 [ADLA-86], INTERNIST-1/CADUCEUS [MILL-84], ONCOCIN [BUCH-84], QUICK [FIRS-85], RECONSIDER [TUTT-83], that are practically operating in the hospital or physician's office. Some of these systems have already been tested with hundreds of clinical cases. Additional examples of medical diagnostic systems are cited in [ANBA-87, BENB-80, BUCH-84, COOP-84, FINK-89, GOME-81, HERN-89, LUGE-89, MILL-82, SIDD-93a, SHOR-74, SHOR-75]. Several medical tutoring systems are also reported in the literature [CLAN-81, HEID-88].

Most of the knowledge-based MDS systems are general care monitoring and analysis systems. The inference mechanism (problem-solving method) used in these systems is basically a variation of the hypothesize-and-test process. However, recently, a number of successful attempts of using elementary PR techniques and typical signal processing methods for simple classification and diagnostic problems in medicine have been reported [BENB-80, BUCH-84, LUGE-89, PAHL-87, SIDD-93b, SLAG-89].

Only a handful of the MDS systems mentioned above are capable of simultaneously analyzing, monitoring and diagnosing. Problems arise because of the complex nature of the input data and

number of parameters affecting it. Biomedical signals are usually extracted from living organisms. The living biological system is a very complex system governed by interactions of numerous biochemical, physical, and chemical subsystems not well understood as of yet [COHE-86a, PICT-88]. The complexity of the biological system introduces difficulties in measurement and processing procedures. In particular, many aspects of the complex hierarchical control of the brain and the nervous system, the genetic control, the neural information transfer and processing, and other systems are still under extensive investigation.

The large variations that exist in biomedical signals usually suggest the use of statistical methods. These variations exist in signals acquired from an individual and, of course, between populations. Consequently, the accuracies and confidence limits that come out of biomedical signal processing are usually not very high, at least in terms used in engineering disciplines. The measurement systems most often use Non-Invasive Techniques (NIT). This means that very often the requisite information cannot be acquired directly and one has to infer it from signals that are non-invasively available. Fetal heart monitoring is a good example. Rather than applying electrodes directly to the fetus' skin which is an invasive procedure, we non-invasively place the electrodes on the mother's abdomen. Unfortunately, the signals thus acquired are heavily contaminated with the mother's strong ECG and muscle activities (EMG).

Thus, one of the MDS problems is to extract pertinent pieces of information from highly contaminated signals and be able to analyze and diagnose (recognize) them using a machine. It appears that, as yet, no operative example of building such a stand-alone monitoring and diagnostic MDS system exists.

## 1.6 The need for a Stand-alone System

From the review of knowledge-based and signal processing systems presented in Sections 1.3 through 1.5, clearly two different problem-solving approaches have emerged, signal processing supplemented with pattern recognition methods and an expert-system approach.

Perhaps signal (waveform) sensing and processing systems which are primarily based on numeric processing are among the most powerful tools currently available for examining the internal structure of materials, chemical spectra and living biological substances. Using the waveforms, these systems have found wide application in industrial flaw detection, thickness gauging, spectroscopy to identify the components in chemical compounds, and very recently in medical diagnostic systems such as electrocardiography (ECG), electroencephalography (EEG), electro-neurography (ENG), electrooculography (EOG), etc. The other approach that heavily entails symbolic processing is that of expert systems which emulates human problem-solving in the form of hypothesize-and-test type symbolic processing.

While both of the above approaches have produced impressive performance at times, they currently face a number of limitations when applied to real-life problems [REGG-83, WEIS-84, SIDD-89d, SIDD-93a,b]. For example, problems where multiple pattern classes with complex representation are present simultaneously have proven very difficult to handle [POPL-77, SIDD-89d]. In addition, AI models of diagnostic reasoning are often criticized as being "ad hoc" because of the absence of a formal, domain-independent theoretical foundation [BENB-80]. Current systems are limited in size (of problem) and scope with limited learning capability and require extensive amount of expert/operator input. Rather, these systems provide enhanced displays and presentations to assist the human analyst in



his/her interpretations more likely with dated knowledge.

Ideally, for any computer software to be called an expert (intelligent) system, it should at least have the following characteristics:

- a mechanism to extract knowledge from raw observation and/ or from the expert, i.e., a mechanism that could combine numeric and symbolic processing.
- a mechanism to optimally associate observations to the pattern classes.
- a domain-specific knowledge base that may be updated as more knowledge becomes available.
- an optimal mechanism to represent and organize knowledge so that the problems such as information explosion and redundancy could be minimized.
- an inference mechanism that based on the knowledge available may choose an appropriate decision-making algorithm.
- an explanation/communication facility that may provide a logical explanation of the solution plan.

More recent efforts in AI research have stressed the use of large stores of domain-specific knowledge as a basis for high performance of expert systems. The knowledge base, which is the key component of this sort of program (e.g., Dendral [FEIG-71], MACSYMA [MART-71], Foetus [BETA-91]) is traditionally assembled manually. Such an assembly of a knowledge base is considered an ongoing task that typically involves numerous man-years of effort and continual interaction with expert(s). A key element in constructing a knowledge base is the transfer of expertise from human expert(s) to the computer program. The process of knowledge transfer is called knowledge engineering. Often, the domain expert is unfamiliar with the knowledge engineering process and usually unable to structure his strategic problem solving approach into a logical algo-

rithmic form. Also, it is virtually impossible for an expert to consistently extract the problem-solving information from physical observations. These factors make the acquisition of an expert's knowledge a very difficult task. Further, the manual acquisition of knowledge from a human expert is a very labor-intensive process. Therefore, we acknowledged a need to have automatic and formal aids for knowledge acquisition, formalization and its processing for the results, as part of the system.

### **1.7. Integration of Information Processing Technologies**

With the previous discussion it appeared to be essential that the methods available in a single field would not be able to provide us a complete set of tools to build the system we intended. A multi-disciplinary approach is essential. Now the question arises as to which disciplines to use and how to blend these disciplines to develop an integrated system to overcome some of the limitations discussed above. The approach we adopted is to borrow the concepts from a field only for those functions that the system can perform best. Thus considering the ideal characteristics, an intelligent system should have (see Section 1.6) the general objectives and scope of the problem we solved were established as below:

- Define a stand-alone signal classification and discrimination system that combines physical observations and expert knowledge.
- Minimize human biases in the selection of knowledge parameters, selection of knowledge processing techniques, and decision-making.
- Be able to solve a fairly large set of problems in a problem domain without degrading the performance.
- Be able to adapt itself to new problem domains by simply providing the system with a knowledge base in that problem domain.

- Be able to function both as an expert and as a consultant.

Two basic approaches of solving classification problems are discussed in previous sections, namely, pattern recognition (PR) and expert systems. Pattern recognition approaches primarily use numerical methods to automate the interpretation process by defining a set of characteristics for a pattern class. However, the PR techniques make no effort to identify the specific structure/property involved; they merely try to suggest structural elements (features) that the unknown may contain. In fact, this is the information that is used for the discrimination of a pattern class. A major advantage of these techniques is that they make no a priori assumption regarding the structural information used to discriminate a pattern class. As such, without imposing any solution, it is possible that useful new information may be uncovered. An expert system, on the other hand, uses the known interpretation logic of the human expert which is usually represented in symbolic form, and since it is stored in machine, the scope and effectiveness of decision-making is limited to the amount and depth of the knowledge stored.

The problem-solving approach adopted in this research uses a hybrid scheme that combines both of the above two approaches to build on their strengths rather than living within their limitations. The functional view of a unified system combining the set of approaches that have been shown diagrammatically in figures 1.1, 1.3 and 1.4 is presented in Fig. 1.5.

The techniques for signal registry and conditioning are borrowed, as is, from typical signal processing systems (Block 1 of Fig. 1.1) and used for information acquisition, conditioning and enhancement. The next set of techniques selected is PR techniques (Block 2 of Fig. 1.3). Pattern recognition

techniques are primarily evolved from the human process of vision, recognition and perception. Selection of good pattern recognition tools could reduce the burden on the human expert and save an unaccountable number of man-years which are otherwise required only to acquire the expert's knowledge. Moreover, since pattern recognition techniques use features which are easier to represent and easier to apply to decide the pattern identity, they will be very useful in formalizing and structuring the reasoning and explaining process for the actions of an inference engine. In addition to PR techniques, information transformation techniques such as Fourier or Hartley transform can be used to transform the signal to a more meaningful and explicit form.

The parameters extracted can be stored in a knowledge base and then usual expert systems' approach shown in Fig. 1.4, or knowledge-based pattern recognition methods can be used for classification. The inference engine, therefore, can be designed using both the declarative knowledge (from the expert) and the procedural knowledge. The procedural knowledge comprised of a set of pattern classification algorithms will use physical observations (pattern features) for decision making. To process the declarative and heuristic knowledge iterative dichomotization as suggested by Quinlan [QUIN-86], or any other algorithm with similar function can be used. Thus by combining these disciplines an intelligent waveform processing and recognition system, shown in Fig.1.5, is proposed. The details are presented in the following section.

### **1.8. A General Intelligent Recognition System (IRS)**

A general intelligent recognition system (IRS) is shown in Fig. 1.5. Before we describe the components of this system, several key terms and the context in which they are used will be defined.

# Knowledge Acquisition, Representation & Organization (KARO)

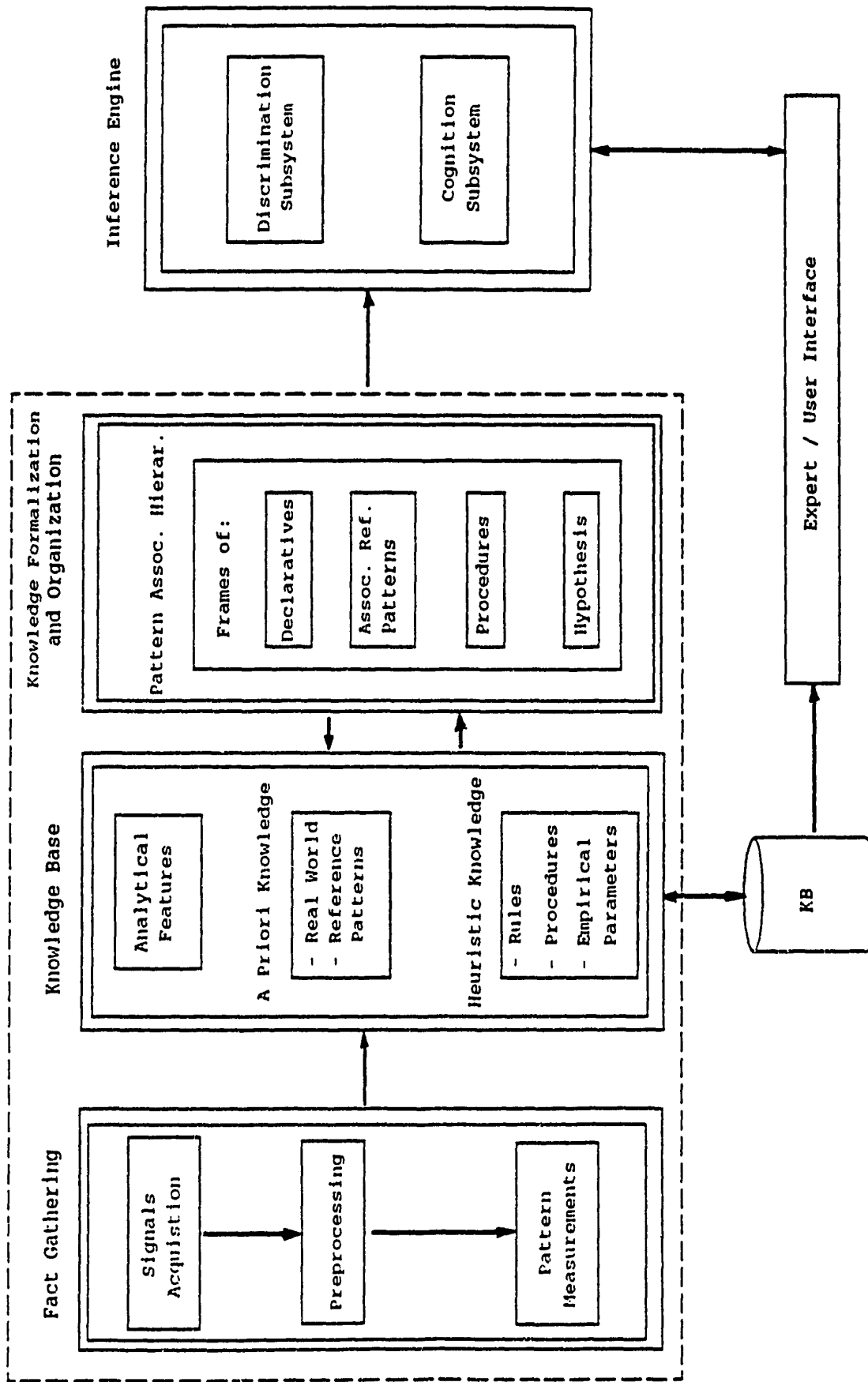


Fig. 1.5 A general design of an Intelligent Recognition System

The basic objects within the knowledge base are Facts and they refer to real-world knowledge about the problem domain which include the physical observation collected from the subject and any additional knowledge the signal may carry. These observations (signals) carry information about the source or the path which may be extracted in the form of pattern measurements (features). Based on their values, these features may indicate the pattern belongs to one particular class, i.e., indication.

The system shown in Fig. 1.5 is composed of three subsystems, 1) Knowledge Acquisition, Representation, and Organization (KARO) subsystem, 2) Inference Engine, and 3) the Expert/User Interface. The design philosophy and operational details of these components are presented in subsequent chapters, however, a functional overview of the system is presented in the following paragraphs.

The basic cycle of the KARO subsystem consists of three phases: namely, fact gathering, knowledge base, and knowledge formalization and organization. The fact gathering phase performs three main tasks, 1) acquisition of input data (waveform signals), 2) data preprocessing, and 3) pattern measurements. Each of these tasks further entails a series of operations. Since the pattern measurements are performed using analytical tools they will be called analytical features hereafter. Analytical features constitute a major portion of the knowledge base which is the next component of the KARO subsystem. A set of analytical features representing one physical observation constitutes a pattern vector. A set of pattern vectors representing the design set serves as a set of examples or reference patterns the system may be able to classify. Other components of the knowledge include a priori knowledge, and heuristic knowledge conforming the problem domain. A priori knowledge includes a number of parameters

pertaining to the physical and operating characteristics of the apparatus required for the recognition/interpretation experiment at hand. The analytical features are further used to derive empirical knowledge parameters some of which in turn are used to simulate human judgement and hence these parameters are considered as heuristic knowledge. Heuristic knowledge comprises of, 1) a set of rules representing expert's fixed (book) knowledge, 2) a set of procedures that an expert may apply on physical observations, and 3) a set of empirical knowledge parameters to formulate an expert's judgement and objectives. The function of a priori knowledge is to identify corrupted signals and based on stored knowledge, direct the preprocessing that the system should perform. The heuristic knowledge serves as meta knowledge and is used to select an appropriate algorithm at different stages of processing and controls its application on requisite data.

The next component of the KARO is knowledge formalization and organization wherein a series of operations are performed to sort, structure and organize the accumulated knowledge. Analytical knowledge takes advantage of the natural association that exists among patterns to build their pattern association hierarchy (PAH) which is a new concept introduced by the author in [SIDD-87a]. PAH supplemented with a priori and heuristic knowledge thus indicates the actions the system should initiate when some parameters reach certain thresholds. To use the knowledge base efficiently, the knowledge is formalized using a combination of knowledge structuring and rule building methods. The analytic features corresponding to an individual pattern are transformed into several sets of selected feature vectors. Empirical knowledge and heuristic knowledge pertaining to a set of associated patterns are stored as declaratives and procedures in structures called knowledge frames. A knowledge frame with appropriate node-dependent knowledge is placed at each intermediate node in the

PAH. Such organization of knowledge not only facilitates the storage and retrieval of pattern class-dependent information, but also increases the pattern recognition performance.

The same pattern association hierarchy is used by the inference engine to classify an unknown pattern. Using the known identity of the patterns, the inference mechanism of the engine is trained for an appropriate classifier at each node of the hierarchy. The classifier and the training information for each intermediate node is stored in the respective frames.

Once the knowledge is formalized and organized using the known identity of the signals, and the inference engine is trained on the problem at hand, the system can be used to characterize an unknown set of input signals.

Such a system could be designed to operate in two distinct modes, executive and consultant modes. In the executive mode, the system can operate autonomously by processing through the PAH and would not require any human input to solve a problem. To train the system for this mode of operations, supervised learning is required in which the system has the knowledge of the pattern classes it will be identifying. In the consultant mode, the system can be designed to function as a synthesizer that will allow the expert to monitor the pattern classification process by modifying a number of key decision parameters that we have identified for this purpose. To train the system for this mode of operation, unsupervised learning would be needed in which the expert has to provide initial input on identification of pattern classes.

If the dual mode of operation is implemented then the utilization of heuristic knowledge will depend on the operating modes of the system. If the system is in the executive mode, only the pre-stored fixed knowledge is utilized for the final



determination of pattern's identity. When the system is in the consultant mode, expert input is sought at each phase of the classification process. At present the system assumes the executive mode of operation only.

The last component of the system which is not implemented in this research, is the user/expert interface. The function of this component is to provide a communication and explanation facility for the actions which the system undertakes. A skeletal picture of the components and the techniques that are required for the development of the whole system are summarized in Fig. 1.6.

### **1.9 The Research Contributions**

The objectives discussed in Section 1.6 were established to define the context of the present work. To successfully achieve those objectives, a number of new algorithms and concepts which essentially constitute the contributions of this research were developed. Some of the notable contributions are listed below.

#### **1. System Concept Level**

Until the work reported in the thesis was carried out, the knowledge based approaches to system development were essentially isolated. That is, the systems were developed either using physical observations only or they were emulated using expert knowledge. This thesis proposes an integrated approach to a knowledge based pattern recognition system that combines both physical observations and empirical knowledge and develops several components (see below). Those components evaluated in this thesis demonstrate the general feasibility of the approach.

### **Knowledge Type**

A priori knowledge

- \* Facts about
  - problem domain
  - test equipment
  - test specimen characteristics
  - test conditions

Analytic knowledge (physical observations)

- \* Statistical features
- \* Waveform features

Heuristic knowledge (implicit/derived observations)

- \* From Expert
  - judgement & objectives;
  - specific methods/procedures including classification algorithms
- \* From System
  - empirical observation
  - meta knowledge

### **Knowledge Representation and Formalization**

Analytical feature vectors

Empirical knowledge and declarative rules

### **Knowledge Organization**

Declarative & Frames

Procedures

Pattern Association hierarchy

### **Inference Engine**

Machine Learning

- supervised/unsupervised

Inference Mechanism

- \* Discrimination strategy
  - parametric classification schemes
  - non-parametric classification schemes
- \* Cognitive strategy
  - information-theoretic decision tree

### **Operation (decision making environment)**

Executive (Tutorial)

Subordinate (Consulting)

### **Explanation/Reasoning Mechanism**

Forward/backward chaining through the hierarchy

Fig. 1.6: Skeletal View of the components and techniques used in the Intelligent Recognition System.

## 2. Feature Level

What features should be selected? Addressing this issue, we suggest to extract all useful features one can think of and then use 'Successive Feature Elimination Process' to weed out the poor performers and later use one of the two feature ranking and selection algorithms, i.e., Fisher discriminant index or Pseudo-Similarity method to select a smaller feature set. Successive Feature Elimination Process and Pseudo-Similarity method are the contributions introduced in this thesis.

## 3. Knowledge Organization Level

To minimize the problem of knowledge explosion and redundancy, a new concept called the pattern association hierarchy is introduced wherein several existing algorithms (a few were modified) and a new clustering algorithm called 'Generalized Variations' method are used to hierarchically organize the pattern classes and their associated knowledge. This arrangement thus always stores the knowledge pertaining to only two groups (or classes) at each non-terminal node of the PAH. Data-dependent 'Rules' are developed to select an appropriate clustering algorithm.

## 4. Classification Level

A number of observation-dependent parameters are designed to automatically determine the data statistics which in turn determine the pattern classification algorithm to be used. The new concept of pattern association hierarchy along with feature elimination and selection methods, data-dependent parameters, and, a number of parametric and non parametric classifiers at each node of the hierarchy gives birth to a flexible PAH-classifier. This method reduces the bias that

may be introduced by the human judgement while designing each of the above phases.

#### 5. Integration and Automation Level

Instead of human judgement, the system primarily relies on analytical tools to synthesize the available information. This synthesis evolves a set of new parameters (empirical knowledge) which are used to partially simulate human judgement. Using empirical knowledge a set of rules are designed to automatically select an appropriate algorithm among several available at different phases of processing. Thus a high level of automation and integration is achieved.

#### 6. Application Level

The project develops a generic signal classification scheme by successfully applying the system to three different application areas, namely, non-destructive testing, spectroscopy and medical diagnosis.

#### 7. Size of Problem, Performance, and Robustness

The algorithms we developed are not restrained by the size or the nature of the problem. We solved four problems with 3 to 20 pattern classes, up to 112 features and 2 to 200 samples in a pattern class with consistently high individual class performance of 60% to 100% mark for various problems.

#### 1.10 Thesis Organization

Altogether, the thesis comprises eight chapters. It begins with a chapter entitled, "INTRODUCTION" that describes waveform sensing and reviews some of the important systems developed for signal processing in first two sections. Review

on the knowledge based systems is included next. The systems developed for two of the application domains, i.e., the NDT signal processing and the medical diagnostic systems are briefly reviewed in the next two sections. After establishing the justifications for a knowledge based intelligent recognition system, the components of the proposed system are described together with a list of contributions this research makes.

Chapter two presents a general IRS system and describes its components. It further describes how the physical observation and other knowledge components pertaining to a problem domain are gathered, and analyzed. Several generic algorithms developed for knowledge organization and selection of optimal parameters for pattern discrimination are described in Chapter 3. Chapter 4 describes the components of the empirical and meta knowledge concentrating on techniques for knowledge abstraction and organization. A priori knowledge, physical observations, and empirical knowledge specifically for the NDT-domain are described in these chapters. The architecture of the inference engine is described in Chapter 5. The inference engine is a composite of two independent solution strategies, discrimination and cognition. The cognition algorithms used are also described in this chapter. The algorithms which formulate the discrimination strategy are presented in Chapter 6.

Chapter 7 presents the results of several experiments conducted on four data sets. These results are studied in detail and the observations made are discussed.

Finally, Chapter 8 concludes the research, reviews individually all the components of the system and summarizes the contributions of this research. It also presents the directions for future research in this area.

## Chapter 2

### Common Elements of an Intelligent Recognition System

#### 2.1. Introduction

We observed four levels of abstraction in knowledge hierarchy. They include primary concepts, physical observations, derived facts, and heuristics. Primary concepts refer to a priori knowledge pertaining to a problem domain. The information that the physical observations (raw signals) may carry is considered as facts. The derived facts and the knowledge derived from the primary concepts are referred to as the empirical knowledge. Using empirical knowledge and statistical decision theory a set of meta-rules were derived. Meta-rules are considered as heuristics and hence these two terms will be used interchangeably. Using this quadruple concept of knowledge abstraction the first subsystem, Knowledge Acquisition, Representation, and Organization (KARO) subsystem, of the intelligent recognition system is designed which forms the main subject of this chapter. The information a waveform carries is modeled and requisite knowledge pertaining to the NDT problem is structured. Some thoughts on how to incorporate instrumentation and operating conditions of apparatus are also presented in this chapter. The empirical and meta-knowledge concepts are the subjects of Chapters 3 and 4 respectively.

#### 2.2 Signal Interpretation - Current Practices

To conceptualize the knowledge requirements for individual problem-domains, it was essential to study how different types of problems included in the present study are being currently solved. After carefully reviewing the current state of the technology we noticed that the majority of such techniques are

based on visual inspection. In visual inspection techniques the discrimination is largely based on the indications of amplitude and arrival time (or intensity and wavelength), or in some cases using the signals in the transformed domains (see Sections 1.1 and 1.2, and Fig. 1.1 and Fig. 1.3) and the discrimination process simply decides on an accept/reject criterion. For large scale interpretation and data synthesis, however, analytical methods for evaluating the signal artifacts have been used [STAL-82].

In current methods for spectroscopy, since a majority of polynuclear aromatic compounds (PNA) have unique peaks at a given wavelength, frequency domain transforms are usually used whereas in electroencephalographs (EEG) analysis, time domain signals are considered important. In NDT-testing, however, the inspection records are usually obtained in three forms, A-scan, B-scan and C-scan. The A-scan is amplitude/time display for a specific point on the specimen and basically a time-domain signal as described above. B-scan is an amplitude curve along a scan line whereas the C-scan is a planar view (2 dimensional) image and is obtained through a series of scans covering the surface of the specimen.

Among the recent practices for NDT-testing the A-scan is further submitted to analysis using signal processing techniques. In most signal processing applications, an important waveform transformation called Fourier Transform is employed. This transform has been used to translate a time domain signal into what is known as power domain signal. A power domain signal can also be referred to as a frequency signal because it actually shows the power of the signal at each frequency. These domains have been used to improve classification considerably [HAYD-84, MATT-89]. A reference signal is often used to remove the local geometry and grain properties, in addition to transducer characteristics. Other techniques involve using

deconvolution for separating signals. The deconvolved signal can then be analyzed in two other domains in addition to the frequency domain. By taking the inverse transform of the deconvolved signal, a time domain signal is obtained. According to the theoretical model, this signal should be trapezoidal function whose dimensions are related to the defect dimensions. Taking the complex logarithm of the spectrum and then performing the inverse Fourier transform yields the cepstrum, where peak locations in frequency are also indicative of crack dimensions.

The above discussion thus suggests that any automatic inspection system should select features based on physics and mechanics of the problem with respect to expected modification and changes in the signal amplitude versus time waveform or versus wavelength (frequency domain). Time and frequency are important domains for information synthesis since many phenomena occur only at a specific frequency or within a particular range of frequencies. To enhance the performance of the categorization process, however, these observations (features) must be supplemented with the additional knowledge that may either be obtained from a priori (real world) knowledge, or the expert knowledge of the operator.

### 2.3 Modeling the Waveform Indications

Reviewing the signal interpretation process presented in previous section, we modeled the information a waveform signal carries. EEG signals, PNA spectra, and NDT applications, or perhaps any application that can be represented by signals, all require electronic data acquisition, preprocessing and other conditioning according to the factors which contribute to the nature of the signal and their information contents. Since each problem domain involves different instrumentation and different test subjects (or specimens) having a different



set of inherent characteristics, the factors affecting the indications would vary from problem to problem. For example, the EEG signals are corrupted by noise and/or artifacts, introduced by the electrodes, eye blinks and other body movements, instrumentation fit-up and other elements in the measurement set up. Here, the application of more complex methods is needed for a proper evaluation of the differences in the received indications regarding disease (defect) signals and interfering signals.

In this section we will individually model the accumulated effect of surrounding factors on NDT, EEG and PNA signals. Several parameters are designed to accommodate these factors individually.

#### **2.3.1 NDT Indications Model**

The inspection and evaluation process poses a special challenge to NDT engineers due to a variety of reasons related to complex metallurgy, structural geometry, unpleasant or hostile radioactive environment, high temperature, noise from the equipment and environment, random variations in signals, varied properties of the test objects including their geometry and thickness, limited access to the component, NDT-method dependent parameters, and the presence of defects as well as distance and orientation of test equipment with respect to the test object and the testing conditions.

Modeling the indications requires that a candidate set of input parameters to the model be defined. This candidate set must be a summary description of the known and instrumental variables. In addition, the parameters input must be related to the physics of the underlying process or at least to an intuitive understanding of the process. Since human interpretation of NDT signals, or EEG/PNA signals for that matter,

is dependent solely on the observed characteristics of the signals, the input parameters should be attributable to mathematical characteristics of the signals.

To sort the indications a variety of contributors to NDT-indications mentioned above have to be considered. With the knowledge pertaining to the source of these factors their influences can be minimized considerably. Considering these factors a waveform can be defined as the sum of overlapping defect and geometrical or structural indications.

Based on simple accept/reject criterion the NDT-indications could be divided into defects, and non-defects. The contributors to the non-defect indications primarily include the parameters from test equipment, operating conditions and the material properties. Based on the dichotomy of indications (defects/non-defects) the parameters which contribute to NDT indications are hierarchically organized in Fig. 2.1.

Thus the basis for modeling these indications in a waveform (NDT) is a measurement model that relates the output indications of the system (test equipment),  $W(n,d)$ , to its various signal and noise components shown in Fig. 2.1. In designing the model we assumed that the effect of the components is independent of each other and used an additive measurement model as defined by:

$$W(n,d) = W(d) + W(E) + W(C) + W(P) \quad \dots 2.3.1$$

where

$W(d)$  : is the indication produced by the defect,  $d$ , in the absence of non-defect artifacts,

$W(E)$  : is the indication (artifact) produced by the equipment,  $E$ , and is the sum of artifacts from wave mode,  $W(mode)$  and electromechanical artifacts,  $W(a)$ , that is,

# NDT Indications

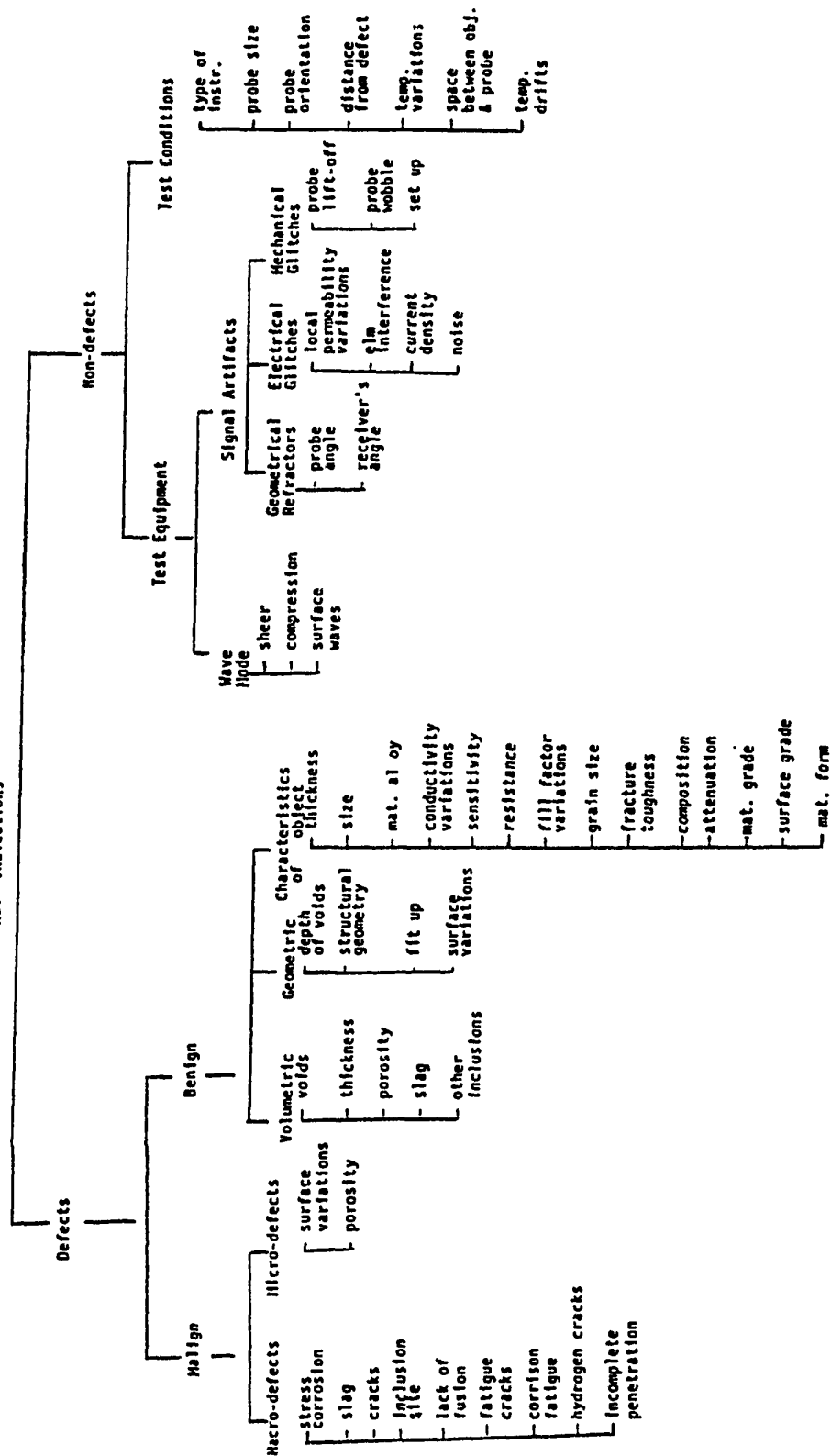


Fig. 2.1: NDT Indications

$$W(E) = W(mode) + W(a) \quad \dots 2.3.2$$

$W(C)$  : is the indication (artifact) produced by the operating conditions,  $C$ , of the equipment,

$W(P)$  : is the indication (artifact) produced by the material properties,  $P$ ,

$W(n)$  : is the accumulated indication (total noise) produced by the non-defect parameters together, i.e.,

$$W(n) = W(E) + W(C) + W(P) \quad \dots 2.3.3$$

$W(n,d)$  : is the composite indication and consists of both signal and artifact indications.

By collecting the parameters shown in Fig. 2.1 the factors in equation 2.3.1 which contribute to corrupt the signals can be eliminated and the quality of the signal can be enhanced right at the source. The components pertaining to NDT-problem domain which would contribute to increased performance are described in the following sections.

### 2.3.2 EEG Indications Model

It has been long known that the brain generates waves called electroencephalographs (EEG) [GEVI-80]. An EEG is a slow (0.01-100 Hz) electromagnetic wave that pervades the brain tissue [PICT-88]. It can be recorded either with electrodes implanted in the brain or with an array of electrodes affixed to the scalp. Of special interest are experiments with evoked responses, for example, see Siddiqui et al. [SIDD-90c].

The evoked responses or event-related potentials (ERP) result from a stimulus applied to a person. The stimulus can be an electrical pulse, a drug, sound, a (soothing) touch, and so on. The result is a wave representing different aspects of the brain's response.



Following the model we developed in equation 2.3.1, the indications in the EEG type of problems were structured in Figure 2.2. However, the components would have the following interpretation:

W(d) : is the indication produced by the disease, d, in the absence of non-disease artifacts,

W(E) : is the indication (artifact) produced by the equipment, E, and is the sum of artifacts from wave mode, W(mode) and electromechanical artifacts, W(a), that is,

$$W(E) = W(\text{mode}) + W(a)$$

W(C) : is the indication (artifact) produced by the operating conditions, C, of the equipment,

W(P) : is the indication (artifact) produced by the limbs and other organs of the body, P,

W(n) : is the accumulated indication (total noise) produced by the non-disease parameters together,

W(n,d) : is the composite indication and consists of both brain signal and artifact indications.

W(mode) : is the indication of the subject in question under normal (healthy) conditions.

### 2.3.3 PNA Indications Model

A PNA-waveform is an ultra violet visual fluorescence (uv-vis fluorescence) composed of discrete spectrum produced by measuring intensity of emission versus wavelength scanned during a certain time which usually excites with steady state light at a fixed wavelength [EAST-83, SOGL-85]. These spectra, like NDT-signals and EEG waveforms are not noise free. In addition to the usual noise from the equipment and experimentation, peak-jitters, optical noise from the mixture and the solvent blank are some of the common factors which corrupt the quality of the spectra. Adopting the same model

of equation 2.3.1, we developed the model for PNA spectra. The PNA indications are structured in Figure 2.3. The components in equation 2.3.1, however, have the following interpretation:

W(d) : is the indication produced by the PNA-compound, d, in the absence of impurities,

W(E) : is the indication (artifact) produced by the equipment, E, and is the sum of artifacts from wave mode, W(mode) and electromechanical artifacts, W(a), that is,

$$W(E) = W(\text{mode}) + W(a)$$

W(C) : is the indication (artifact) produced by the operating conditions, C, of the equipment,

W(P) : is the indication (artifact) produced by the constituents of the compound, P,

W(n) : is the accumulated indication (total noise) produced by the impurities and other non-concerned constituents present in the compounds,

W(n,d) : is the composite indication and consists of both spectra and artifact indications.

W(mode) : is the indication when running a blank mixture under the same operating conditions.

## 2.4 Ideal Knowledge Requirements

To evaluate the components of equation 2.3.1 and to acquire pertinent knowledge for signal processing and recognition, accumulated effect of all their constituent parameters has to be evaluated. To simplify the task we considered the major noise contributors which are usually known before the experiment - a priori knowledge. These contributors are described below.

A priori knowledge is considered to be problem-domain dependent generic knowledge that is independent of the presence of

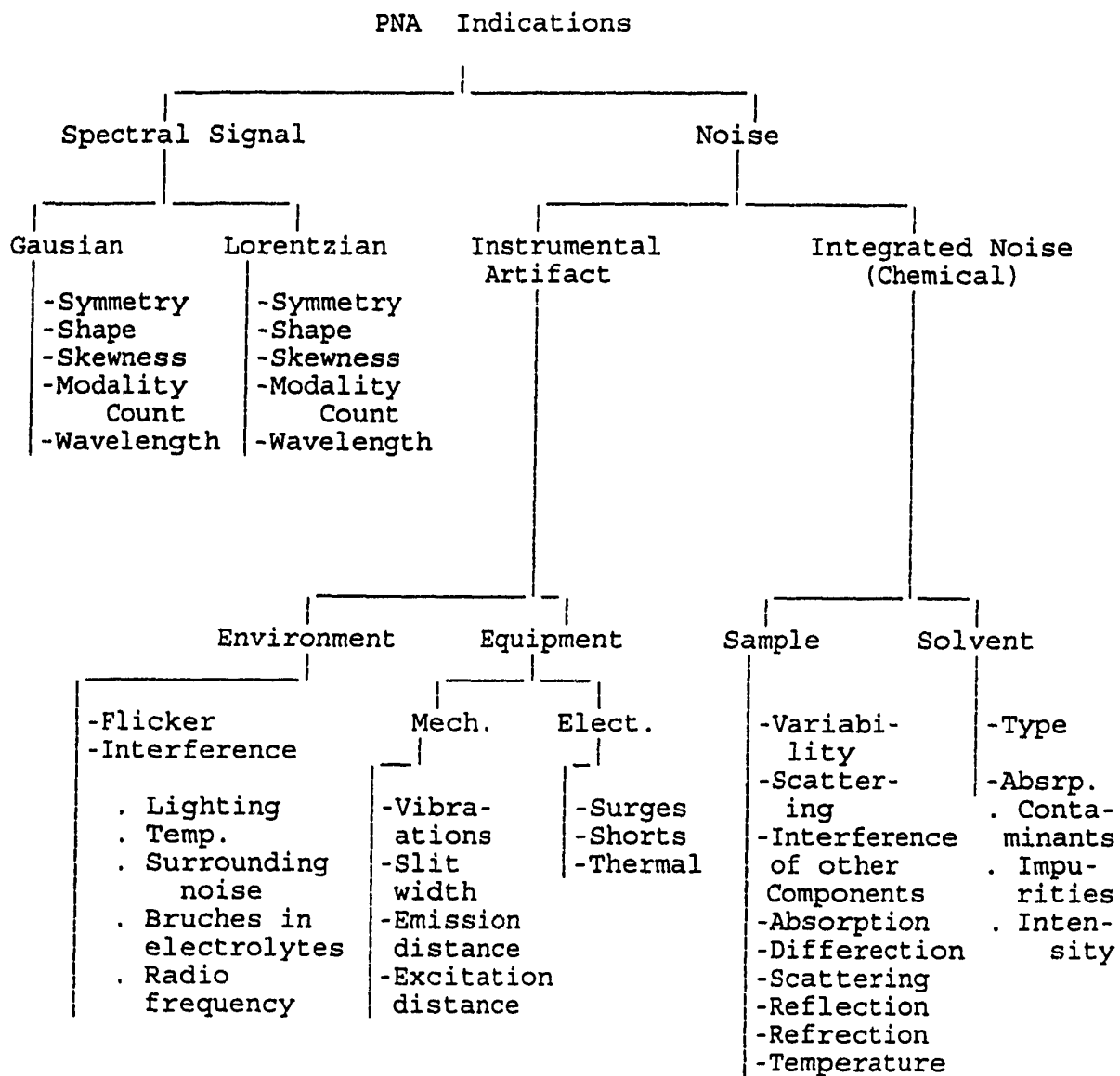


Fig. 2.3: PNA Indications



any pattern class. We defined it as the hard core real world knowledge which can be used for, a) identification of sources of noise and artifacts, b) reduction/elimination of noise contributors, c) preliminary interpretation of signals, d) preparatory arrangements for problem solution and establishment of the scope of the problem.

A defect is a material inhomogeneity which significantly affects the performance of material or component. Considering the factors shown in Fig. 2.1, in NDT-problems, a priori knowledge about the defects could be divided into six basic inputs: 1) domain-specific parameters, 2) method-specific parameters, 3) test specimen characteristics, 4) detailed defect (source) characteristics including location and orientation of defect(s), 5) test equipment characteristics, and, 6) the operating parameter ranges of the test.

The domain specific parameters include the general principles and guidelines that an expert may adopt for, a) describing the pattern classes and label assignment to pattern classes, b) selecting the physical (NDT) method and apparatus for signal acquisition, c) selecting the information physical observations may carry, d) selecting the methodology and algorithms in each phase of solving a problem, and, e) the methods to interpret the results and methods to associate pattern class to a physical phenomena.

To correlate defects with appropriate properties of signals and materials the NDT technology provides a large number of methods. Most commonly used methods are; eddy currents [AULD-83, MARZ-83], ultrasound [KRAU-69, VARY-79], acoustic emission [HAYD-84], and acousto-ultrasound [DESR-86, KAUT-86, VARY-82, VARY-87]. Each method encompasses its own set of principles and characteristics which should be considered when a method or application is selected. The selection of an NDT method is

highly dependent on a large number of factors such as components to be tested, the kind of defects to be identified, the kind of defects a method can determine, etc. These factors are summarized in Table 2.1. A set of theme enquiries designed to establish method versus defect relationship and to determine the kind of knowledge required in the NDT problem domain are shown in Table 2.2.

The NDT-signals also affected by inherent characteristics of the test object. These characteristics include material property data, shape and geometry of the object, and the knowledge pertaining to the defects the specimen may contain. For example, natural materials produce a more or less pronounced effect which usually weaken the propagation of sound. This results from attenuation, which is the sum of scattering, and absorption losses. The studies in material science suggest not measuring the individual effect of waves propagation based on grain [KRAU-69], and geometry or structure [MOYZ-82]. The point, however, to be made is that the inherent properties of the material should not be confused with artifacts or defects. Important characteristics to consider are listed in Table 2.3. An expert, perhaps reviews them in some logical fashion.

Defect characteristics are normally suspected with high certainty by experts or seasoned operators, if a material specimen is presented to them implying that they have sufficiently learned the properties of different defects and can recognize them by simply eye-balling a few samples. A number of physical characteristics of commonly found defects are shown in Table 2.4.

Test equipment characteristics pertaining to initial set-up and calibration of the testing equipment have immediate effect on the quality of signals. It is a well known fact that the pattern recognition techniques are sensitive to an input

signal which is dependent on the characteristics of the system. Proper selection of the test equipment is essential for an acceptable reliability of a system and this would basically improve signal-to-noise ratio by reducing the electrical artifacts, mechanical glitches and other electromechanical artifacts and hence would lead to a high resolution data requiring the least amount of preprocessing and other rectifying measures for improving the quality of data.

Table 2.1

Popular NDT Methods, their Applications and useful Method dependent Parameters

Method	Measures/Detects	Test Specimen/ Application	Method dependent Parameters
Acoustic Emission	Crack initiation & growth rate; internal cracking in welds during cooling, boiling or cavitation; friction or wear; plastic deformation; phase transformations;	Pressure vessels; stressed structures; turbine or gear boxes; fracture; mechanics; weldments;	Transducer must be placed on part's surface; highly ductile material yield low amplitude emissions; part must be stressed; operating test system noise needs filtered out;
Ultra- sonics	internal defects & variations; cracks; lack of fusion; porosity; inclusions; delaminations; lack of bond texturing; thickness;	Wrought metals; welds; brazed joints; adhesive bonded joints; non-metallics; in-service parts;	Couplant required; small, thin, complex parts may be difficult to check; reference standards required; special probes;
Acousto Ultra- sonics	bonded joint defects & strength variations; hidden impact damage; degradation from cyclic fatigue; hydro- thermal; aging; overt flaws; delamination; erosion; corrosion;	Metals; composites; structural composites; ceramic materials; porous metals; fiber glass composites;	broadband transducer; sending & receiving transducer on the same side of the specimen; couplant required; distance between transmitting & receiving transducer, location; amount of pressure applied;

Table 2.2

Theme Rules Describing Problem Domain Information

- 
1. Define the problem and determine the nature of defects (classes) to be identified and also determine the desired extent of the solution(s), that is, the knowledge pertaining to:
    - Number of pattern classes and their identity
    - Number of pattern samples in each class
    - Number of features and their identity that could best describe each class
  2. Observe the methods given in Table 2.1 and determine the method to solve the given problem?
  3. Using the above methodology search through Tables 2.1 through 2.5; identify the knowledge required to solve the problem and search the answers to the questions:
    - What are the known physical properties of the test object?
    - What is the geometry of the test specimen?
    - What potential defects are we looking for?
    - What is the test goal?
    - What test apparatus is available?
    - What are the Operating parameters of the apparatus?
    - What are the operating conditions?
  4. Set up the design data set.
  5. Expert Choices
    - Method to be used at each step of processing
    - Solution strategy
    - Decision parameters and their thresholds
- 

An NDT-test equipment is usually a transducer/pulsar system. As a general guideline a rather strict acceptance criterion is necessary for acquisition of the test equipment. In addition, a transducer acceptance check should be made before acquiring test data on a particular day which involved acquiring a maximized echo from the edge of the component, and then comparing it with the reference echo in time and frequency domains. Necessary factors for proper set-up and calibration of the test equipment are listed in Table 2.5.

Table 2.3

## A priori knowledge - Test Specimen

Material Properties	Specimen Geometry	Visually Confirmed Defects
Material alloy	Dimensions: height, width, length;	Type (Class or Candidate classes)
Material Grade	Shape	Dimensions (size)
Base Material	Surface conditions: degree of roughness;	Shape
Weld,	inspection area:	Orientation
Material Form	plane, cylindrical, machine parts,	Location
Surface conditions: roughness, porosity, coatings	bar-type	Causes of defect (Procedure for producing defects)
Thermomechanical history - grain structure & size		Confidence levels of Observation
Organic structure		Method of actual crack size determination
Anisotropy, Texture		
Density, Acoustic Impedance		

Table 2.4

## Shape Factors of Some Common Defects





	Cracks	Porosity	Slag	Erosion
Shape				
Description	Vertical Planer Sharp edges	Spherical Volumetric smooth edges	Variable Shape Volumetric Sharp/smooth edges	Vertical Planer Concave/convex Sharp edges
size	singular	singular/ clusters	singular	singular

Table 2.5

## Test Apparatus &amp; Test Conditions

Transducer	Other Components	Operating Parameters
size	couplant: type, amount	skip distance;
Frequency	and distribution	motion control:
Material length	wedge angle	auto-indexing,
focal length, if	Reference Block	manual-indexing,
focused		digital-indexing;
Damping media		preparation of part
Frequency spectrum		Operating Temperature
Beam width (2 axes)		Scanning area, angle
Focal length/water		Location of Transducer
path		Incidence angle
Wavelength		Transient angle
emission rate		Sampling period
		Cleanliness
		Surface preparation

Proper selection of equipment goes in hand with its proper usage. As discussed above, proper calibration of the equipment for the type of the experiment at hand is essential for high quality of data. The operating parameters indicative of effects on signals are listed in Table 2.5.

Although it might be ideal to know all the parameters listed in Tables 2.1 through 2.5 encompassing the problem at hand, acquisition of knowledge pertaining to all six areas in addition to the structural properties of the signals would be a formidable task. Therefore, we decided to consider an overall effect of the material properties. Once assured of the proper quality, the signals based on their structure are carefully reviewed. The defects identified can be attributed to decisions later such as retirement of material for cause, remaining life analysis, life extension probability, etc.

## 2.5 The Knowledge - Our Perspective

Traditionally, a knowledge-based system is meant to mimic the decision making process of human experts in a specific problem

domain. It contains large amounts of subtle knowledge of expert(s) organized into a knowledge base yet separate from the decision making process.

In these systems, the knowledge pertaining to a given domain consists of descriptions, relationships, and procedures [JACK-90]. The descriptions which identify and differentiate objects and classes are sentences in some functional or object oriented language such as Lisp, Prolog [CLAR-82], OPS5 [BRWR-85, FORM-77], and KL-One [BRAH-85] whose elementary components consist of either logical constructs or primitive concepts. A description system generally includes rules or procedures for applying and interpreting descriptions in specific applications. A knowledge base also contains particular kinds of descriptions known as relationships. They express dependencies and associations between items in the knowledge base. Typically, such relationships describe taxonomic, definitional and empirical associations. Procedures specify operations to be performed when attempting to reason or solve a problem [HAYE-83].

This traditional concept of knowledge and its representation and organization tends to provide a rigid mechanism which can be constructed only through an exhaustive interaction of a knowledge engineer with expert(s) and creating a knowledge base that could be examined only through the inference engines based on exhaustive or optimal/suboptimal search strategies. Furthermore, any update in the knowledge base would require going back to the knowledge engineer and the expert which in some cases may lead to a major restructure of the system. Unfortunately, the structure of expert systems based on such concepts has now become a commercial standard and a majority of acclaimed expert systems currently available have blindly followed such standards.

In fact, knowledge is an integrated concept which is acquired through the extensive use of six human senses. The five senses of vision, smell, hearing, touch and taste usually help in carrying out the everyday chores. The sixth sense is the human perception and intuition (deep knowledge) which an individual acquires through his experience spanning over his/her lifetime. There are disciplines such as signal processing, computer vision, statistical decision theory and pattern recognition that use a sub-/super-set of these senses implicitly or explicitly, and have techniques available which are rich in abstracting and formalizing domain-specific knowledge concepts and can be used to partially simulate, if not to replace, the human perception.

For example, pattern recognition (PR) techniques have evolved from the human processes of vision, recognition and perception. One of the theories considers pattern recognition as a paradigm-oriented inductive process [WATA-84] that has also been one of the approaches used in training expert systems. This theory suggests that the selection of appropriate PR tools may reduce the time required by the knowledge acquisition process. The use of PR techniques, however, cannot be generalized for acquiring knowledge pertaining to all application areas. They are more useful than other methods in some application areas.

One of the suitable areas, for example, is the problem domain of random signal processing which has applications in material testing, quality assurance and evaluation, medical diagnostic systems, chemometrics, spectroscopy, etc. In these applications, the domain-specific knowledge and the decision parameters used by a human operator or expert can be broken down to very low level primitives. A majority of these knowledge primitives can be obtained automatically using feature extraction methods used in PR and signal processing fields.



However, some of the knowledge components such as intuition, judgement, etc., cannot be directly represented using PR or signal processing methods. There, we can borrow some concepts from statistical decision theory. Human judgement and intuition is primarily based on experience and observations. Knowledge primitives extracted using signal processing and PR techniques, on processing through analytical procedures, would provide some empirical results which can be validated by using decision theory methods. This approach thus provides analytical and empirical means to simulate human judgement. This wholesome perspective of knowledge was then used to design an integrated system for knowledge acquisition, representation, and organization system, hereafter referred to as the KARO subsystem.

## **2.6 The Design of IRS System**

The Intelligent Recognition System (IRS) outlined in Section 1.8 was a formidable task, however, we carefully identified the major components with three objectives in front of us: 1) components constituting the system should be implemented as realistically as possible, 2) conceptually achieve the overall functionality of the original design, 3) consider the components not implemented as "black boxes" at present and that they may be added on later without any restructure of the system.

Our original efforts were concentrated towards the development of the entire system. Acquisition of problem-dependent a priori and heuristic knowledge for four diverse areas of applications was a formidable task, particularly, in situations where we were dealing with commercially sensitive and proprietary applications such as EEG problem and the CEL data (19 class) problem. In addition, the availability of analytically oriented expert in each field of our interest was a

next to impossible task, let alone the transfer of the heuristic knowledge into procedural knowledge. Hence we decided to rely on analytical knowledge and the knowledge derived therefrom.

The architecture developed here consists of only the components that could be automated using analytical, empirical and procedural knowledge and algorithms. However, the design is structured in such a form that expert knowledge and other components dropped at present could be incorporated at a later stage as a black box.

In developing this architecture we basically kept the same structure and the same functionality of the system as described in Section 1.8 with the exception that knowledge will comprise mainly of analytical and empirical knowledge. In addition, we eliminated the expert/user interface, since it is mainly an exhaustive programming exercise, and called the system as intelligent recognition system instead of a knowledge based system.

#### **2.6.1 The Knowledge Acquisition, Representation and Organization (KARO) Subsystem**

The first major component of the recognition system is the Knowledge Acquisition, Representation and Organization (KARO) subsystem. The objective behind the design of this subsystem was to acquire a larger set of knowledge primitives and concepts so that an integrated knowledge base could be developed. The KARO subsystem is a composite of three independent phases, namely, Fact Gathering, Knowledge Base, and Knowledge Formalization and Organization (see Fig. 2.4). The fact gathering phase includes the acquisition of the input data (waveform signals), data preprocessing and the measurements of physical

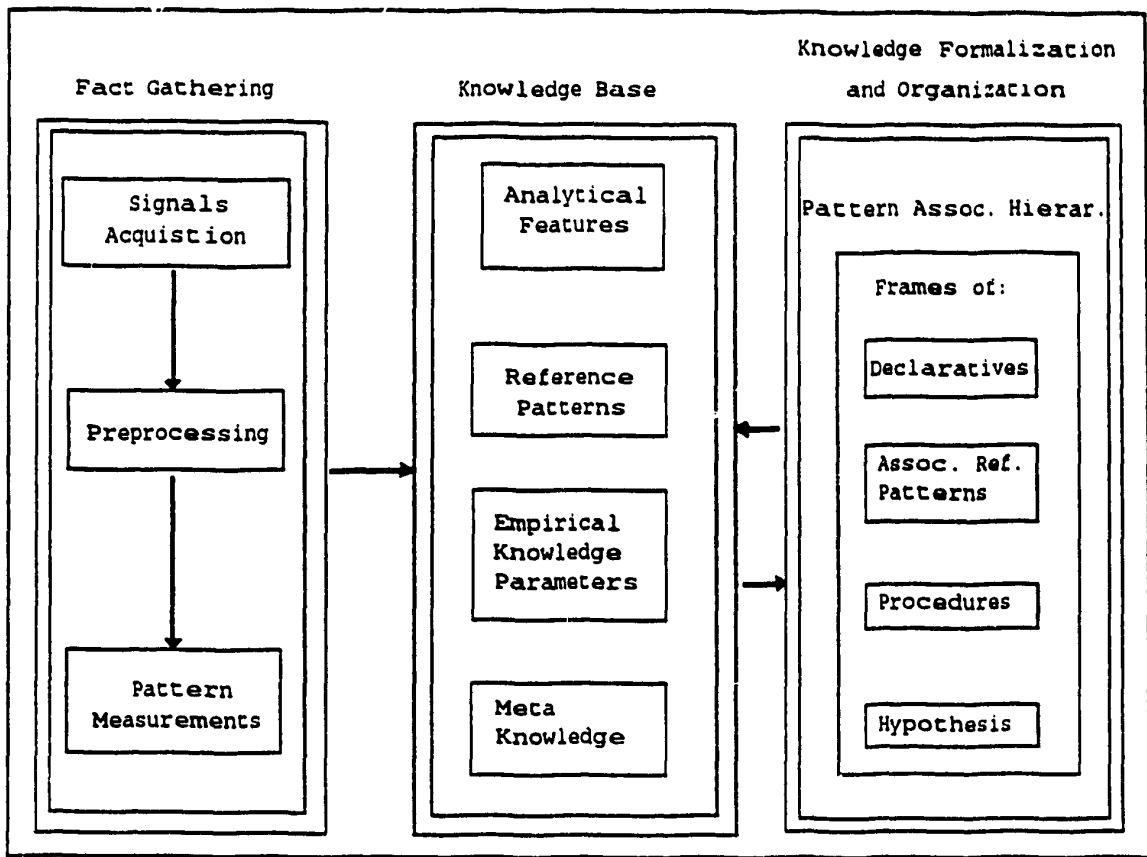


Fig. 2.4: The Knowledge Acquisition, Representation, and Organization (KARO) Subsystem

observations from waveform patterns (feature extraction). The features thus extracted are organized into a knowledge base.

The Knowledge Base is a knowledge storage and houses three types of knowledge: analytical, empirical, and meta-knowledge about the problem-domain. Analytic features are a large subset of pattern measurements performed during the fact gathering phase. Empirical knowledge is derived from the analytical observations and meta knowledge is obtained through a combination of empirical knowledge and statistical inference. Meta-knowledge, in fact, partially simulates expert's judgement.

### **2.6.2 Inference Engine**

The structure of the inference engine was kept the same as discussed in Section 1.8, except that it was trained using supervised learning only. It will still operate in two modes of operation, i.e., executive and consultant. In the consultant mode of operation it will still allow the user to select an algorithm of his/her choice, however, he/she will not be able to train this mode of operation using unsupervised learning.

### **2.7 Signal Conditioning and Treatment**

The model presented in equation 2.3.1 was considered as a composite of several independent terms. Hypothetically by evaluating the six input components one can minimize the effect of a majority of noise contributing parameters, but in practice it cannot be justified as cost effective and a small gain in signal quality may not worth the time and efforts. In addition, the interaction of the factors influencing the components in equation 2.3.1 is not possible to determine exactly. However, using software based conditioning and treatment of the waveforms, the noise factor  $W(a)$  (see equation 2.3.2) can be significantly reduced. In addition, the effect of other parameters, in reality, can be identified if more information is extracted from the physical observations and the parameters listed in Tables 2.1 through 2.5.

The NDT data used in this research was collected using Acousto-ultrasonic (AU) method. The amount of information in the ultrasonic (UT) waveform is not known precisely. However, information theory tells us that the greater the rate of information transmission, the greater is the required bandwidth. Thus a guiding principle in the design of UT or AU-based NDT experiments is to use broadband transducers and

amplifiers. The use of broadband transducers and instruments assures that the system will not limit the information contents of the signals. There is, however, a corresponding increase in the signal noise level due to broader bandwidth. Therefore, additional steps must be taken to determine that signals do, in fact, contain sufficient information on the defect(s) to permit full exploitation.

Defect detection under simultaneous occurrence of interfering effects, discussed in previous sections present a considerable problem in NDT. If a comprehensive approach is to be used to relate signal characteristics to reflector properties, it would be necessary to consider the characteristics of the defect as a function of signal properties. Therefore, before a satisfactory application of a test method it is always necessary to determine the relationship between different occurring indications by means of pilot experiments, i.e., between the magnetic as well as electrical and the mechanical properties of the test objects. It has been suggested that reference samples of known properties have to be available for exact calibration considering the test parameters [MATT-88].

Even after the above stated calibration it is not guaranteed that the signals received are noise free. Cross correlation analysis seems to be promising for testing, because this method compares a selected reference signal with each signal occurring during testing and is giving a measure of the similarity. Additional treatment measures suggested are as follows:

1. Deconvolution: to remove the transducer characteristics from the observed signal.
2. Spatial Averaging: to reduce the spatial resolution and the effect of grain scattering.
3. Stationary (i.e., temporal) Averaging: to remove noise

mainly from amplifier and other random events and to improve signal-to-noise ratio.

4. Calibration for Maximum Defect Response: by jogging the probe in both axial and circumferential positions determine the maximum defect response and repeat the process on all visually identifiable defects [ORR-79]. The associated position coordinates of the transducer were then noted and the same should be used for data collection.

The conditioning and treatment on NDT data was performed by the staff at the Tektrend Int., Montreal. The preprocessing performed on PNA data is reported in [SIDD-91a].

## **2.8 Mapping and Parameterization of Waveforms**

From the previous discussion it became clear that the time waveform alone is not sufficient for analysis. For flexible and reliable machine processing, and analysis additional information is required. In the field of signal processing it is known that from preprocessed time waveforms, a number of other domains (mapping spaces) can be generated to provide a more illustrative representation of the available information. Siddiqui et al. suggested a pattern measurement system in [SIDD-90a] and this is reproduced in Fig. 2.5. The important feature of the system is the transformation switch whose function is to transform a waveform into other information domains and this can be done by using a number of suitable transformation techniques available in the field of signal processing [BRAC-86]. One such approach is described in the following section.

### **2.8.1 Mapping Space**

Although there are a number of transformation techniques available in the field of signal processing and information theory, the Fourier transformation has been used extensively in waveform analysis [BRAC-86].

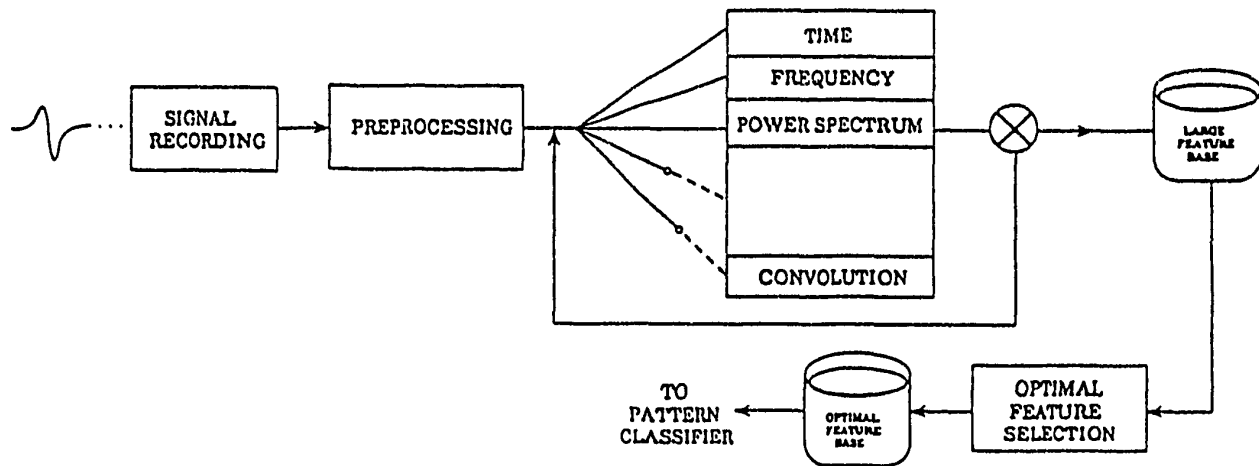


Fig. 2.5: Pattern Measurement System [SIDD-90a]

The time varying signals, their transformations and the functions derived therefrom such as a spectrum, correlation function or cepstrum, can be processed further to extract the information contents. A formal discussion on waveform transformation now follows.

Formally, we defined a signal as: "a sequence of discrete real and/or complex numbers and is a time function,  $x(t)$ . The value of the function at any time  $t_i$ ,  $x(t_i)$ , is a random variable."

The variable  $t$  is chosen since the signals that are considered here are time-dependent signals. We assume that we have a source which generates the function  $x(t)$ , which is denoted as

a sample function. The source generates  $N$  sample functions which together are known as an ensemble. At any time,  $t_1$ , we observed the values of all sample functions, to get many "results of experiments." For example, consider the NDT signal taken by means of a surface transducer located at a certain location on a piece of material. We want to investigate the properties of the source material through the NDT signals.

Thus a data sequence consisting of  $N$  samples of signal  $x(t)$  is given by:

$$\begin{aligned} \text{samples of } x(t) &= [ x_k ] & \dots 2.8.1 \\ &= [ x_0, x_1, \dots, x_{N-1} ] \end{aligned}$$

where  $k$  is a time index which ranges from 0 to  $N-1$ . The discrete Fourier transform (DFT) of  $[ x_k ]$  consists of  $(N/2 + 1)$  complex samples (assume that  $N$  is even) given by

$$\begin{aligned} [ X_m ] &= \text{DFT} [ x_m ] & \dots 2.8.2 \\ &= [ X_0, X_1, \dots, X_{N/2} ] \end{aligned}$$

where  $[ X_m ]$  is the DFT of  $[ x_k ]$  and that the index  $m$  designates the frequency of each component  $X_m$ .  $X_m$  can also be represented in polar coordinates as:

$$X_m = | X_m | \exp (j \theta_m) \quad \dots 2.8.3$$

where  $| X_m |$  is the amplitude of  $X_m$ , it is equal to

$$| X_m | = [ R^2(x) + I^2(x) ]^{1/2} \quad \dots 2.8.4$$

where  $R(x)$  and  $I(x)$  are the real and imaginary components of the transform. The square of the spectrum  $| X_m |$  is commonly referred to as its power and is denoted as,

$$\begin{aligned} P(x) &= | X_m |^2 \\ &= R^2(x) + I^2(x) & \dots 2.8.5 \end{aligned}$$



The relationship implemented by the transform between  $[x_k]$  and  $[X_m]$  can be expressed as:

$$X_m = \sum_{k=0}^{N-1} x_k \exp(-j(2\pi m k/N)) \quad \dots 2.8.6$$

for  $m = 0, 1, \dots, N/2$

In this formula  $X_m$  is an exponential function and it is complex sinusoidal and periodic. These characteristics will be more clearly represented by separating the real and imaginary parts,

$$X_m = \sum_{k=0}^{N-1} x_k \cos(2\pi m k/N) - j \sum_{k=0}^{N-1} x_k \sin(2\pi m k/N)$$

for  $m = 0, 1, \dots, N/2$

... 2.8.7

The equations 2.8.2 through 2.8.7 were used to derive a number of information domains wherein the analytical features in a problem domain at hand can be measured. The information domains that this study includes are described below:

- Time waveform: The original time waveform, given by equation 2.8.1 (raw time versus amplitude signals),
- Frequency waveform: Real components of the complex exponential in the DFT, equation 2.8.6 (Fourier transform of the time waveform),
- Phase spectrum: The Fourier transform also produces a value of phase for each particular frequency in the spectrum of signal which is useful in determining the ratio of stored energy to that dissipated in the system. Phase spectrum of  $[x_k]$  is obtained by plotting  $\theta_m$  against  $m$  (equation 2.8.3),
- Power Spectrum: The square of the spectrum given in equation 2.8.5, which is the plot of the power against each frequency component of the transform,
- Auto Correlation: This domain is useful in determining whether a signal is periodic or cyclic. The time

domain signal given by equation 2.8.1 is compared with itself at different positions (time) and the similarity between signal segments is determined,

- Cepstrum: Using a Fourier transform of the logarithm of the power spectrum given by equation 2.8.5),
- Log Power Spectrum: Taking the logarithm of the power spectrum,
- Convolution: Taking the inverse Fourier transform of the product of Fourier transform of two sample signals from a pattern class.

### 2.8.2 Parameter Extraction

It is difficult to measure the pulse shape descriptors directly from the above waveforms or spectra. Alternately, an envelope is constructed from which the pulse information is extracted. The literature [HAYD-88, SEDG-88] provides several standard signal envelope extraction algorithms to facilitate these operations and the feature extraction process. One such algorithm is outlined below:

1. A convex shape with finite arc length is used to construct a rough convex hull of the given signal profile.
2. Smoothing is then performed to obtain the desired signal envelope.
3. A set of typical features listed in Table 2.6 were then extracted from the envelope. Figure 2.6 shows their geometrical interpretation.

The feature extraction process essentially measures a number of statistical, waveform, geometrical, absolute and shape features in a number of selected domains from the list described above. These features, generically, are listed in Table 2.6. The choice of a transformation domain is again a problem-dependent activity. For different types of data different domains and features were used. The set of

procedures that were used to measure these features are described in Appendix A.

Table - 2.6

Analytical knowledge features used

Statistical	Waveform	Geometrical	Absolute	Shape
mean, standard deviation, higher order moments, maxima, minima	impulse sum in different data windows	peak location, rise time, fall time, rise slope, fall slope, peak width	skewness, kurtosis, no. peaks, full pulse, half pulse	amplitude, area, weighted area, energy

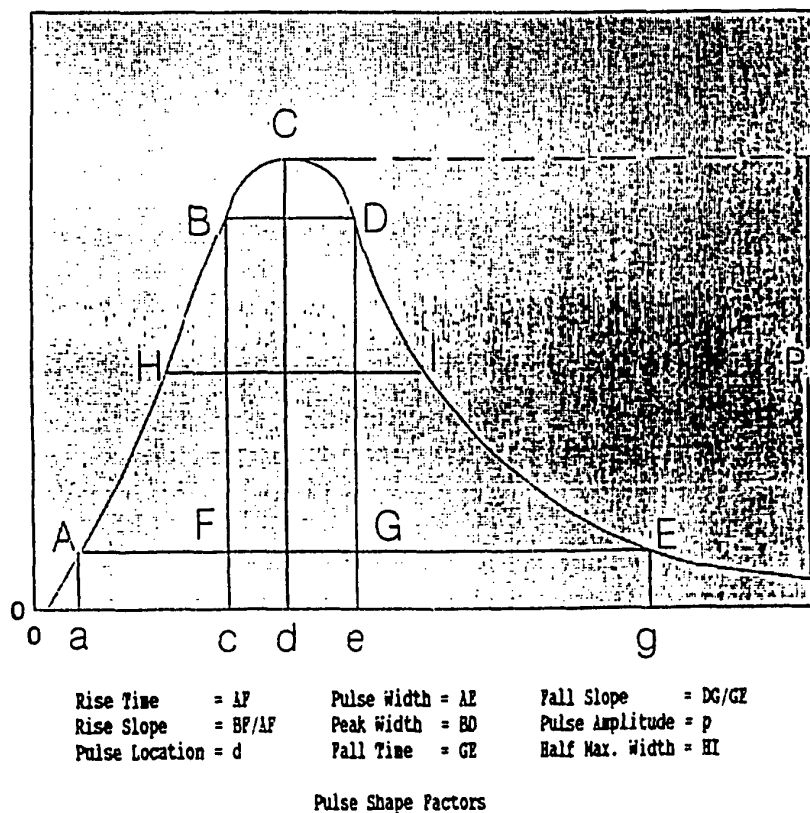


Fig. 2.6: Geometrical Interpretation of Envelope and Features Derived Therefrom

## Chapter 3

### Analytical Features and Pattern Association Hierarchy

#### 3.1. Introduction

Analytical features constitute a major subset of pattern measurements and they may be large enough to prohibit exhaustive analysis. To process them with good performance (recognition) a hierarchy of naturally associated pattern classes was developed. The fact gathering phase and the techniques to develop a pattern association hierarchy are described in this chapter.

#### 3.2 Fact Gathering Phase

This phase includes three components, namely, signal acquisition, preprocessing, and pattern measurements. Signal acquisition and preprocessing are not among the mainstream of the subjects we studied in this thesis. Both of these steps were performed by the institutions who have provided their respective data sets. Tektrend International, Inc. of Montreal supplied the data on NDT-signals, EEG signals, and cell data in the form of feature vectors; the Lockheed-ESC, located in Las Vegas, furnished the digitized data on polynuclear aromatic compounds (petroleum oils). These data sets will be referred to as NDT-data, EEG-data, CEL-data, and PNA-data in the following chapters.

Brief descriptions of instrumentation and experimental conditions underlying data collection for each problem domain are presented in Appendix B.

### 3.2.1 The Pattern Data

The algorithms we developed were tested on problems from four different areas; classification of material defects non-destructively (NDT-data), the classification of electroencephalogram signals (EEG-data), classification of polynuclear aromatic compounds (PNA-data), and classification of body cells (CEL-data). The characteristics of the latter three data are described in Appendix B, whereas the characteristics of the NDT-data are described below.

NDT data were collected by applying the ultrasonic signals to mild steel bars of measurements 0.5 inch thick, 1.0 inch wide, and approximately 2 feet long, wherein slots of different depths and lengths were artificially machined to simulate varying degrees of wall erosion. A total of nine flaws of different lengths were introduced into the bars. In addition, to acquire signals representative of no flaw condition, a bar without flaw was also tested. The dimensions of each slot machined into the bar and their corresponding flaws are shown in Table 3.1. A micro-computer controlled acousto-ultrasonic data acquisition system ARIUS [LACA-85, MATT-89] was then used to acquire, digitize, process and store the signals. A total of 400 data files were created, 40 for each of the nine flaw types and 40 for the flawless bar. Each data file consisted of 2048 data points. Four sample waveforms representing four typical pattern classes are shown in Figures 3.1a through 3.1d.

### 3.2.2 Pattern Measurement Problems

A central problem in using PR techniques is that of extracting from the pattern the information (features) which is most relevant for classification. If effective features have been obtained, then, the pattern classification problem becomes one

of partitioning the feature space into regions, one region for each class. Selection of properties that contain the most discriminatory information is important because the cost of decision making is considered directly related to the number of features used in the decision rules. Thus for large applications when the complexity of the problem increases, it becomes especially important to develop methods for efficient design of feature selection and classification algorithms.

Selection of features strongly affects the design of a classifier. That is, if the features show significant differences from one class to another, the classifier can be designed more easily with better performance. Therefore, the selection of features remains a key issue in PR.

Table 3.1

Sizes of Defect Areas and Their Identification

	length	76mm (3")	152mm (6")	228mm (9")
depth				
0.25mm (0.01")	sms	sh (A)	mesh (D)	lash (G)
1.52mm (0.06")	sm	me (B)	meme (E)	lame (H)
3.18mm (0.125")	sm	de (C)	mede (F)	lade (I)
		no (J)		

Legend:

length => la: large (9"), me: medium (6"),  
sm: small (3")

depth => de: deep (.125"), me: medium (.06"),  
sh: shallow (.01")

no: no flaw

Example: meme: a class of defect of length 152mm (6")  
at depth 1.52 mm (0.06")

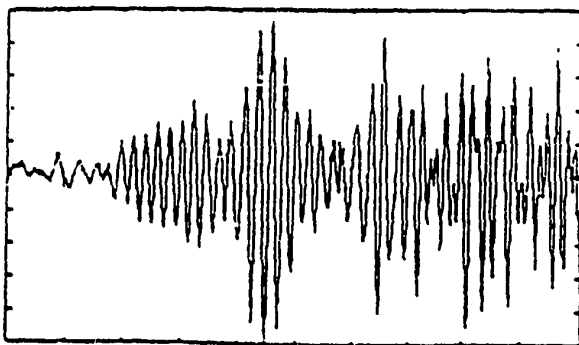


Fig. 3.1a: A typical waveform from a flawless bar (class J)

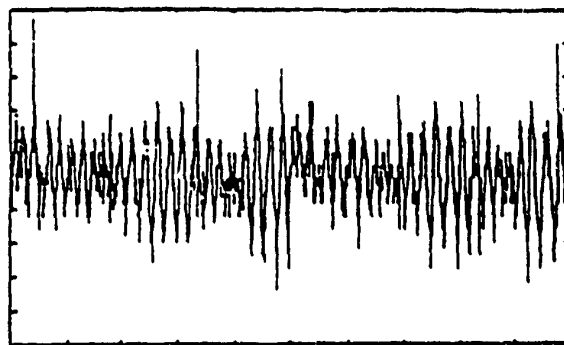


Fig. 3.1 b: A typical waveform representing a smsh pattern (class A)

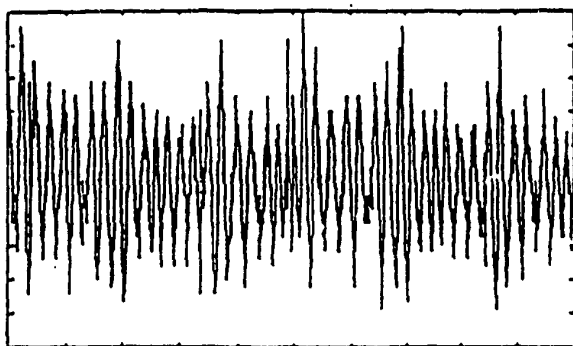


Fig. 3.1c: A typical waveform representing a meme pattern (class E)

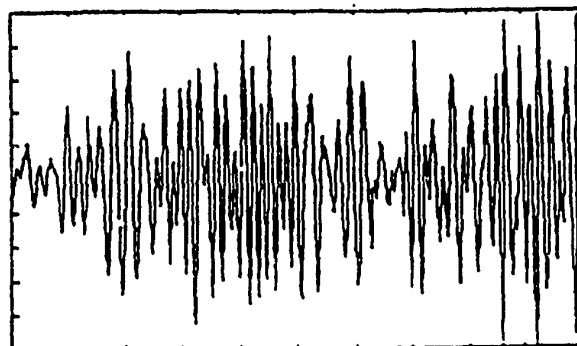


Fig. 3.1 d: A typical waveform representing a lade pattern (class I)

Legend:

Horizontal Scale: 12.8  $\mu$ s/div.  
Vertical Scale: 5.14 units/div.  
Data Width : 2048 points

Fig. 3.1: A few typical samples from NDT signals.

The more complex the pattern we are dealing with, the more difficult it is to decide what the important measurements are. The approach we adopted to get around this difficulty is to collect whatever specific knowledge about the problem has been suggested in the literature and by the practicing experts [HAYD-87, MATT-88]. In addition, we included all those measurements that, in our opinion, could possibly provide additional valuable information.

An increase in the number of measurements that resulted from this approach brings about an increasingly complex classifier structure. Also the larger the feature set (compared to number of classes) the greater the possibility that a number of irrelevant and redundant features may have been selected. The presence of such redundancies in the input data detrimentally affects the reliability of the classifier.

To minimize these problems, several measures for feature dimensionality reduction and optimization were employed. These measures are described in Chapter 4. The classifier design complexity is reduced by organizing the pattern classes using a pattern association hierarchy, a concept introduced in Section 3.3. The PAH concept is also used to resolve the information redundancy and explosion problems (see Chapter 4).

#### **3.2.2.1 Analytical Feature Extraction**

Considering the pattern measurement issues discussed above, a set of features, which are referred to as analytical features were extracted from an array of waveforms in a class of problems, such as NDT signal classification.

Based on the problem domain a set of information domains given in equations 2.8.2 through 2.8.7 were used to measure a number of analytical features discussed in Section 2.8.2. The choice



of a transformation domain is a problem dependent activity. For different types of data different domains and features were used. The information domains that were selected for NDT-problems include, time waveform, phase spectrum, power spectrum, auto correlation, and cepstrum [HAYD-88].

The feature extraction process, essentially measures a number of statistical, waveform, geometrical, absolute and shape features in a number of selected domains from the list described in Section 2.8.2. The set of procedures that were used to measure these features are described in Appendix A. The procedures adopted for the measurement of the features from NDT signals are described in [HAYD-88]. The list of features provided by Tektrend for the NDT data is presented in Table 3.2.

The scheme we adopted conceivably captures a large number of features among which only a few may be needed. The few which will be most suitable for classification, were identified using feature selection algorithms described in Chapter 4. A number of empirical features were estimated using the analytical features. The empirical features, in turn, were used in deriving the components of meta knowledge (see Chapter 4).

#### **3.2.2.2. Homogenizing the Analytical Features**

Each analytical pattern measurement varies both in unit and the range of magnitude from pattern to pattern. However, the result of analysis (or classification) should be independent of these variations. Rather than trying to develop an analysis algorithm which tolerates these variations, it seems reasonable to eliminate them by proper normalization prior to analysis. Thus it is expected that analysis will become easier if parameter variations are eliminated in advance.

Table - 3.2

Analytical Features extracted from NDT signal data [HAYD-88]					
Feature	TM	PW	PH	CP	AC
Description	Feature number by domain				
peaks above baseline	1	37	55	73	91
# of peaks above 10% max. ampl.	2	38	56	74	92
# of peaks above 25% max. ampl.	3	39	57	75	93
Greatest peak position	4	40	58	76	94
Greatest peak amplitude	5	41	59	77	95
2nd Greatest peak position	6	42	60	78	96
2nd Greatest peak amplitude	7	43	61	79	97
3rd Greatest peak position	8				
3rd Greatest peak amplitude	9				
% of total area under 1st peak	10	44	62	80	98
% of total area under 2nd peak	11	45	63	81	99
% of total area under 3rd peak	12				
inter-peak distance: 1st to 2nd	13	46	64	82	100
inter-peak distance: 1st to 3rd	14				
inter-peak distance: 2nd to 3rd	15				
Greatest peak rise time	16				
Greatest peak rise slope	17				
Greatest peak fall time	18				
Greatest peak fall slope	19				
Greatest peak full pulse width	20				
Greatest peak half pulse width	21				
Greatest peak width	22				
2nd Greatest peak rise time	23				
2nd Greatest peak rise slope	24				
2nd Greatest peak fall time	25				
2nd Greatest peak fall slope	26				
2nd Greatest peak full pulse width	27				
2nd Greatest peak half pulse width	28				
2nd Greatest peak width	29				
3rd Greatest peak rise time	30				
3rd Greatest peak rise slope	31				
3rd Greatest peak fall time	32				
3rd Greatest peak fall slope	33				
3rd Greatest peak full pulse width	34				
3rd Greatest peak half pulse width	35				
3rd Greatest peak width	36				
% of partial power in 1st Octant		47	65	83	101
% of partial power in 2nd Octant		48	66	84	102
% of partial power in 3rd Octant		49	67	85	103
% of partial power in 4th Octant		50	68	86	104
% of partial power in 5th Octant		51	69	87	105
% of partial power in 6th Octant		52	70	88	106
% of partial power in 7th Octant		53	71	89	107
% of partial power in 8th Octant		54	72	90	108

Domain Key:

TM:Time, PW:Power, PH:Phase, CP:Cepstrum, AC:Autocorrelation

The principal idea in equalization is to remove the artifacts of the measurement unit and convert each variable to some common numerical property. Disposing of the measurement unit involves dividing all the scores for a variable by a suitable equalizing factor expressed in some units. The sense in which the variables are equalized depends on the nature of data and the kind of equalizing factor chosen.

We investigated several such homogenizing factors. A number of homogenizing techniques described below, have been applied depending upon the nature of data. These techniques either have been borrowed directly from the statistical theory, or developed here using underlying statistical theory [ANDE-73].

#### Standard Normal Scheme

One such homogenizing technique which has been most commonly applied, uses zero mean and unit variance transformation and is given by:

$$x_{ij} = (x'_{ij} - \bar{x}_j) / s_j \quad \dots 3.2.1$$

where

$x_{ij}$  = transformed value of the  $j$ -th feature of the  $i$ -th pattern

$x'_{ij}$  = original value of the  $j$ -th feature of the  $i$ -th pattern

$\bar{x}_j$  = average of  $x'_{ij}$  over the whole design set

$s_j$  = standard deviation of  $x'_{ij}$  over the whole design set

This normalization scheme has the serious effect of diluting the differences between classes on the variables which may be the best discriminators. To overcome this problem it was found more effective to standardize using within-class standard deviations. Thus features were homogenized using this modified standardizing technique.

### Range Normalization

Next approach we adopted for normalization was to convert the data into proportions:

$$x_{ij} = (x'_{ij} - x_{\min}) / (x_{\max} - x_{\min}) \quad \dots 3.2.2$$

where

$x_{ij}$ ,  $x'_{ij}$  = have same meaning as given in equation 3.2.1

$x_{\min}$  = minimum value of  $x'_{ij}$  over all design set pertaining to a class

$x_{\max}$  = maximum value of  $x'_{ij}$  over all design set pertaining to a class

### Value Transformation

In PR applications, the features are assumed independent; in reality it may not be the case. The vast variations within pattern classes dictate that their distribution is non-normal. To make up for such situations a transformation is selected so that the distribution of the transformed features is sufficiently close to normal. The logarithm stabilizing transformation was performed on the NDT-data, i.e., each feature is transformed to its logarithm, i.e.,

$$x_{ij} = \log ( x'_{ij} )$$

### Elimination of Outliers

A problem inherent to EEG brainwaves and NDT-signals classification is the existence of outliers. If the variations among a few sample patterns and the rest of the samples are very large then we choose to substitute these outliers. To identify an outlier a discordancy test was performed. Comparing the individual feature vectors pertaining to a class with their respective class means, the outliers were detected.

The test was conducted for each pattern class. The test statistic with 95% confidence interval used is the internal extreme deviation from the mean:

$$(x_j - \bar{x}) / s < 2.56 \quad \dots 3.2.3$$

where  $\bar{x}$  is the maximum likelihood estimate for the mean of each feature element, and  $s$  is the estimate of its standard deviation, i.e.,

$$\bar{x} = 1/p_i \sum_{j=1}^{p_i} x_j$$

$$\text{and } s^2 = 1/(p_i - 1) \sum_{j=1}^{p_i} (x_j - \bar{x})^2 \quad \dots 3.2.4$$

for  $i=1, \dots, N$

for a set of signals having  $p_i$  samples in class  $i$ . Note that,

$$p_o = \sum_{i=1}^N p_i \quad \dots 3.2.5$$

Once identified the outlier is collapsed to its nearest neighbor that is counted twice in the new unweighted vector. This scheme thus produces a more homogeneous set of samples within each pattern class without losing much information, that is the number of samples within a class remain the same. Upon the resolution of the outliers in the design set, one of the normalization schemes described above is used.

After normalization the first question to ask is whether the correct variables have been chosen in the sense that they are relevant to the type of classification being sought. A further problem, in general, is that of the number of variables measured on each pattern. Since the amount of computer time taken increases dramatically with an increase in the number of variables, it is desirable to seek to reduce the

number of variables before using the classification scheme. This problem of dimensionality reduction and the appropriateness of the variables are addressed in Chapter 4.

### 3.3 Pattern Association Hierarchy

At this point we assumed that the identities of the pattern classes in a problem domain are either known or are being determined by the expert during the initial design phase of the system (see Section 3.4).

Thus assume that  $S$  is a signal and let  $X = [x_1, x_2, x_3, \dots, x_n]^T$  be the result of  $n$  measurements on  $S$  where each  $x_i$  is a measurement or a feature. Assume that several samples of  $S$  are available and that their label or identity is known. A portion of this known data set is considered as the design set,  $Z$  which is used for machine learning (see Chapter 5). Let  $Z = \{X_1, X_2, X_3, \dots, X_{p_0}\}$  be the set of  $p_0$  of these  $n$ -features pattern  $X_j$ , for  $j=1, \dots, p_0$ .

A representation is a label which designates a pattern. Assume that a function  $h$  obtains such representation. Thus each  $X_j$  is a representation of signal  $S$ .

From design set,  $Z$ , and the prior knowledge of the initial description of pattern classes, the function  $g$  gives a "symbolic" description which abstracts the information. Abstraction implies suppressing the details that are not essential while keeping the important properties. For instance, all the patterns of a cluster or class will be described or represented by the same symbolic representation, which may reduce to a name, the same name for all samples in a pattern class (or cluster).

Any representation may be associated with at least one interpretation. Assume that there are  $N$  possible interpretations on the design set  $Z$ , i.e.,

$$\Omega : \{\omega_1, \omega_2, \omega_3, \dots, \omega_N\} \quad \dots 3.3.1$$

An identification is a mapping  $f$ , which may or may not be defined on all the space  $X$ ; or

$$f: X \rightarrow \Omega \quad \dots 3.3.2$$

$$\text{or} \quad : [x_1, x_2, x_3, \dots, x_n]^T \rightarrow \omega_j$$

Such an ' $f$ ' defines equivalence classes  $C_i$  on the space  $X$  or the set  $Z$ , thus

$$C_i = \{X | f: X \rightarrow \omega_i\} \quad \dots 3.3.3$$

All the samples of a group (cluster) or a class  $C_i$ , having the same name  $\omega_i$ , are interpreted as the different occurrences of the same kind (class) of signal  $S$ .

To group the classes in the design set  $Z$  so that not only the processing time is reduced but also the efficiency in information organization and retrieval is obtained, a concept called pattern association hierarchy (PAH) is introduced. The PAH concept is defined by three cooperating functions,  $h$ ,  $g$ , and  $f$ , i.e.,

$$\text{PAH: } \{h, g, f\} \quad \dots 3.3.4$$

each of which represents the following:

- i) *The function  $h$  as shown above describes the signals as feature vectors pertaining to a particular problem domain.*

This description is represented:

- by a set of sample patterns belonging to the design set  $Z$ ,

- by similarity (or distance) measure between patterns computed from the design set.
- by known labels  $\omega_i$  given to the patterns in the design set.

The domain of  $h$  is the space  $p_o$  of objects (total number of samples).  $p_o$  is defined in equation 3.2.5 with  $N$  number of pattern classes and  $p_i$  sample patterns in class  $i$ .

*ii) The function  $g$  gives the symbolic description of a class, and,*

*iii) The function  $f$  produces the hierarchy of associated patterns.*

From the design set and the symbolic description ' $g$ ' a hierarchy of associated pattern classes is obtained. This means that all the objects  $p_o$  of the domain of  $h$  have names  $C$ 's from a finite set of names  $\omega$ . An effective procedure that will be referred to as clustering procedure performs the mapping  $f: X \rightarrow C$ . The groups  $G_i$ , determined by mapping function  $f$  on  $Z$  are called clusters. The set of  $G_i$  is called a partition  $G$ :

$$G = \{G_1, G_2, \dots, G_k\}$$

The properties of  $Z$  allow us to compute from the different patterns a similarity measure between patterns.

### Similarity (or distance) Measures

The function  $h$  uses a similarity (or distance) function. We will define this function. Let us specify the notations we will be using.

Let  $X_i$  be the  $i$ -th class and  $X_{i1}, X_{i2}, \dots, X_{ip_i}$  be  $p_i$  samples belonging to the  $i$ -th class in the design set. Each sample  $X_{ij}$



is represented by  $n$  measurements,  $x_{ij1}, x_{ij2}, \dots, x_{ijn}$ . Let  $Y_j$ ,  $j=1, \dots, n$  be the list of measurements performed on each pattern. Thus the patterns in the design set are represented as:

Class	feature:Y Sample	$Y_1$	$Y_2$	$\dots$	$Y_n$
$X_1$	$X_{11}$	$x_{111}$	$x_{112}$	$\dots$	$x_{11n}$
	$X_{12}$	$x_{121}$	$x_{122}$	$\dots$	$x_{12n}$
	.	.	.	$\dots$	.
	$X_{1p-1}$	$x_{1p-1}$	$x_{1p-2}$	$\dots$	$x_{1p-n}$
$X_2$	$X_{21}$	$x_{211}$	$x_{212}$	$\dots$	$x_{21n}$
	$X_{22}$	$x_{221}$	$x_{222}$	$\dots$	$x_{22n}$
	.	.	.	$\dots$	.
	$X_{2p-2}$	$x_{2p-1}$	$x_{2p-2}$	$\dots$	$x_{2p-n}$
.	.	.	.	$\dots$	.
.	.	.	.	$\dots$	.
$X_N$	$X_{N1}$	$x_{N11}$	$x_{N12}$	$\dots$	$x_{N1n}$
	$X_{N2}$	$x_{N12}$	$x_{N22}$	$\dots$	$x_{N2n}$
	.	.	.	$\dots$	.
	$X_{Np-N}$	$x_{Np-1}$	$x_{Np-2}$	$\dots$	$x_{Np-n}$

A similarity measure  $\rho$  (or distance measure  $d$ ) gives a numerical value to the notion of closeness (or farness) between two pattern classes  $X_q, X_r$ , for  $q, r = 1, 2, \dots, N$ ; and  $q \neq r$  form the design set.

$\rho (X_q, X_r)$  [or  $d (X_q, X_r)$ ] is a real valued symmetric function whose domain is the set of possible class pairs. A high value of  $\rho$  (or  $d$ ) indicates high similarity (or farness).

A distance  $d(X_q, X_r)$  is a real valued, symmetric function, which obeys the three axioms of reflexivity, symmetry and triangular inequality, and whose domain is all pairs of classes. Thus,

$d(q, q) = 0; \quad d(r, r) = 0 \quad \dots$  according to reflexivity

$d(q, r) = d(r, q) \quad \dots$  according to symmetry

$d(q, r) \leq d(q, k) + d(k, r) \quad \dots$  according to triangle inequality

Many association (similarity or farness) measures have been proposed between two objects  $X_q, X_r$  [DUDA-73]. Correlation is usually used to represent the similarity whereas the distance is a measure of farness. The fundamental purpose of a distance or similarity measure is to induce an order on the set of couples  $(X_q, X_r)$  for any objects (classes/groups)  $q$  and  $r$ . In fact, simplicity, calculability and objectivity help to guide the specialist in selecting a clustering algorithm. He/she tries to choose a function which seems to be reasonable, according to what he/she knows of the properties of population  $U$ . In Section 3.4.4, we attempt to provide an objective solution to the algorithm selection issue.

In order to keep maximum flexibility without outgrowing the computational complexity we included both kinds of association measures, that is, similarity and the distance. The similarity measure used is borrowed from [DUDA-73] and is shown in Fig. 3.4. In the estimation of the distance data-dependent parameters decide whether to use a linear (Euclidean) distance or a quadratic distance (Mahalanobis).

The quadratic distance has advantages over linear measure, that it allows to consider correlations between features as well. When correlations are zero, it is equivalent to the square of Euclidean distance measured using standardized variables (features). The Euclidean distance  $d(q, r)$  between classes  $q$  and  $r$  is given by:

$$\begin{aligned}
 d(q,r) &= [ (X_q - X_r)^2 ]^{1/2} \\
 &= [(X_q - X_r)^T (X_q - X_r)]^{1/2} \quad \dots 3.3.1
 \end{aligned}$$

and the Mahalanobis distance is given by

$$d(q,r) = (X_q - X_r)^T S^{-1} (X_q - X_r) \quad \dots 3.3.2$$

where

$S$  = pooled within class (group) variance-covariance matrix,  
and

$X_q, X_r$  = are pattern vectors belonging to classes  $q$  and  $r$   
respectively

The purpose of clustering is to obtain a hierarchical fusion (or partition) of a set  $Z$  of  $p_0$  objects by the use of an association measure.

### 3.4 Clustering Procedures

In this section the main clustering functions (algorithms)  $f$ 's are described. There is a large variety of clustering algorithms [ANDE-73,SIDD-87b]. At start, the  $p_0$  objects  $X_i$ 's are known by their measures (features). The goal of the clustering process is to define a mapping. To give order to a large number of clustering algorithms each with so many variations, we classified them according to the way they fuse (or split) the classes or groups to develop a hierarchy (PAH). All of these methods start with an initial description:

1. A design set  $Z$  of patterns  $X_m$ , for  $m=1, \dots, p_0$ ; each  $X_m$  being a list of measurements  $x_{ijk}$ ;  $i, j, k$  respectively designate the class, sample and feature; and that their domains are known.
2. A triangular measure matrix, each measure being an association (similarity or distance) between  $X_q$ , and  $X_r$  of  $Z$  (see Fig. 3.5).

From this initial description, a clustering algorithm gives a PAH (symbolic description).

For the sake of efficiency the clustering procedures 'f' used here are hierarchical clustering algorithms. These algorithms can be subdivided into agglomerative methods which proceed by a series of successive fusions of the  $N$  classes into groups until one single group is reached, and divisive methods which partition the set of  $N$  individual pattern classes successively into finer partitions until no further partition is possible. The agglomerative procedures were adopted when the identity of the pattern classes was already known whereas the divisive procedures were adopted when the identity of the classes was not known. The results of both agglomerative and divisive techniques can be presented in a tree-like diagram illustrating the fusions or partitions which have been made at each successive level (see Fig. 3.2). The clustering techniques suitable for each of these approaches are described in detail in the following sections.

#### **3.4.1 Bottom-up Organization**

Using this scheme the features and their associated class(es) are organized as a tree in a bottom-up fashion. The guiding principle is based on the fact that the association index (distance or similarity) between two classes in the feature-space directly reflects the closeness (association) of the two classes.

The method considers first the  $N$  classes as  $N$  clusters in the feature space and then gradually merges them to form, finally, one group containing all the  $N$ -clusters. In this procedure the two clusters with the maximum inter-class similarity (or minimum distance) are merged earlier. Each operation of merging two classes and/or groups results in a new group and after  $N-1$

operations, N clusters become one group and the tree structure is obtained.

A large variety of clustering algorithms using both similarity and distance can be used to organize the tree. Following Lance and William's [LANC-67] general algorithm, the distance measure between a group k and a group (ij) is computed by using a recurrence formula:

$$d(k,ij) = a_i d(k,i) + a_j d(k,j) + b d(i,j) + c |d(k,i) - d(k,j)| \quad \dots 3.4.1$$

where  $d(i,j)$  is the distance between groups i and j and a, b and c are parameters whose values for the clustering algorithms used in this thesis are given in Table 3.3.  $p_i$ 's and  $p_j$ 's are the number of samples in classes (or groups) i and j.

Table 3.3  
Parametric values for different clustering algorithms

Algorithm	Parameters			
	$a_i$	$a_j$	b	c
Single Linkage	1/2	1/2	0	-1/2
Centroid	$-p_i/(p_i+p_j)$	$p_j/(p_i+p_j)$	$-a_i \cdot a_j$	0
Group Average	$-p_i/(p_i+p_j)$	$p_j/(p_i+p_j)$	0	0

The recurrence relation 3.4.1 is not suitable for methods in which similarities rather than distance measures are employed.

A general clustering procedure is demonstrated in blocks of Fig.3.3. Using the recurrence relation 3.4.1 with an appropriate choice of a distance measure the clustering algorithms shown in Table 3.3 were implemented using the blocks of Fig. 3.3. However, depending on the choice of distance measure some of the blocks may not be used. For example, in case of Euclidean distance only the blocks 2, 3, 4, 7 and 8 will be used in order. The results of applying the algorithms of Table 3.3 on NDT design set are respectively shown in Tables 3.4 through 3.6.

Table 3.4  
Pattern Association Hierarchy using  
Single Linkage Method

node	cluster-q	cluster-r	distance
1	GHJ	ADCEBIF	376.164
2	GH	J	16.169
3	G	H	16.169
4	ADCEBI	F	16.169
5	ADCEB	I	16.169
6	AD	CEB	14.488
7	A	D	12.344
8	CE	B	10.713
9	C	E	10.562

Table 3.5  
Pattern Association Hierarchy using  
Centroid Method

node	cluster-q	cluster-r	distance
1	CEBDFGHIJ	A	11.359
2	CEBDFGHI	J	10.225
3	CEBDFGH	I	10.225
4	CEBDFG	H	10.225
5	CEBDF	G	10.225
6	CEBD	F	10.225
7	CEB	D	10.225
8	CE	B	9.394
9	C	E	10.562

Table 3.6

Pattern Association Hierarchy using  
Group Average Method

node	cluster-q	cluster-r	distance
1	FIBEAJ	GHCD	47.475
2	FIBEA	J	47.187
3	FI	BEA	46.621
4	GHC	D	45.715
5	GH	C	44.620
6	F	I	43.764
7	BE	A	43.689
8	G	H	43.575
9	B	E	40.027

To use the similarity measure as a clustering criterion, again following the block of Fig. 3.3, a stage-wise clustering algorithm shown in Fig. 3.4 was developed by modifying the similar algorithm given in [DUDA-73]. This algorithm was used by Siddiqui et al. on similar NDT-data the results of which are reported in [SIDD-89b].

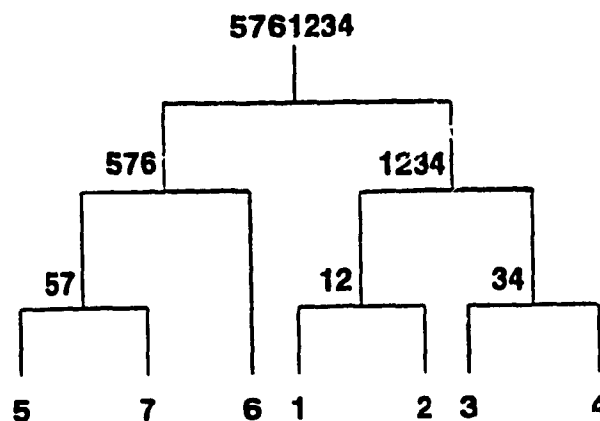


Fig. 3.2: Hierarchical Clustering Procedure.

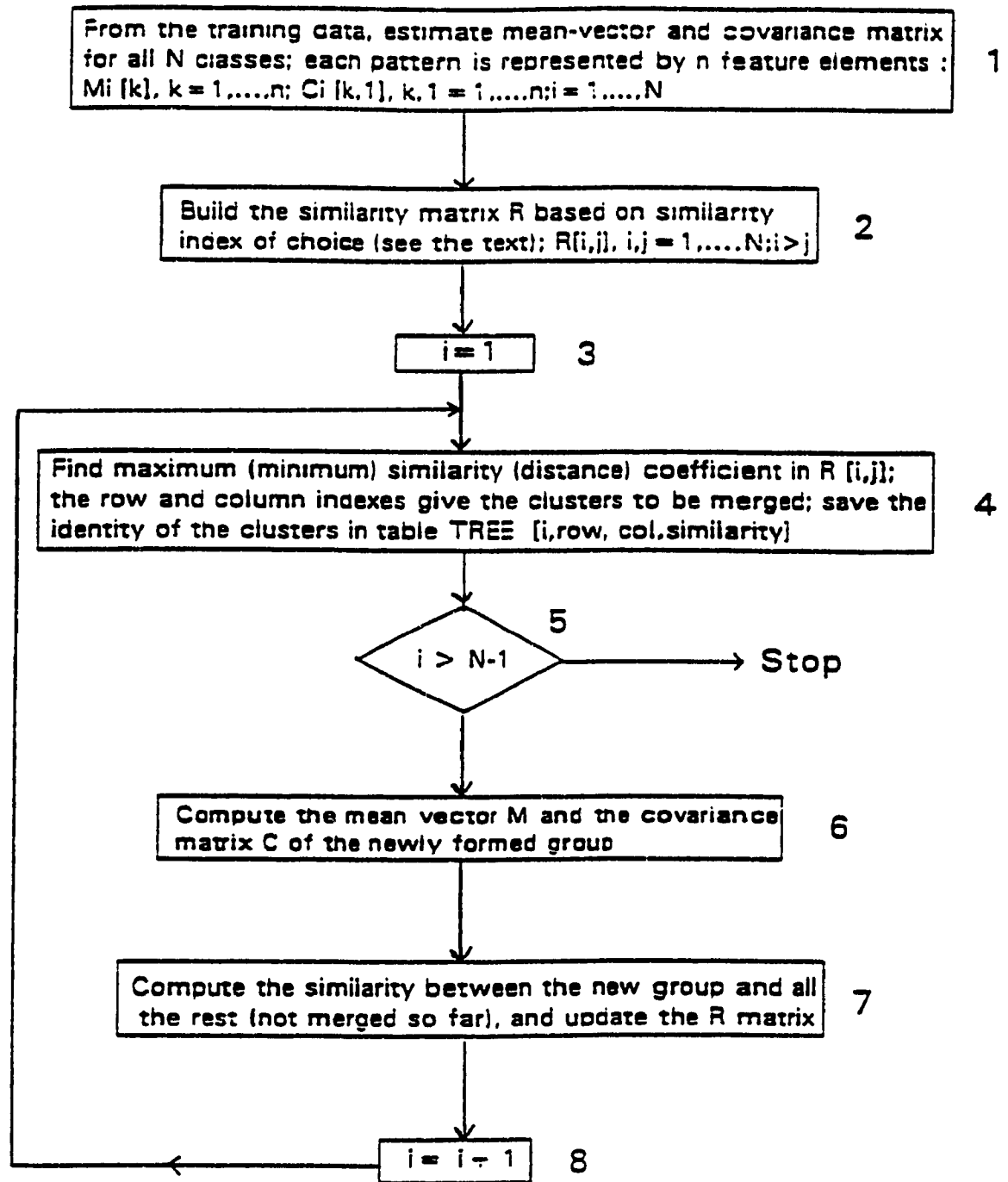


Fig. 3.3: Block diagram of Tree Organization.



/\* Consider an (N-1) X (N-1) lower triangle similarity matrix  $R = [\rho_{ij}]$ , shown in Fig.3.5, where the coefficient  $\rho_{ij}$  is related to the class correlation or to distance. If  $\rho_{ij}$  is related to the class correlation then entry  $\rho_{ij}$  is given by:

$$\rho_{ij} = \sigma_{ij} / \sqrt{\sigma_{ii} \cdot \sigma_{jj}} \quad \dots 3.4.2$$

where  $0 < \rho_{ij}^2 < 1$ , with  $\rho_{ij}^2 = 0$  for dissimilar classes and  $\rho_{ij}^2 = 1$  for completely similar classes or groups. If  $\rho_{ij}^2$  is related to inter-class distance then entry  $\rho_{ij}$  is given by:

$$\rho_{ij} = d(X_i, X_j) \quad \dots 3.4.3$$

where  $\rho_{ij}$  designates the distance (Euclidean/Mahalanobis) between mean-vectors for classes/groups i and j. \*/

- Step 1. Begin with N classes. Every training class is considered to be a cluster with its mean-vector as sample in that cluster. From the mean feature-vectors belonging to the training set, estimate the mean vectors and the covariance matrices for N classes,
- Step 2. Define between-class similarity, build up the similarity matrix R in the feature space,
- Step 3. Search the similarity matrix for the most similar pair of classes. Let the chosen clusters (group of classes) be labeled q and r and let their associated similarity be  $s_{qr}$ ,  $q > r$ .
- Step 4. Reduce the number of clusters by 1 through the merger of clusters q and r. Label the product of the merger with qr and form a new cluster. Compute the similarity between this new cluster and each of the remaining clusters and update the R matrix. Delete the rows and columns of R pertaining to clusters q and r. Copy the identity and the statistics about clusters q and r to the knowledge frame. The merger constructs a node of the knowledge tree.
- Step 5. If all the clusters have been merged into one big cluster, stop (i.e., steps 3 and 4 have been performed a total of N-1 times), otherwise go to Step 3.

Fig. 3.4. Bottom-up Algorithm for organizing the analytic features and corresponding classes in a tree structure.

	1	2	3	4	.	.	N-1
2	$\rho$						
3	$\rho$	$\rho$					
4	$\rho$	$\rho$	$\rho$				
5	$\rho$	$\rho$	$\rho$	$\rho$			
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
N	$\rho$	$\rho$	$\rho$	.	.	.	$\rho$

Fig. 3.5: Lower triangle similarity matrix ( $\rho_{ij}$  is the similarity between classes (groups)  $i$  and  $j$ , it can be defined by any one of the association measures described in Section 3.3. Assuming that the similarity is symmetric, i.e.,  $\rho_{ij} = \rho_{ji}$ , the schedule of similarities for all  $N(N-1)/2$  possible pair combinations of classes can be arranged in a lower triangle similarity matrix).

### 3.4.2 Generalized Variations Clustering Procedure

Ward [WARD-63] proposed that at any stage of an analysis the loss of information which results from the grouping of individual classes into clusters can be measured by the total sum of squared deviations of every pattern from the mean of the cluster to which it belongs. At each hierarchical step in the analysis, the union of every possible pair of classes or clusters is considered and the two clusters whose fusion results in minimum increase in error sum of squares, i.e.,

$$d = \underset{(j=1, \dots, N-1)}{\text{Min}} \left( \sum_{i=1}^p X_i^2 - p M_j^2 \right) \quad \dots 3.4.4$$

are combined. In above equation  $X$  is a pattern vector, and  $M_j$  is the mean of cluster  $j$  at the first stage of fusion.

We extended this idea to handle clustering problems involving the comparison of several pattern classes and rest this extension on the assumption of multivariate normal distribution.

We picked this idea from univariate statistical theory for testing the hypothesis of equality of population means. Assuming that samples are from a normal population, the appropriate test statistic is,

$$t = (\bar{X} - M_0) / (s / \sqrt{p}) \quad \dots 3.4.5$$

This is Student's t distribution with  $p-1$  degrees of freedom. Taking the square of  $t$  and transforming 3.4.5 into a squared distance from the mean vector,  $M_0$ , i.e.,

$$t^2 = p (\bar{X} - M_0)^T [s^2]^{-1} (\bar{X} - M_0) \quad \dots 3.4.6$$

The value  $t^2$  is the squared distance from sample mean  $\bar{X}$  to the test value  $M_0$  (or cluster mean) and the units of distance are expressed in terms of estimated standard deviation of  $X$ . Instead of using feature vectors, we decided to use inter-class variations between two classes to evaluate the association between pattern classes. This was done by computing their paired differences, thereby eliminating much of the influence of extraneous feature-to-feature variations.

For multivariate extension of the paired comparison procedure it is necessary to distinguish between  $n$  features, two classes, and  $p$  observations (samples) in each class. We labelled the  $n$  features associated with the  $j$ -th sample of  $N$  classes as:

$$\begin{array}{ll}
x_{11j} & = \text{feature 1 under class 1} \\
x_{12j} & = \text{feature 2 under class 1} \\
. & \\
. & \\
. & \\
x_{1nj} & = \text{feature n under class 1} \\
\hline
x_{21j} & = \text{feature 1 under class 2} \\
x_{22j} & = \text{feature 2 under class 2} \\
. & \\
. & \\
. & \\
x_{2nj} & = \text{feature n under class 2} \\
\hline
. & \\
. & \\
. & \\
\hline
x_{N1j} & = \text{feature 1 under class N} \\
x_{N2j} & = \text{feature 2 under class N} \\
. & \\
. & \\
. & \\
x_{Nnj} & = \text{feature n under class N} \\
\hline
\end{array}$$

and the n paired difference random variables between classes, say, q and r become,

$$\begin{array}{l}
D_{1j} = x_{q1j} - x_{r1j} \\
D_{2j} = x_{q2j} - x_{r2j} \\
. \\
. \\
D_{nj} = x_{qnj} - x_{rnj}
\end{array}$$

Let  $D_j = [D_{1j}, D_{2j}, \dots, D_{nj}]^T$  and assume for  $j=1, 2, \dots, p$ , that

$$E(D_j) = \delta = [\delta_1, \delta_2, \dots, \delta_n]^T$$

and

$$\text{Cov}(D_j) = \sigma_n$$

Assume that,  $D_1, D_2, \dots, D_p$  are independent normal with  $N_n(\delta, \sigma_n)$  random vectors, the association between pattern classes q and r about the vector of mean differences  $\delta$  can be based upon following similarity (distance) index,

$$T^2 = p (\bar{D} - \delta)^T S_d^{-1} (\bar{D} - \delta) \quad \dots 3.4.7$$

where

$$\bar{D} = 1/p \sum D_j \quad \text{and}$$

$$S_d = 1/(p-1) \sum (D_j - \bar{D}) (D_j - \bar{D})^T$$

The statistic  $T^2$  is called Hotelling's  $T^2$  distribution for multivariate sampling distribution. Thus classes  $q$  and  $r$  will be merged only if  $T^2$  is minimum for all  $q, r = 1, \dots, N$ ; and  $q \neq r$ . The results of applying this algorithm to the NDT design set are shown in Table 3.7.

Table 3.7

Pattern Association Hierarchy using  
Generalized Variations Procedure

node	cluster-q	cluster-r	distance
1	AIFBCD	GHEJ	10974.178
2	AIF	BCD	10680.808
3	GHE	J	10630.133
4	GH	E	10444.704
5	AI	F	10355.179
6	G	H	10185.185
7	BC	D	9948.950
8	A	I	9637.046
9	B	C	9523.489

### 3.4.3 Top-Down (Divisive) Organization

As mentioned earlier in Section 3.4 the PAH can be built bottom-up or top-down. The top-down method can be used in cases where the identity of the classes is unknown. The general idea of this approach is to use the splitting (divisive) method iteratively to construct the tree structure in a top-down order. This is just the inverse process of the bottom-up approach described in Section 3.4.1.

First, consider all  $N$  classes as one big group, and according to the optimization function given by equation 3.4.10, split the group to form two subgroups respectively and then their groups in a left to right sequence. The process will continue until all the terminal subgroups contain only one class and the entire binary structure is formed. The algorithm developed for this organization scheme is presented in Fig. 3.5.

The criterion function  $J$  used here is based on the scatter matrix concept [FUKU-72]. Note that any other criterion function can also be used. Let  $S_w$  and  $S_b$  be the inter- and intra-class divergence matrices, respectively:

$$S_w = 1/p_o \sum_{i=1}^N p_i S_i \quad \dots 3.4.8$$

$$S_b = 1/p_o \sum_{i=1}^N p_i (M_i - M_o) (M_i - M_o)^T$$

and the total divergence matrix  $S_t$  given by

$$S_t = S_w + S_b \quad \dots 3.4.9$$

where

$p_i$  : number of samples in class  $i$ , for  $i=1, \dots, N$

$S_i$  : covariance matrix for class  $i$

$M_i$  : mean vector of class  $i$

$M_o$  : overall mean of  $N$  classes

The scatter matrices  $S_w$  and  $S_b$  are not independent, and one can either choose to minimize  $S_w$  or maximize  $S_b$ . In either case one will affect the other, so we choose only the trace of  $S_b$  as the basis of the  $J$  value:

$$J = \text{trace } S_b = 1/p_o [p_q (M_i - M_o)^T (M_i - M_o) + p_r (M_i - M_o) (M_i - M_o)^T]$$

$$= p_q \cdot p_r / p_o [(M_i - M_o)^T (M_i - M_o)] \quad \dots 3.4.10$$

where  $M_q$  and  $M_r$  are, respectively, the mean vectors of the two groups  $G_q$  and  $G_r$ . From equation 3.4.10 it is clear that  $J$  can be obtained from any two mean vectors of the three groups  $G_o$ ,  $G_q$ , and  $G_r$ .

Several other methods have been implemented by varying the procedures used for defining the most similar pair in above two algorithms bottom-up and top-down, (at step 2) and for updating the revised similarity matrix. For the algorithm given in Fig. 3.6 a number of choices can be made to compute the  $J$  value.

#### 3.4.4 Clustering Algorithm Selection Criterion - Meta Knowledge

Clustering procedures,  $f$ 's described in previous sections are potentially very useful techniques. However, they require care in their application because of the many procedures and various decision criteria associated with them. Different algorithms may provide different groupings on the same data. To maintain the objectivity of an approach we did not merely accept the results of one algorithm or the other, instead we used meta knowledge (statistical or empirical knowledge derived from the analytical knowledge) about the data to determine the algorithm to be used.

The number of clustering algorithms available is large, as is the number of procedures in applying them. Hierarchical techniques are the most suitable since they require far less computing time; consequently they are feasible for use with large data sets particularly in the situations such as ours where the objective is to build a hierarchy of associated classes.

/\* Consider all  $N$  sample patterns (feature-vectors) belong to certain group  $G_o$ . Let  $G_q$  and  $G_r$  be the two subgroups of group  $G_o$ . Initially, let all the  $N$  samples be in  $G_q$  and let  $G_r$  be empty. Define an objective function  $J$  as a criterion of the divergence between two subgroups. \*/

Step 1.

Initialize the groups  $G_o$ ,  $G_q$  and  $G_r$  as defined in comments. Define the criterion function  $J$  (see the text). From the mean feature-vectors belonging to each group, estimate the mean vectors and the covariance matrices for groups  $G_q$  and  $G_r$ .

Step 2.

For initial division use the K-means algorithm to allocate the samples to each of the groups  $G_q$  and  $G_r$ .

Step 3.

Compute the criterion function  $J$  for all possible splitting combinations of the patterns in each group;

Step 4.

Search the criterion functions for maximum  $J$  value in each group. This value gives the way splitting should be done.

Step 5.

Increase the number of clusters by 1 for each split. Label the subgroups and form new clusters. Compute the criterion function for each new cluster. Copy the identity and the statistics about clusters  $q$  and  $r$  to the knowledge frame. Each split constructs a node of the knowledge tree.

Step 6.

If all the clusters have been split until no division is possible stop (i.e., steps 3 and 4 have been performed a total of  $N-1$  times), otherwise go to Step 3.

Fig. 3.6. Top-down Algorithm for organizing the analytic features and corresponding classes in a tree structure.

Their suitability is also dictated by number of classes in a given domain (which is also a variable) and the features to represent their classes may be quite large. The major difficulty with these techniques lies in the choice of one method from the many available and in the choice of which similarity



or distance measure to use. Of the agglomerative hierarchical techniques, single linkage is the only one that satisfies various mathematical criteria [JARD-71]. Forgey [FORG-65] also concluded that single linkage performed well with very distinct clusters of any shape, but as soon as a moderate amount of noise was added, the results quickly become erratic.

Thus to decide on a particular clustering algorithm 'f' a decision tree shown in Fig. 3.7 was developed. The scheme we developed considers the pattern classes that are optimally compact in the sense of minimum intra-class variations as homostats and the one that are not compact, as segregates - the terminologies originally used by Cattell and Coulter [CATT-66] in the similar semantics. For example, assuming that the pattern classes are known, if pattern classes fulfill the criterion for being homostat then obviously one would select the single linkage method. Following the decision tree of Fig. 3.7 a set of meta rules shown in Table 3.8 have been designed to guide the expert/user in selecting an appropriate procedure. Another set of meta rules for the selection of classification procedures are described in Chapter 6.

### **3.5 Knowledge Organization Strategy**

No matter whichever of the above clustering algorithm is used, they all lead to a hierarchical organization of the pattern classes, or simply a pattern association hierarchy (PAH). This hierarchical organization is transferred to the organization of the layered structure of the knowledge contained in features and the pattern classes.

This transfer leads us to modify our terminologies. The tree will be called a knowledge tree wherein each non-terminal node stores the knowledge pertaining to the groups or pattern classes merged at that node. In this context each non-terminal

node will be called a knowledge frame and the terminal nodes designate the ultimate decision node where an individual pattern class is identified. Thus the tree would become a tree of associated knowledge frames, realizing an efficient representation and formalization of knowledge tree designing strategy. The pattern-dependent information pertaining to each group at the individual intermediary nodes of the knowledge tree is obtained and stored in the corresponding node while performing the clustering. The concept is further elaborated in Chapter 4.

Table 3.8

Rule set for the selection of Clustering Procedure  
and Similarity Index

- 
1. Is the number of pattern classes known? [2. Yes, 3. No]
  2. Decide whether classes are homostats or segregates.
    - 2.1 Enter the pattern classes in order.
    - 2.2 Enter the number of samples in each class in the same order.
    - 2.3 /\* processing by the system \*/

Arbitrarily pick 20% (minimum 2 classes) of pattern classes and the system will read all samples belonging to the pattern classes chosen, compute the mean and variance.

intra-class variation: \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_

threshold: \_\_\_\_\_

Is the intra-class variation  $\geq$  threshold [4. Yes, 5. No]
  3. Classes are unknown; expert input is required.
    - 3.1 Select the top-down (divisive) Clustering Procedure
    - /\* The system would apply the algorithm; build the tree and ask the user to assign the actual identities to the dummy identities \*/

contd.
-

Table 3.8 (Contd.)

Rule set for the selection of Clustering Procedure  
and Similarity Index

---

4. Sample patterns in a class are segregates.  
  
4.1 /\* This step automatically determined by evaluating the inter-class variations. If the inter-class variations are small; select either the "Variations Scheme" or the "Group Average" method \*/  
  
Are the variations in the feature values significant?  
[6. Yes, 7. No]
  5. Sample patterns in a class may be homostats.  
  
5.1 /\* If the intra-class variations are  $\leq$  threshold, select "Single Linkage Method". The system would show the inter-class variations for the selected classes \*/  
Can you call the pattern a homostat ?  
[8. Yes, 9. No]
  6. Select the "Generalized Variations" clustering algorithm.  
/\* Apply the algorithm and exit \*/
  7. Select the "Group Average" clustering algorithm.  
Go to Rule 10
  8. Select the "Single Linkage" clustering algorithm.  
Go to Rule 10
  9. Select the "Centroid" clustering algorithm.  
Go to Rule 10
  10. /\* Clustering algorithm has been selected, now select the proximity index \*/  
  
Do you want the variables to be weighed?  
[11 Yes, 12. No]
  11. Select the Mahalanobis proximity index  
/\* Apply the algorithm using the selected index, build the tree and Exit \*/
  12. Select the Euclidean proximity index  
/\* Apply the algorithm using the selected index, build the tree and Exit \*/
-

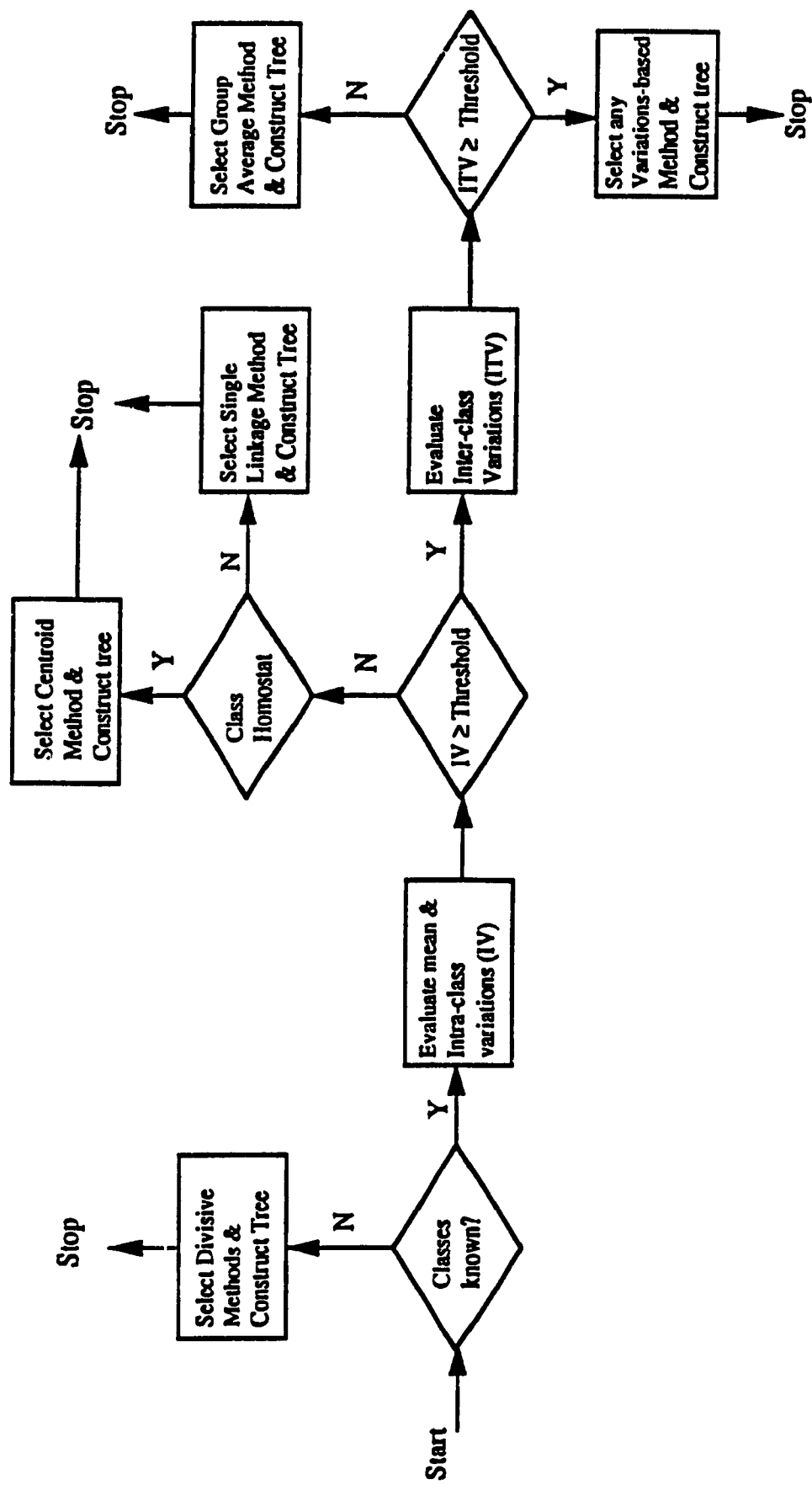


Fig. 3.7: Decision Tree for the Selection of a Clustering Procedure.

## **Chapter 4**

### **Feature Selection, Empirical Knowledge, and Organization of Knowledge Base**

#### **4.1. Introduction**

The issue of information explosion is addressed in this chapter. Several algorithms developed to eliminate redundant features are also described. The empirical knowledge which is another major component of the knowledge base is identified and structured. The empirical knowledge as considered here includes problem-domain dependent statistical parameters derived from analytical knowledge. This knowledge is used as meta knowledge and performs transition when switching from one phase of processing to another. These knowledge components are described in detail and the entire knowledge base is hierarchically organized using the PAH concept introduced in Chapter 3. The schemes for knowledge formalization, update and information retrieval are also developed. At this stage we have gathered all the knowledge that is required by the system.

#### **4.2 Selection of Optimal Features (Analytical)**

The amount of information gathered through the extraction of analytical features was prohibitively large. Two basic questions arise; what are most discriminatory features and how many of them should be used. Both of these questions are resolved using two basic postulates, 1) reduce the dimensionality of the problem, and, 2) select an optimal feature set, in the sense of giving maximum discrimination between classes. Several analytical methods are developed to resolve these issues.

#### 4.2.1 The Size Selection

The initial set of raw measurements on patterns was quite large. For classification, such a large set of measurements is prohibitive. Our goal was to find a smaller set which shall contain the most discriminatory information needed to identify the given pattern classes. The difficult question of how many features as compared to the number of samples in the design set are needed for an adequate classifier design naturally arises.

The deleterious effects of inadequate sample size have been discussed in the past. Estes [ESTE-65] showed that the error rate using the Fisher linear discriminant for feature selection deviates severely from the theoretical optimum when the ratio,  $\psi = p_0/n$ , of sample space  $p_0$  to the number of features  $n$  is small. Hughes [HUGH-68] and Abend et al. [ABEN-69] showed that the average probability of correct classification over all possible discrete class distributions deteriorates as this ratio decreases. Kanal and Chandrasekaran [KANA-68] noticed that the number of features that can be used for a fixed sample size depends upon the probability structure assumed for the problem.

In literature it has been shown that the design-set error rate is a biased estimate of the test-set error rate and the amount of this bias is dependent upon the ratio  $\psi$  [FOLE-72]. It is recognized that the greater the value of ratio  $\psi$ , the better the recognition results will be. One of the approximation suggests that when the ratio  $\psi$  is greater than 3 the design-set error rate (on average) is close to the test-set error rate which implies a closeness to the optimum error rate [KANA-68]. These observations, thus led us to assume that the size of the design-set and the information contained in it will determine the approximate number of features one should

select. We found that a smaller value of the ratio  $\psi$  will be useful only if the features selected are discriminatory. Thus for NDT problem in which 108 features were selected to represent a pattern class with 20 samples each, 4 to 15 discriminatory features depending on the choice of the classifier were considered optimal. In the following discussion  $n'$ ,  $n$ ,  $n'$  and  $n$ , all represent the number of features in a pattern vector at different stages of processing.

#### 4.2.2 Feature (Label) Selection

After deciding on the number of features, the answer to the second question was sought. That is, which of  $n$  (size) features among the larger set should be selected. The problem of feature subset selection can be considered as a problem of selecting a subset of size  $n$  features from a large set of  $n'$  features. There are  $(n' C n) = n'! / n! (n' - n)!$  such subsets to choose from. Exhaustive evaluation of all the subsets is computationally prohibitive, as the number of subsets to be considered grow very rapidly with the number of features.

A number of techniques for optimal/suboptimal selection of features have been applied in the past [AHME-85, AHME-86, DEVI-82, SIDD-90a]. For example, stepwise techniques [MUCC-71] and dynamic programming solutions [CHAY-73] are more efficient because they avoid exhaustive enumeration, but they offer no guarantee that the selected subset would yield the best subset among all subsets of size  $n$  [NARE-76]. Considering the size of the problem a number of optimal/suboptimal solutions such as Fisher's discriminant function, branch and bound and dynamic programming would also produce a prohibitive amount of partial solutions and thus exceeding the computational bounds. In order to encompass a broader range of characteristics in a feature selection problem, including number of

classes, number of features extracted, discrimination among the classes, a stage-wise feature selection scheme with a number of options at each stage is developed below.

The scheme involves two steps, Front-end and Back-end dimensionality reduction. Up front a simple feature elimination scheme called Successive Elimination Process is introduced in order to eliminate the redundant features and to reduce their size. At the second step two powerful feature optimization schemes which use a linear and a quadratic objective function, respectively, are introduced. The user/system can select one of these optimization schemes at each node of the tree to further reduce the size of the feature set without losing the recognition correctness. This stepwise optimization scheme will also give a choice to the user to select a particular optimization criterion among several available.

#### **4.2.2.1 Successive Elimination Process**

Front-end dimensionality reducer is a feature preprocessor which uses a Successive Feature Elimination Process consisting of three steps. The objective of this preprocessor is three fold, 1) eliminate those features that remain constant over entire design set of a class, 2) eliminate features whose variations do not qualify the Student's t-test at a significance level of 99%, and, 3) remove and/or merge the highly correlated features. These steps are described below.

##### **A. First Step: Removing The Stationary Features**

The first step involves the elimination of stationary features. For deleting stationary features, the following algorithm is developed:



- Step 1. Compute the mean feature-vector of the samples for each group at each node of the inference tree.
- Step 2. Compute the pair-wise within-class variance of individual features.
- Step 3. Delete the features with variance less than a small threshold  $\theta_0$ .

Using this algorithm features listed in Table 4.1 were deleted from the NDT-data.

Table 4.1

Stationary Features (NDT-data) removed

feature id's	feature id's	feature id's
5	41	43
59	76	77
84	85	86
87	88	89
95		

#### B. Second Test: Discordance Test

The second step eliminates features that fail to meet a discordance test. The discordance test performs a Student's t-test to eliminate features which had very close class means.

Let  $\bar{x}_1$  and  $\bar{x}_2$  be the estimates of class means of feature  $x$  for classes  $C_1$  and  $C_2$  and each class consisting of  $p_1$  and  $p_2$  samples respectively; and  $s^2$  a weighted average of the variance estimates  $s_1^2$  and  $s_2^2$ , then the statistic:

$$t = (\bar{x}_1 - \bar{x}_2) / s (1/p_1 + 1/p_2)^{1/2} \quad \dots 4.2.1$$

where  $s$  is given by

$$s^2 = \{ (p_1 - 1) s_1^2 + (p_2 - 1) s_2^2 \} / (p_1 + p_2 - 2)$$

follows the t distribution with  $f = p_1 + p_2 - 2$  degree of freedoms. Generalizing, the distance for a general PR problem with N classes, t can be defined as:

$$t = \sum_{i=1}^{N-1} \sum_{j=2}^N \frac{(\bar{x}_i - \bar{x}_j)^2}{s^2 (1/p_i + 1/p_j)^{1/2}} \quad \dots 4.2.2$$

$i \neq j$

The following hypothesis of closeness of means was selected to delete the features:

$H_0$ : Reject by the application of Student's t-test the features which have means within a specific threshold  $\theta_1$  and occur over distinct pairs of features with frequency threshold  $\theta_2$ .

The tabulated value of t was then determined at degrees of freedom  $f = p_1 + p_2 - 2$  at significance level of  $\alpha = 0.01$ . The hypothesis of closeness of means was then tested by comparing the absolute value of t against the tabulated value, say,  $t_{\alpha}'$ , with degrees of freedom corresponding to level of significance of 0.01. If  $|t| > t_{\alpha}'$ , and the value has occurred in  $\theta_2$  distinct pairs of features, the hypothesis of closeness of class means was rejected with 99% confidence limits. This implies that the feature  $x_i$  has some discriminatory power and may be kept. By checking all the features with t-test, those with significant class mean were selected. The features that were rejected by the algorithm for NDT-data are listed in Table 4.2.

### C. Third Test: Colinearity Test

The statistical independence of the features was checked next by determining the linear correlation coefficient between them. If these coefficients distinctly deviate from zero, for example, in absolute value greater than 0.45 for a given number of patterns, it may be concluded that the features are not statistically independent. Therefore, the third step

performs the colinearity test which involves eliminating features that have correlation with several other features higher than a preset threshold, say,  $\theta_3$ . The value of threshold  $\theta_3$  should be such that loss of useful information is minimum. In this test the feature to feature correlations are considered for the remaining features obtained from step B, above. The features deleted by applying the test are listed in Table 4.3.

Table 4.2

Features (NDT-data) rejected by t-test

feature id's	feature id's	feature id's
15	16	17
37	38	58
60	66	67
68	69	72
73	74	75
81	91	92
93	94	96
97	98	99
100	101	102
103	104	105
106	107	108

Table 4.3

Features (NDT-data) deleted by Collinearity test  
(features in parenthesis indicated original id's)

feature id's	feature id's	feature id's
9 (10)	11 (12)	14 (18)
15 (19)	17 (21)	19 (23)
20 (24)	23 (27)	24 (28)
25 (29)	26 (30)	27 (31)
30 (34)	31 (35)	41 (49)
43 (51)	44 (52)	46 (54)

If there are too many collinear features; their successive elimination may cause severe loss of information which may not be affordable; therefore, to retain the effect of both features we decided to merge them. Using a variation of PAH-correlation algorithm the features with a high correlation to another feature were merged. The algorithm is given below:

#### D. Merge Correlated Features

Step 1. Consider a  $n$ "-by- $n$ " correlation matrix,  $R = [\rho_{ij}]$ , over entire design set, where the correlation coefficient  $\rho_{ij}$  is related to the covariance by

$$\rho_{ij} = \sigma_{ij} / (\sigma_{ii} \cdot \sigma_{jj})^{1/2} \quad \text{for } i, j = 1, 2, \dots, n$$

Step 2. Find the most correlated pair of features, say,  $y_i$  and  $y_j$  and the amount of correlation, say,  $\text{cor}$ .

Step 3. Set the initial correlation threshold  $\theta_c = \text{cor} \pm 0.099$ .

Step 4. Merge all distinct pairs of features,  $y_i$  and  $y_j$  by averaging them, replace  $y_i$  by this average and delete  $y_j$ , and decrement  $n$  by number of pairs merged.

Step 5. Reset the threshold  $\theta_c = \theta_c \pm 0.1$ .

Step 6. Repeat steps 1 through 6 until the correlation between two subsequent deletions of features becomes less than the threshold  $\theta_{\min-c}$ . Go to step 7.

Step 7. Determine the number of features left, say it is  $n$  and exit.

The features that were deleted as a result of applying this algorithm on NDT-data set are shown with lower order number (id) and the ones that were replaced are shown with higher number in Table 4.4. A two tuple in parentheses represents the corresponding mapping onto the 108 original features.

Table 4.4

Features (NDT-data) merged by 2nd Collinearity test

Number	feature id's		Number	feature id's	
1.	24, 21	(46,24)	3.	29, 28	(56,55)
2.	41, 38	(82,78)	4.	27, 25	(53,48)

#### 4.2.2.2 Back-end Feature Dimensionality Reduction

Since the criterion for front-end feature selection does not warrant that a best feature set has been selected, so it cannot be considered an optimal feature selection algorithm. Therefore, it is not recommended to use the front-end algorithm alone, i.e., selecting a very small  $n'$  compared to  $n''$ . It was observed that use of successive elimination process would cause significant loss of information if the thresholds are lowered further. However, to further reduce the size from  $n'$  to  $n$  ( $n \ll n'$ ) one of the two powerful schemes suggested for this stage, is recommended. Two criteria are being investigated for this stage: 1) selecting features one by one, and, 2) selecting features simultaneously.

#### 4.3 Optimal Feature Selection - One By One Criterion

In this formulation the features are assumed to be independent and the feature selection problem is designed as a linear discriminant problem using Fisher's index [DUDA-73]. This formulation would require to find  $n$  best features, one at a time, which the classifier should choose for recognition in order to maximize the performance. In particular, the method ranks features in order of decreasing uncorrelated discriminatory power in a stepwise manner [COOL-71]. It was also assumed that the classes have multivariate normal distribution with equal covariance matrices. This covariance matrix is

estimated by pooling the covariance matrices of different classes. The resulting matrix is called the intra-class or within class estimate  $S_w$ . A second matrix  $S_T$  represents the overall covariance matrix estimated from all patterns of the classes together. The features were ranked on the basis of Fisher ratio  $F_i$ . The Fisher ratio of feature  $i$  was computed with the corresponding feature variances from matrices  $S_w$  and  $S_T$ .

$$F_i = \left[ (1 - s_{w,i} / s_{T,i}) / (s_{w,i} / s_{T,i}) \right] [(p_0 - N) / (N - 1)]$$

... 4.3.1

where  $s_{w,i}$  and  $s_{T,i}$  are the variances of feature  $i$  in matrices  $S_w$  and  $S_T$ , respectively. The largest values of  $F_i$  correspond to the most important features which best reflect the discrimination among all pattern classes. In the first step the feature  $Y_1$  with the largest  $F$  ratio is selected as the best discriminating feature. By performing covariance adjustments in matrices  $S_w$  and  $S_T$  to remove the covariance of  $Y_1$ , with the other features and an adjustment of the degrees of freedom, the  $F$  ratios are computed again to select the second feature,  $Y_2$ , with the greatest amount of uncorrelated discrimination power. In this manner, the features were ranked in an order of decreasing discrimination power. Considering the multi-class situation the features selected by the algorithm for the NDT-data are listed in Table 4.5.

The problem with this method is that one best feature is selected at a time and once a feature is selected its effect must be removed by recomputing the variance-covariance matrix from the remaining features. This characteristic makes the algorithm very expensive and time consuming. Another disadvantage is that best combination of more than two features cannot be found, because the already ranked features limit the checking of all possible combination of features.

Table 4.5

Features (NDT-data) ranked using Fisher Index

Rank	feature id's	Rank	feature id's
1	40	21	20
2	44	22	6
3	53	23	22
4	50	24	79
5	47	25	56
6	3	26	25
7	1	27	36
8	2	28	33
9	82	29	64
10	83	30	8
11	90	31	39
12	4	32	70
13	45	33	62
14	46	34	71
15	11	35	26
16	13	36	65
17	80	37	32
18	7	38	57
19	61	39	14
20	9	40	63

#### 4.4 Optimal Feature Selection - Simultaneous Features Selection Criterion

Although we have developed a comprehensive preprocessing scheme to eliminate poorly performing features, the features remain cannot be considered linearly independent since the thresholds were chosen arbitrarily. Fisher's method becomes very time consuming particularly in situations where one has to select a small feature subset from a very large one. We formulated the selection problem using mathematical programming. Using this approach we can always select the best  $n$

features subset simultaneously in the sense of minimum Mahalanobis distance without removing the influence of selected features. We called this method as Pseudo-Similarity method and it is described below.

#### 4.4.1 The Pseudo-Similarity Algorithm

In an effort to determine different ways to use variance-covariance matrices gave birth to the Pseudo-Similarity algorithm. One of the main characteristics of the algorithm is that it uses the variance-covariance matrix from the original data, i.e.,  $n'$  feature patterns and it does not require us to re-evaluate this matrix every time a feature is selected and removed from comparison. Hence it is more general than its counterpart, i.e., the Fisher's method. For the sake of comparison with Fisher's method we decided to use the sample mean as the representative of each pattern class. Thus only the means of the pattern classes were used to select an optimal feature set. The feature subset selection problem is viewed as an optimization (optimal selection) problem where the objective was to maximize the inter-class variation so that the pattern classes are as distinguishable as possible. The inter-class variation is evaluated by computing the covariance - the class dissimilarity. However, this distinction should be possible with significantly smaller feature subset. In this selection process, we selected  $n$  features, where  $n \ll n'$ . The value of  $n$  is selected according to the size (or user objectives) of problem (see Section 4.2.1) and recognition performance. These  $n$  features are those that maximize the discrimination among the pattern classes.

##### A. Feature Selection Criterion

Let  $X_i = (x_{i1}, x_{i2}, \dots, x_{in'})^T$ ,  $i = 1, 2, \dots, p_0$ , be a sample pattern of  $n'$  feature components, and that it belongs to



pattern class  $C_i$ . The overall mean,  $M$  is:

$$M = (m_1, m_2, \dots, m_n)^T$$

where

$$m_i = 1 / p_i \left( \sum_{j=1}^{p_i} x_j \right) \quad \dots 4.4.1$$

and  $p_i$  is the number of samples in class  $i$ . The overall unbiased patterns variance-covariance matrix is  $S = (s_{ij})_{n' \times n'}$ , where  $s_{ij}$  is covariance of features  $i$  and  $j$  and is defined in Chapter 3. It is clear that  $S$  is a dissimilarity measure between the pattern classes and it is symmetric. Without loss of generality we may assume it is positive definite. Each column  $S_i$  (or row  $S^i$ ) of  $S$  corresponds to one component  $m_i$  of  $M$ , or precisely, one feature in the  $n'$ -feature set.

For an  $n$ -feature selection process,  $n \leq n'$ , we select an  $n$ -features subset:

$$\mathcal{S}_i = \{ Y_{i-k} : k = 1, 2, \dots, n, i_j < i_k \text{ if } j < k \}$$

from an  $n'$ -feature set:

$$\mathcal{S} = \{ Y_i : i = 1, 2, \dots, n' \}.$$

This is equivalent to discarding  $(n' - n)$  features from the subset  $\mathcal{S} \setminus \mathcal{S}_n$ . From an  $n$  feature selection process, we obtain a sample mean of  $n'$  features:

$$M_i = (m_1^i, m_2^i, \dots, m_n^i)^T$$

where

$$m_j^i = \begin{cases} m_j & \text{if } Z_j \in \mathcal{S}_i \\ 0 & \text{otherwise} \end{cases}$$

for  $j=1, 2, \dots, n'$ , and  $m_j$  is  $j$ -th element of  $M$ . There are  $(n' C_n) = \{n'! / (n! (n' - n)!)\}$  number of  $M_i$ 's. Let  $\Omega = \{ M_i : i=1, 2, \dots, n'! / (n! (n' - n)! ) \}$ . It is clear that

for each  $M_i$ , there exists a corresponding  $\mathfrak{F}_i$ . Among all  $M_i$ 's in  $\Omega$ , a feature set  $\mathfrak{F}_{i_0}$  is said to be the best  $n$  feature subset of  $\mathfrak{F}$ , if the corresponding  $M_{i_0}$  satisfies the following:

$$M_{i_0}^T S^{-1} M_{i_0} \geq M_i^T S^{-1} M_i, \text{ for all } M_i \in \Omega.$$

It is clear that  $S^{-1}$  is positive definite, since  $S$  is assumed positive definite.

#### B. Feature Selection Model

In order to select the best  $n$  feature subset of  $\mathfrak{F}$ , we considered the following mathematical model:

$$\begin{aligned} \text{(P) Maximize } & M_i^T S^{-1} M_i \\ \text{subject to } & M_i \in \Omega \end{aligned}$$

It is clear,  $M_i = MZ$ , where  $Z$  is an  $n' \times n'$  diagonal matrix, whose components are:

$$z_j = \begin{cases} 1 & \text{if } m_j^i = m_j \\ 0 & \text{if } m_j^i = 0 \end{cases}$$

Then, by the construction of  $\Omega$ , solving the above program (P) is equivalent to solving the following quadratic program (QP):

$$\begin{aligned} \text{(QP) Maximize } & M^T Z^T S^{-1} Z M \\ \text{subject to } & \sum_{i=1}^{n'} z_i = n, \end{aligned}$$

to choose  $n$  features out of  $n'$ , and that,

$$z_i = 0 \text{ or } 1.$$

Maximizing  $M^T Z^T S^{-1} Z M$  can be expressed as,

$$= [m_1, m_2, \dots, m_{n'}] \begin{bmatrix} z_1 & & & \\ & z_2 & & \\ & & \ddots & \\ 0 & & & z_{n'} \end{bmatrix} \begin{bmatrix} s_{11} & s_{12} \dots & s_{1n'} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ s_{n'1} & \cdot & s_{n'n'} \end{bmatrix} \begin{bmatrix} z_1 & & & \\ & z_2 & & \\ & & \ddots & \\ 0 & & & z_{n'} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \cdot \\ m_{n'} \end{bmatrix}$$

$$= [m_1 z_1, m_2 z_2, \dots, m_{n'} z_{n'}] \begin{bmatrix} s_{11} & s_{12} \dots & s_{1n'} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ s_{n'1} & \cdot & s_{n'n'} \end{bmatrix} \begin{bmatrix} z_1 m_1 \\ z_2 m_2 \\ \cdot \\ z_{n'} m_{n'} \end{bmatrix}$$

$$= \begin{bmatrix} m_1 z_1 s_{11} + m_2 z_2 s_{21} + \dots + m_{n'} z_{n'} s_{n'1} \\ \cdot & \cdot & \cdot \\ m_1 z_1 s_{1i} + m_2 z_2 s_{2i} + \dots + m_{n'} z_{n'} s_{n'i} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ m_1 z_1 s_{1n'} + m_2 z_2 s_{2n'} + \dots + m_{n'} z_{n'} s_{n'n'} \end{bmatrix}^T \begin{bmatrix} z_1 m_1 \\ \cdot \\ z_i m_i \\ \cdot \\ \cdot \\ z_{n'} m_{n'} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1}^{n'} m_i z_i s_{i1} \\ \sum_{i=1}^{n'} m_i z_i s_{i2} \\ \cdot \\ \cdot \\ \sum_{i=1}^{n'} m_i z_i s_{in'} \end{bmatrix}^T \begin{bmatrix} m_1 z_1 \\ m_2 z_2 \\ \cdot \\ \cdot \\ m_{n'} z_{n'} \end{bmatrix}$$

$$= \sum_{j=1}^{n'} m_j z_j \sum_{i=1}^{n'} m_i z_i s_{ij}$$

$$= \sum_{j=1}^{n'} \sum_{i=1}^{n'} m_i z_i s_{ij} \cdot m_j z_j$$

$$= \sum_{j=1}^{n'} \sum_{i=1}^{n'} m_i m_j s_{ij} \cdot z_i z_j$$

since  $z_i = 0$  or  $1$ ,  $z_i \cdot z_j = 0$ , or  $1$

Letting  $t_{ij} = z_i z_j$ , the following integer linear program (ILP) is obtained, which is equivalent to (QP).

(ILP) maximize

$$\sum_{i=1}^{n'} \sum_{j=1}^{n'} s_{ij} m_i^2 z_i + 2 \sum_{i=1}^{n'-1} \sum_{j=i+1}^{n'} s_{ij} m_i m_j t_{ij}$$

subject to

$$\sum_{i=1}^{n'} z_i = n$$

$$z_i + z_j - t_{ij} = 1$$

$$z_i, t_{ij} = 0 \text{ or } 1 \text{ for all } i, j.$$

Using the optimization model the optimal solution  $z^* = (z^*_1, z^*_2, \dots, z^*_{n'})$  was obtained. If  $z^*_i = 1$  the  $i$ -th feature is selected, otherwise it is discarded. The optimal set of features selected are shown in Table 4.6.

Table 4.6

Features (NDT-data) ranked using  
Pseudo-similarity Algorithm

Rank	feature id's	Rank	feature id's
1	40	21	47
2	47	22	45
3	7	23	33
4	90	24	82
5	2	25	57
6	64	26	79
7	80	27	65
8	46	28	56
9	63	29	39
10	82	30	71
11	9	31	36
12	32	32	83
13	22	33	6
14	70	34	53
15	20	35	4
16	26	36	11
17	62	37	13
18	61	38	8
19	1	39	3
20	14	40	44

#### 4.5 Weight Allocation to Features

It was desirable to achieve high recognition performance without necessarily using the large feature set. One method for increasing the performance of decision criterion (proximity index, or discriminant value) involves additional analysis of features in the design set. Those features which were found to be more reliable than others were given more importance when making classification. The idea behind this was to try to make the intra-class distance as small as possible while maximizing the inter-class distance simul-

taneously. Thus several weighing factors were determined which cause the more reliable features to make larger contribution to the decision measure between two pattern classes. Several parametric and non-parametric approaches to estimate feature weights are discussed in [CASH-87, ULLM-73].

Intuitively, the feature that has smaller intra-class variance is more reliable and should contribute more to the decision process. Following Sebestyne [SEBE-62], therefore, a reasonable approach would be to assume:

$$w_i = 1 / \overline{\sigma_i} \quad \dots 4.5.1$$

The denominator of the above equation,  $\overline{\sigma_i}$  is the average standard deviation of feature  $i$  over all  $N$  classes and computed as follows:

$$\overline{\sigma_i} = 1/N \left[ \sum_{j=1}^N s_{ij} \right] \quad \dots 4.5.2$$

Hsia [HSIA-81] has proposed a weighing factor and is given by,

$$w_i = s_i / | m_i | \quad \dots 4.5.3$$

where  $s_i$  is the standard deviation of the  $i$ -th feature, and  $| m_i |$  is the magnitude of the mean of the  $i$ -th feature over the entire design set. Tou and Gonzalez [TOU-74] derived the following weighing factor:

$$w_i^i = \left\{ 1 / \left[ s_i \sum_{j=1}^N 1/s_{ij}^2 \right] \right\}^2 \quad \dots 4.5.4$$

This approach differs from the two above in that a weighing factor is computed for each feature of every class, rather than just one weighing factor for each feature that is used for all of the classes. Another factor that could be useful if the intra-class variation is small would be:

$$w_i = s_i \quad \dots 4.5.5$$

This is the same as the factor suggested by Hsia except that the denominator has been set equal to 1. If there is a large

intra-class variation, then the following weight might prove useful,

$$w_i = s_i / \overline{\sigma_i} \quad \dots 4.5.6$$

where the denominator is defined in equation 4.5.1. If there are large inter-class variations, perhaps it is not necessary to allocate variable weights, and as such  $w_i = 1$ , for  $i=1, \dots, n$  may serve the purpose.

Fisher weights for each feature were also computed. To separate class  $j$  from  $k$  by means of feature  $i$ , the Fisher weight  $w_i$  is given by:

$$w_{i,j,k} = \{ m_{i,j} - m_{i,k} \}^2 / \{ p_j s_{i,j}^2 + p_k s_{i,k}^2 \} \quad \dots 4.5.7$$

The Fisher weighing factor  $w_i$  of feature  $i$  for all  $N$  linear class separation is:

$$w_i = 2 \left\{ \sum_{j=1}^{N-1} \sum_{k=j+1}^N w_{i,j,k} / N(N-1) \right\} \quad \dots 4.5.8$$

These weighing factors were incorporated and tested on different data sets.

#### 4.6 Empirical Knowledge

Although, empirically a large number of parameters can be used, human experts tend to rely more on visual, graphic, and summary characteristics of the data. To recognize a signal in NDT problems, for example, most popular features that have been considered by a fairly large group of experts include rise time, peak amplitude, duration, peak counts above certain threshold and to identify the location of a defect, the time at which the signal/peak has occurred. These features have already been represented by analytical knowledge components in earlier chapters. In addition to the analytical knowledge

parameters listed in Table 3.2 (Chapter 3), more specific data dependent information is also available. The data dependent knowledge comprised of a set of empirical observations which were derived using analytical features from the design set of data by subjecting it to a number of analytical and statistical procedures. These procedures evaluate the parameters that we considered useful for classification and include:

- individual class probabilities (a priori and conditional) obtained from the design set, see Chapter 5,
- threshold settings based on the characteristics of the design set,
- statistical distribution parameters for the design set, see Table 4.7,

Table 4.7

Empirical and Statistical Decision Parameters

---

Empirical Parameters

inter-group (cluster/class) distance  
intra-group (cluster/class) distance  
number of features and their labels  
feature selection algorithm  
feature weighing criterion  
classification procedure selection algorithm  
PAH building criterion

Statistical Parameters

mean (Class & Overall)  
dispersions (Class & Overall)  
Covariance (Class & Overall)  
Correlation coefficients  
first four central moments  
Inverse (Class & Overall)  
shape characteristics - skewness and kurtosis

---



The empirical and statistical parameters described above were estimated and the acquired knowledge corresponding to each group of clusters was allocated to the respective nodes of PAH. These parameters were then stored in a knowledge base with conceptually the same structure as PAH's. While processing, each intermediary node of the PAH can be selected and modified to provide the user/expert various options to modify the following decision parameters.

- threshold adjustments based on the nature of the problem domain, objectives and the design data,
- option to use a variety of algorithms to build the initial Pattern Association Hierarchy,
- option to use a variety of proximity indices to establish the pattern association,
- option to use a variety of discrimination algorithms to classify an unknown pattern,
- option to use a variety of algorithms to select an optimal feature set, or to use subjective features,
- option to retrain a nodal classifier,
- option to use a variety of feature weighing criteria.

Thus empirical knowledge was used to drive a set of meta rules which in turn are used to select, 1) a suitable algorithm for building pattern association hierarchy (PAH), 2) suitable algorithm(s) for feature selection, 3) appropriate algorithm(s) for pattern classification, 4) parameters for decision making in regards to both (1) and (2) above, 5) appropriate weights for each feature, and, 6) the rules required for various stages of processing including the rules for the final stage of classification. Another function of this knowledge is to identify an anomalous event.

#### **4.7 Knowledge Formalization, Representation and Organization**

Several techniques for knowledge organization have been developed for AI systems [RICH-91, CHAR-85]. These techniques can roughly be divided into two types: declarative methods (such as predicate logic) and procedural methods. The declaratives are of two types, static and dynamic. Static declaratives in which most of the knowledge is represented as a static collection of facts, are the knowledge objects constructed either at the design-time using a priori knowledge, or at the learning-time, using empirical knowledge. The dynamic declaratives are the objects which are created by the system during the learning time and may be modified as the learning progresses. The declaratives are further accompanied by a small set of general procedures for manipulating them. In procedural methods, the bulk of the knowledge is represented as a set of procedures for using it. We conceptually used similar methods for knowledge representation and organization which are described below.

##### **4.7.1 Knowledge Formalization and Representation**

We used pseudo-dynamic objects to formalize and represent knowledge which is basically a hybrid of the two concepts described above. The pseudo-dynamic objects constitute a knowledge frame storing the requisite knowledge corresponding to each internal node of the PAH accompanied by a set of procedures for manipulating them [STEF-81, WEIS-84, IGRO-90]. A frame, as shown in Fig. 4.1, is a collection of information associated with a group of classes at a specific node. The information consists of a set of group-dependent knowledge objects and appropriate indices to further lower level components in the knowledge base. The information regarding the classification procedures to be used by the discrimination

system is also stored in the frame. The analytical features are organized in arrays of vectors and the reference samples pertaining to each class were stored separately. This formalization has greatly enhanced the flexibility compared to the traditional approaches.

#### **4.7.2 Hierarchical Knowledge Organization**

The Knowledge Base holds analytical features, and empirical and statistical knowledge about the problem domain at hand. Not all the knowledge components are utilized simultaneously. To structure the knowledge according to its order of utilization, the knowledge base was partitioned into groups corresponding to the way the groups (classes) are merged during building of PAH-tree (see Chapter 3). The organization of knowledge uses the PAH skeleton as its design strategy. The structure of the tree as shown in Fig. 3.2 is composed of layers of nodes linked with branches. Nodes are of two types, terminal and non-terminal nodes. At each nonterminal node a knowledge frame is placed wherein the knowledge acquired during the system design phase is stored. A frame can be accessed by the discrimination system when needed. Each terminal node represents a class (an ultimate decisive situation). Non-terminal nodes represent the intermediary decisions. A frame, as described in previous section is a collection of information associated with a group of classes at an internal node of the PAH. Thus with this arrangement of knowledge structure the knowledge base becomes a binary tree of frames containing knowledge pertaining to associated classes distinguishable by their characteristics stored in the knowledge frame at each non-terminal node.

Major advantages of this arrangement include minimum information ordering and redundancy problems with greatly increased flexibility as compared to their traditional counterparts,

which consequently brings in efficiency in storing and retrieving the pertinent information to and from the knowledge base.

#### Group Information

- within group: mean, covariance
- No. of features, no. of samples

#### Cluster p/q information

- identity of membership
- within group: mean, covariance
- between group: mean, covariance
- distance (weighed/unweighed) -- Euclidean, Mahalanobis

#### Feature-related parameters

- full set
- Successive Elimination process
- ranked feature set
  - + Fisher ranked
  - + Pseudo-Similarity ranked
- feature weights

#### Classifier-related parameters

- No. of samples in the design set
- decision thresholds
- feature optimization procedures
- analytical knowledge (optimal feature set)
- discriminant functions

#### Nodal Classifiers (Procedures)

- Parametric
  - + Linear Discriminant function
  - + Quadratic Discriminant function
  - + Bayesian
    - \* approximation using distance for
    - \* approximation using covariance (within/between groups)
    - \* heuristic approximation
- Non-parametric
  - + K-nearest neighbor;  $k=1, \dots, 5$
  - + Minimum distance

#### User-defined Parameters

- selection of the design set
- no. of samples in the design set
- no. of samples in the testing set
- features and their weights
- procedures to be used for weighing
- learning procedure
- decision thresholds
- procedures to be used for feature optimization
- classification algorithms to be used
- decision criterion for the selected classifier

Fig. 4.1: A typical organization of knowledge frame.

## Chapter 5

### Inference Engine and Machine Learning

#### 5.1 Introduction

The inference engine primarily signifies the problem-solving mechanism which may be adopted by the system. Traditional inference engines are inductive and in these systems the decision is evolved through an unwieldy interaction with the user or expert. We adopted an evolutionary approach that does not entirely rely on the bidirectional human-machine interaction. The strategy we developed uses the PAH tree as the sequence of events that the system has to follow and as such the terms PAH and inference tree will be used synonymously. The inference tree uses both the decision theoretic and the information theoretic type of classification algorithms. The system can operate in two modes, executive mode and consultant mode. In executive mode the system is intended to function as a stand-alone system whereas the consultant mode is designed to let the user/expert use the parameters and algorithms of his/her choice. In either modes, for classification the system accepts a signal classification problem that falls in one of the system's domains of expertise, preprocesses, analyzes, performs classification process and finally comes out with a solution. In the consultant mode, however, the system performs like an assistant to the user and provides a large number of choices which a user can choose at each stage of processing, ranging from the building of the PAH to the selection of a classification algorithm at each non-terminal node of the PAH.

This problem-solving strategy has been programmed into the inference engine through two cooperating processes, Discrimination subsystem and Cognition subsystem. This incorporation

of evolutionary approach makes the inference engine more realistically a problem-solving machine which tactically analyzes the problem, sorts out the knowledge to be used and interprets the knowledge accessed from the knowledge base; subsequent knowledge gained during the operation, supplements the decision-making. The inference engine developed for the intelligent recognition system is described in this chapter. Several algorithms for training the inference engine are also developed. A new classification algorithm based on information theoretic approach is also described in this chapter.

## **5.2 Components of the Inference Engine**

The design of the inference engine developed here is shown in Fig. 5.1. The two-fold structure of the inference engine - the discrimination subsystem and cognition subsystem allows to meet the following classes of requirements:

- dual (executive and consultant) mode of system operations;
- knowledge accumulation (acquiring, extending, modifying and automatically maintaining the consistency of the knowledge base);
- hierarchical parametric inference, providing most suitable classification scheme for each subset of the classes;
- natural grouping of pattern classes using supervised/un-supervised hierarchical clustering schemes;
- deductive inferencing, based on pattern classification procedures;
- tree-based inference structure providing backtracking facility;
- parametric and non-parametric learning of the inference tree by using appropriate nodal classifier, features and pattern-class dependent decision parameters;
- plan refinement during problem decomposition through selection of optimal features or user-defined features and classification algorithm;

- classification or learning path modification, according to the user requirements.

The above requirements are accomplished by an inference engine consisting of three main components, namely, discrimination subsystem, cognition subsystem and the failure control subsystem (see Fig. 5.1).

The discrimination system, using a suitable pattern classification algorithm (see Chapter 6), performs the primary classification of the signals under a very strict range of decision parameters determined from the design set. The classification process may in fact terminate here, if the system is in the executive mode and the input pattern closely agrees with the characteristics of one of the reference patterns.

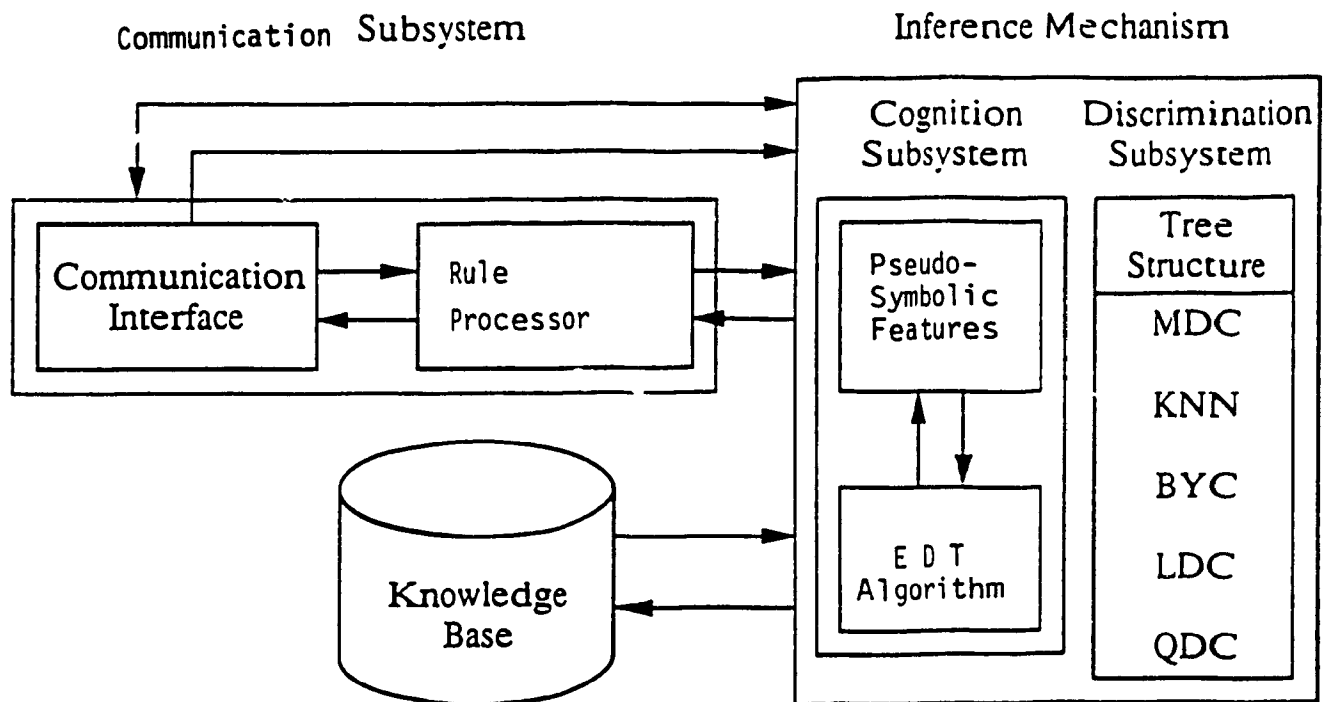


Fig. 5.1: Schematic Design of the Inference Engine.



The cognition system is another classifier. It uses an information theoretic algorithm that we called Entropy based Decision Tree (EDT) classifier. The success of the cognition system can be measured by its recognition performance. Chapter 8 presents several suggestions to improve its performance even further.

The discrimination system may fail at any intermediate node. When this happens the Failure Control scheme will attempt to classify the unknown pattern one more time using the single layer classifiers. If the Failure Control system also fails an unresolvable situation is detected which may yield the consequence of, 1) an anomalous class might have been found, or, 2) the pattern in question may belong to a gray area that may still fall between two or more of the existing classes. In the latter case the user may change a number of parameters in an attempt to reclassify the event, however, the former situation warrants the presence of an unknown event which may require the retraining of the system. This situation is not dealt with in this research.

### **5.3 Inference Mechanism**

As described earlier the incorporation of several classification algorithms has truly transformed the PAH into an intelligent recognition system wherein the signal classification can be performed by one of the two types of independent classifiers and managed by a control subsystem. These classification schemes include decision theoretic and information theoretic algorithms.

The controlling task is the coordination of the other two components (see Fig. 5.2). The primary function of this subsystem during the learning phase is to examine the data (design set) characteristics and based on them apply the meta

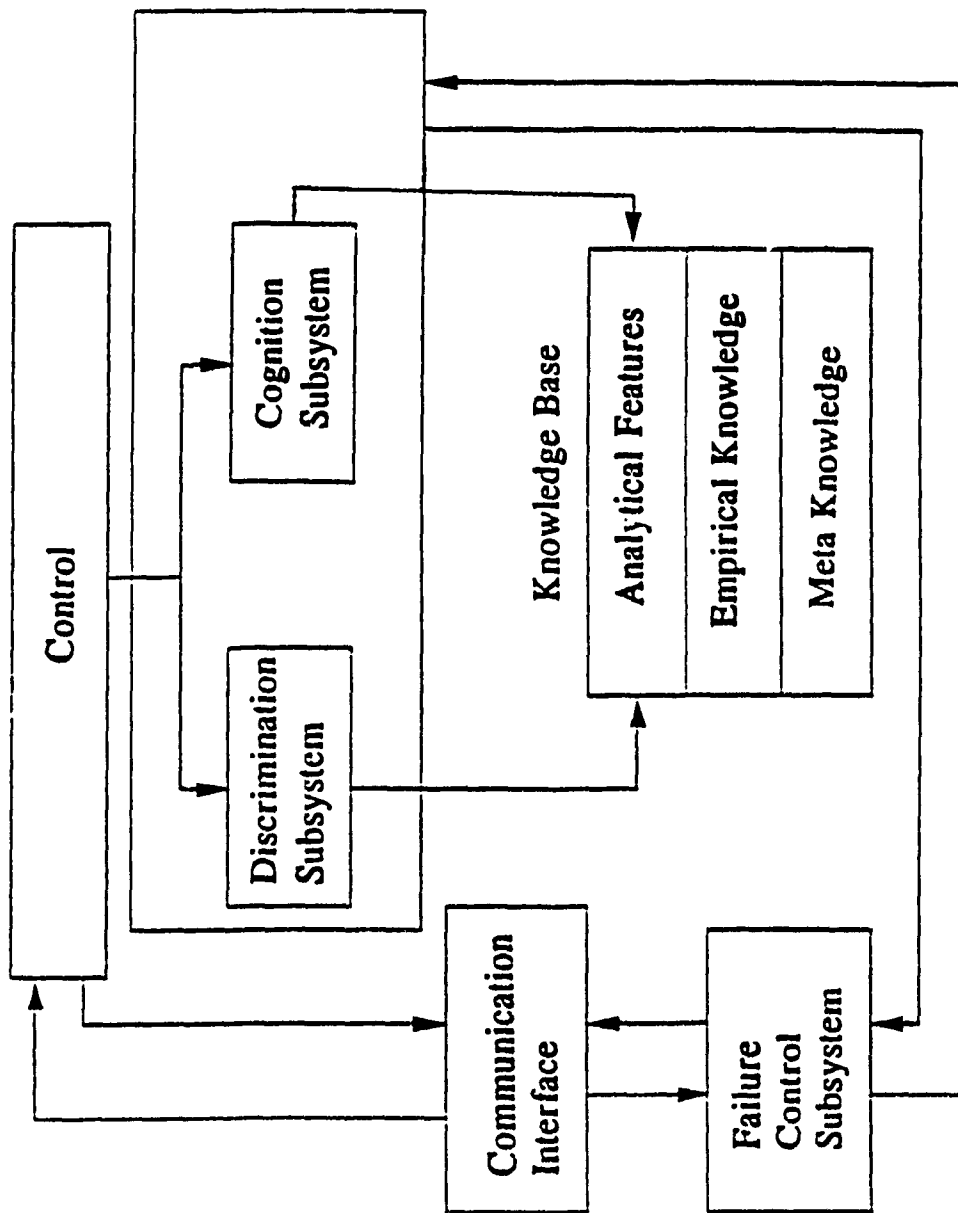


Fig. 5.2: Inference Mechanism and Control Strategy.

knowledge pertaining to the selection of a classifier and to choose an appropriate classification scheme. For the classification of unknowns it acts as an interface between the classification algorithms and the main inference mechanism and directs the classification process to select the algorithm determined during the training.

To classify and interpret, it basically lets a pattern traverse through the PAH, invoking the appropriate components of the inference mechanism at each node and performing the classification in order, until a decision regarding its identity is made.

The primary components of inference engine, namely, the cognition system, and the discrimination system are basically the knowledge processing systems. The choice of a particular type of classifier depends on the objectives of the user when one is using the system as a consultant. However, when the system is working as a stand-alone system it will use discrimination subsystem only. The discrimination subsystem is a procedure based system and uses several decision-theoretic algorithms. These algorithms include two basic types of classifiers, parametric and nonparametric. Among parametric classifiers, linear discriminant classifier (LDC), quadratic discriminant classifier (QDC), and Bayesian classifier (BYC) are included, whereas minimum distance classifier (MDC) and K-nearest neighbor (KNN) were selected as non-parametric classifiers. Based on the nature of the design data set or the user/expert choices the system selects an appropriate classification algorithm at every non-terminal node while traversing (inferencing) through the knowledge tree. The discrimination algorithms are described in Chapter 6.

The cognition system is a decision tree based processing system and uses an information theoretic algorithm for classi-

fication (see Section 5.6). The failure control mechanism is a back up system and will only be invoked upon the failure of the discrimination subsystem. It also uses the same decision theoretic algorithms of the discrimination system.

In order to let the inference mechanism perform its task of problem solving and decision making, the machine needs to be trained on the kind of problems it will be solving. Since the inference engine is a composite of two types of classification algorithms, appropriate learning schemes for each type of mechanism were sought accordingly.

#### **5.4 Machine Learning**

The ability to learn must be the part of any system that would exhibit general intelligence. Feigenbaum [FEIG-83] has called the 'knowledge engineering bottleneck' the major obstacle to the widespread use of knowledge based systems. This bottleneck refers to the cost and difficulty of building such systems through the efforts of knowledge engineers and domain experts.

There is as yet no unifying theory for machine learning [LUGE-89]. However, Carbonell [CARB-86] suggested a few guidelines to approach the learning problems. This list of themes categorizes learning in terms of, a) the specific type of training data, and, b) the data structures and operators of the learning program.

Two schemes were developed for the learning program emphasizing particularly on the second issue of data structures and operators for training. The design data set discussed in Chapter 3 was used to train the system. The discrimination system was trained using a supervised learning scheme.

The cognition system is an information theoretic classification system and is trained using the pseudo-symbolic analytical knowledge (see Section 5.6).

The learning from the design set involves performing the induction of general principles from a set of reference patterns. Such learning may be either incremental, modifying its concepts in response to each training instance, or in batch form, forming concepts in response to the entire design data. The Cognition system learns the process by selecting the feature that contains the most discriminatory information to categorize the reference patterns.

#### **5.4.1 Learning by the Discrimination System**

The discrimination subsystem attains its decision making capability through a supervised learning scheme. The system uses several pattern classification algorithms which utilize the pattern association hierarchy (PAH) concept. This concept primarily constructs a tree of classes by assigning the classes or group of classes with greater similarity to the same node.

The learning process involves two steps, first it selects an appropriate clustering algorithm to construct the tree through automatic assignment of known pattern classes from the design set to an appropriate node. Second, it provides a mechanism (see Chapter 4) to automatically select the optimal feature set and evaluate the empirical knowledge parameters necessary for selecting the most suitable classification algorithm that maximizes the discrimination between the classes or groups associated with each node.

Generically, the discrimination is done through a decision plane of thickness  $2\tau$ , where  $\tau$  is a method dependent threshold

and is determined empirically from the design set of respective data sets. The learning process determines a decision plane with no patterns within a dead zone of  $\pm 7$  from the plane.

This arrangement is intended to improve the correct assignment of an unknown pattern into one of the designated classes while conducting the classification. The actual learning process is described below.

### Supervised Learning Process

The learning from any design set available can be done either incrementally using one instance (pattern sample) at a time or as a single batch process using all the samples simultaneously. The incremental learning, although computationally more complex, did not perform any better than the single batch process and hence will not be discussed any further.

In batch learning, the feature vectors (analytical knowledge) from the design set, having known the identity of the pattern classes, were iteratively used to build the initial inference-tree in a bottom-up fashion. The tree building procedures are already explained in Chapter 3. The tree in the form of class hierarchy represents the range of signal classes that can be inferred. The learning process yields the known identity of the class(es) or group of classes at every node of the tree which are recorded in the corresponding knowledge frames. Using the knowledge already stored in the frames a set of optimal features for the individual groups at every node were selected. A set of statistical parameters described in Chapter 4 were also evaluated and saved in the respective knowledge frames.

The procedure for supervised learning of the discrimination

system using the design set is shown in Fig. 5.3, whereas the training process for classification adopted at individual non-terminal nodes is given below.

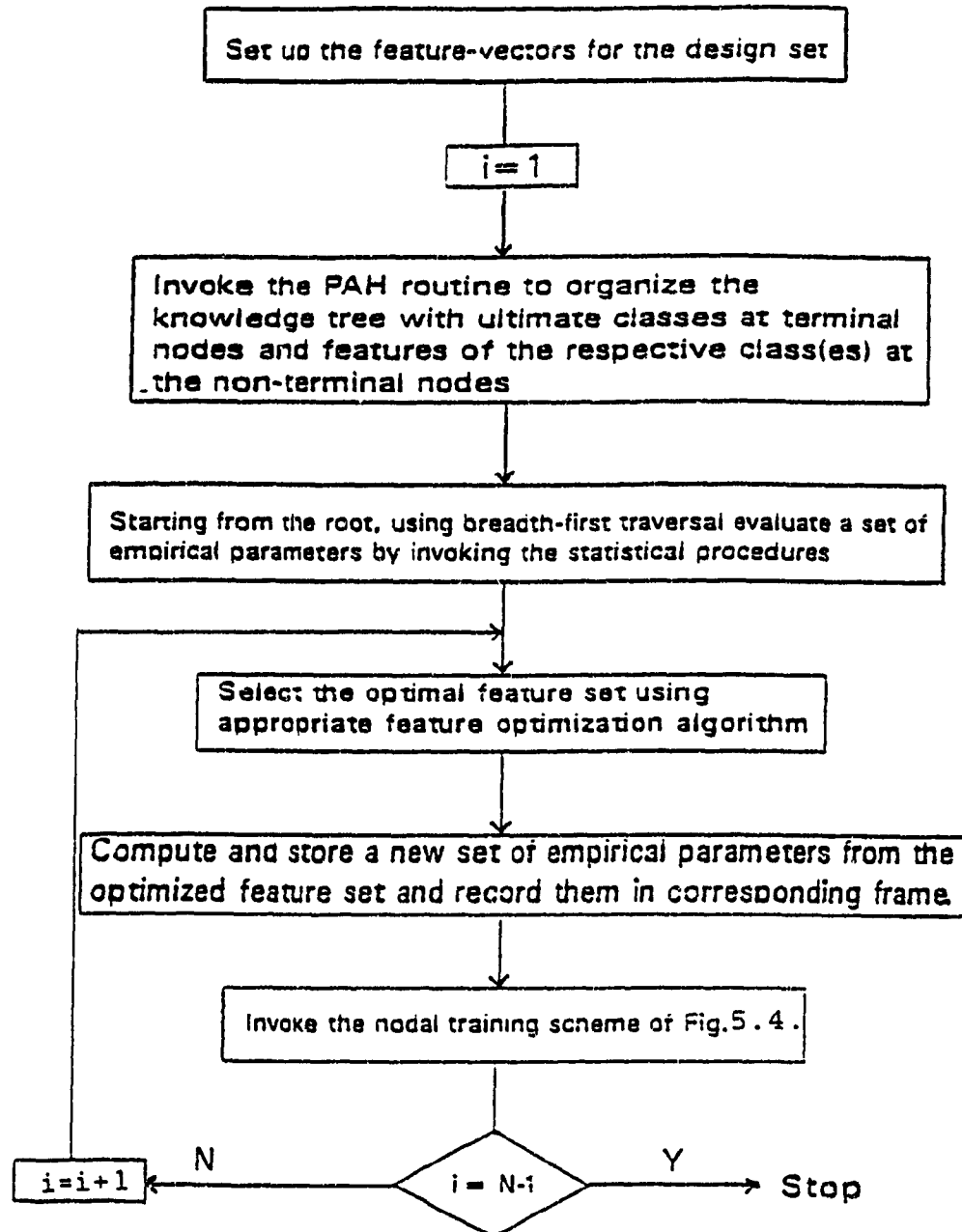


Fig. 5.3: Supervised Learning procedure for the Inference Tree.

In fact, the learning of the discrimination process was performed simultaneously while the tree building process was administered.

Once the system is trained for a specific kind of problem, an inference tree (intelligent recognition system) which will be capable of using several classification algorithms with same/different set of features at every node, is obtained. The discrimination algorithms and the decision parameters for the selection of an appropriate classifier are described in Chapter 6.

### Nodal Training

The process of building PAH has produced two classes at each node. If the two classes form well separated clusters in the pattern space, it is possible to find a decision plane which would maximally separate the classes. The objective of the nodal training was to construct such decision planes. At each node, the training starts with an appropriate discrimination plane passing orthogonally through the origin of the cluster (see Fig. 5.4). The decision plane is defined by a decision vector which is a function of the discrimination function, and the reference group decides whether a pattern lies on the left or right of the decision plane implying belonging to cluster (or a class) on the left or right of the current node. The scalar product  $r$  for linear discriminant function (see Chapter 6), for example, may be given by equation 5.4.1. It is computed from weight vector  $W$  and pattern vector  $X$ . Assume that it is positive for the cluster on the left and negative for the cluster on the right.

$$r = \sum_{i=1}^n w_i x_i \quad \dots 5.4.1$$

where  $x_i$ ,  $i=1, \dots, n$ , is a feature of a pattern of the



cluster at the current node, and  $w_i$ 's are components of the vector  $W$ . If a pattern is misclassified during the training phase the weight vector is corrected as follows:

$$W' = W + \zeta \cdot X \quad \dots 5.4.2$$

After this correction the distance between  $X$  and the decision plane lies on the correct side of the plane. The same consideration is valid for the scalar product  $r$  before correction and  $r'$  after correction, i.e.,

$$r' = -r \quad \dots 5.4.3$$

From equation 5.4.1 we can write,

$$W' \cdot X = -W \cdot X \quad \dots 5.4.4$$

From 5.4.2 and 5.4.4 an equation for the correction factor can be obtained, i.e.,

$$(W + \zeta \cdot X) \cdot X = -W \cdot X \quad \dots 5.4.5$$

Substituting  $r' = W' \cdot X$ , and using equation 5.4.2 the value for the correction factor is obtained as,

$$\zeta = -2 \cdot r / (X \cdot X) \quad \dots 5.4.6$$

Based on the nature of the data, several methods for computing the weight vector were developed. These methods include linear weighing and variance weighing of a feature and have already been described in Section 4.5.

#### 5.4.2 Learning by the Cognition System

The cognition system is designed to perform independently using the transformed analytical features which will be called

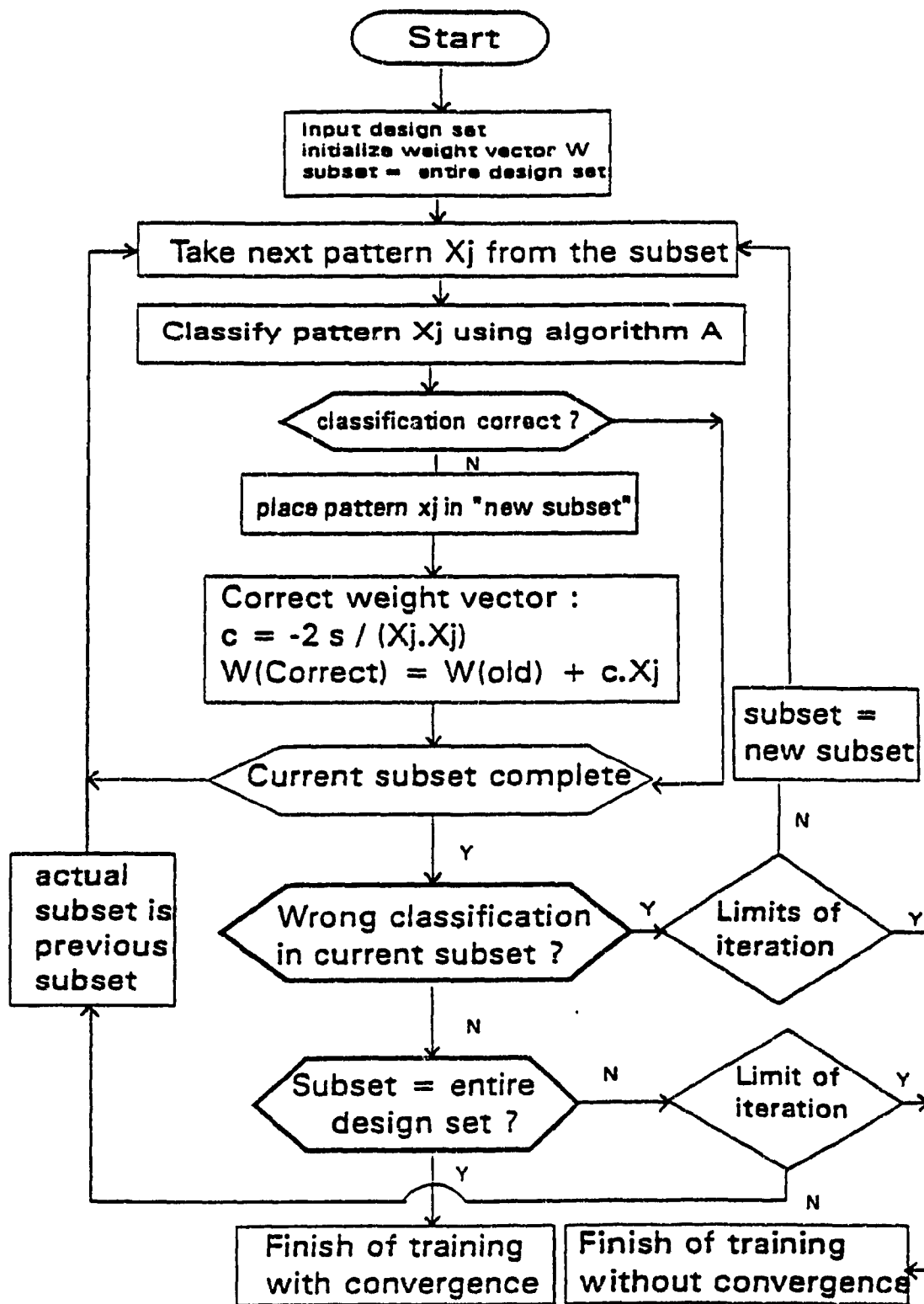


Fig. 5.4. Nodal Training Scheme - Discrimination System.

pseudo-symbolic features. The cognition system dealing with the pseudo-symbolic knowledge uses an information theoretic algorithm called entropy-based decision tree (EDT) algorithm to select an appropriate set of features to discriminate the pattern classes. The actions corresponding to the selected features and the classification scheme are performed in order, to arrive at the final conclusion, i.e., determination of class labels of the unknown.

The EDT algorithm first ranks all the pseudo-symbolic features and then builds the tree by successively selecting one feature at a time and determining the classes that can be identified using the selected feature. The algorithm then recurs on each branch with the remaining features. When all the branches lead to a single classification the algorithm halts. This learning process is used to train the cognition system. The learning cycle includes the following steps:

- Determination of the relationship between the pattern classes and the pseudo-symbolic features.
- Exercising of detective actions using the EDT algorithm and building a decision tree.

The EDT algorithm uses analytical features from the design set as knowledge objects. To work with the information-theoretic concept the analytical features were transformed to pseudo-symbolic features. Based on the information the pseudo-symbolic features carry all sample patterns in the design set were hierarchically dichotomized until all pattern classes are individually identified.

### **5.5 Discrimination System - The Process**

The discrimination subsystem (see Fig. 5.2) performs the primary signal classification. In either of the two operating

modes, executive, and subordinate, the system (See Fig. 5.1) uses the same tree which has been organized using the known identity of the waveform signals during the knowledge organization phase. In effect, at a node, if the q-th group of class(es) and the r-th group of class(es) have the highest similarity and had been merged to form an i-th group then while performing discrimination, these two groups (or classes) should be distinguished at this node. The unknown sample to be classified iterates from the root to the leaves in hypotheses and test fashion. It is first assigned to one of the groups, each of which contains several classes, of the nodes in the first level. This procedure continues at each hierarchical layer of the tree and the classification becomes finer and finer while the sample goes higher and higher up the tree, until finally it arrives at one of the terminal nodes containing only one designated class. This algorithm is described in Fig. 5.5. The discrimination algorithm and the parameters used for their solution are described, in detail, in Chapter 6.

## **5.6 Cognition System - The Process**

As described in Section 5.4.2 the cognition system is designed to perform independently using the pseudo-symbolic features and EDT classification algorithm. It builds the tree by first ranking all the features in terms of their effectiveness, in partitioning the design set into two target groups (or classes) from an information-theoretic standpoint. It selects the feature with the highest rank and then makes this feature as the root of the tree; each branch represents a partition of the set of classes. The algorithm then recurs on each branch with the remaining features. When all the branches lead to a single classification with specified thresholds, the algorithm halts. Complete details of the EDT algorithm are provided in Section 5.7.

### 5.7 Entropy-based Discrimination Tree (EDT) Algorithm

A cognition algorithm called Entropy-based Discrimination Tree (EDT) algorithm which is based on information theoretic approach is introduced in this section. This approach hierarchically selects one feature at a time based on its information content. Then using a decision function, it determines the samples which can be placed in either of the two groups.

- Step 1.  
Initialization;  $i = 1$ , to represent the root of the tree;
- Step 2.  
Scan the knowledge frame at the  $i$ -th node of the knowledge-tree and determine the classification algorithm, a best feature set (Fisher ranked or Pseudo-similarity ranked), and other parameters required by the algorithm.
- Step 3.  
Invoke the nodal classifier and assign the input pattern to one of the groups (child) at the current node. If the classifier fails, go to Step 6.
- Step 4.  
Assign  $i$  the rank of the cluster in the knowledge-tree, and repeat steps 2 through 4 until the unknown input class is assigned the identity of a leaf node.
- Step 5.  
If  $i$  corresponds to the highest terminal node the pattern has been classified; so exit; otherwise go to Step 6.
- Step 6.  
The system failed to classify the unknown pattern; invoke the Failure Control system and exit.

Fig. 5.5. Tree classification mechanism of the Discrimination System.

The ID3 (Iterative Dichotomization) method [QUIN-83] is a popular example of such an approach to learn a discrimination

tree which is closely related to the Concept Learning System of Hunt [HUNT-62]. A major difficulty with this approach is that it is particularly useful when there are a small number of patterns, and each of which is made up of a short list of qualitative symbolic feature values. An additional problem is that the features are considered mutually exclusive and that they may be binary valued. These limitations restrict the scope of the algorithm. Borrowing the basic decision making concepts we transformed this algorithm to acquire the knowledge from the design set wherein the sample patterns may be modeled with numeric features. In addition to this premise we did not consider the features to be mutually exclusive. Instead we considered a feature with a specific range of values to be mutually exclusive. Incorporating these improvements a modified ID3 algorithm called EDT (Entropy Based Decision Tree) algorithm is developed.

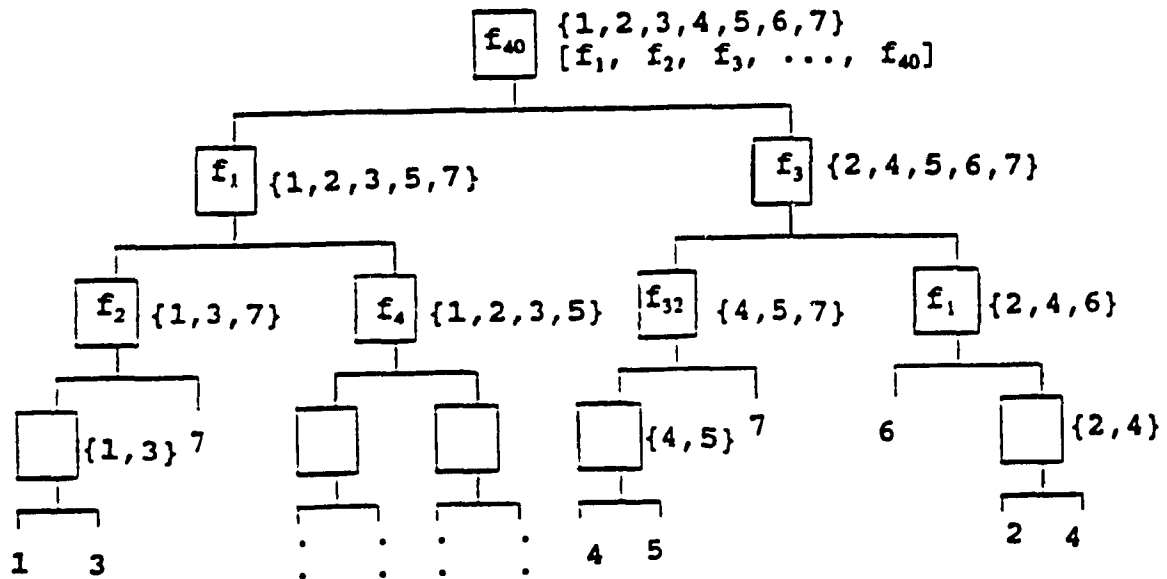


Fig. 5.6 Information-theoretic organization of knowledge and pattern classes.

The EDT classifier consists of a root node, a number of non-terminal nodes and a number of terminal nodes. Figure 5.6 illustrates a hierarchical organization of features and classes. The nodes containing the information enclosed in braces represent a group of classes. The nodes with bare numbers, 1, 2, ..., 7, representing identity of the classes, are the terminal nodes whereas the top node, having the entire set of classes, in braces, into which a sample could possibly be classified, is the root. A nonterminal node has both ascendant and descendant nodes, and as such represents an intermediate decision. The immediate descendant nodes of a non-terminal node represent the outcomes of an intermediate decision. A terminal node corresponds to a terminal decision, i.e., the decision-making procedure terminates and an unknown being classified at this stage is assigned the label of the corresponding class of the node.

#### 5.7.1 Design of the EDT Algorithm

The EDT algorithm consists of five components, 1) the value of knowledge objects used at each non-terminal node, 2) computation of entropy for corresponding classes (or groups) at each non-terminal node, 3) best feature selection based on reduction in entropy between two subsequent nodes, 4) a hierarchical ordering of features based on their information contents, and 5) the decision function to be used at each non-terminal node. The algorithm for the accomplishment of these steps is presented in Fig. 5.7 and the explanation of the procedures for the accomplishment of these steps are described below.

##### 1. Evaluation of Knowledge Objects

The knowledge objects are derived from the analytical features. Originally, the analytical features available were

- Step 1.  
Initialize the root node with the input group consisting of the entire design set.
- Step 2.  
Traverse the tree, beginning from the root and using a depth-first search algorithm until a decision node with a terminal output group is found; assign the label of the output group to this node, i.e., becomes a designated class, and go to Step 3. If no such node exists, then EXIT because the tree construction process is completed. Any terminal nodes found during this traversal are ignored.
- Step 3.  
At the current node, compute the entropy of the input group. If entropy is zero, a terminal output group is found; assign the label of the input group to this node, i.e., becomes a designated class, and go to Step 2. If the entropy is greater than zero then the current node is just an intermediary node and go to Step 4.
- Step 4.  
At the current node, use each knowledge object to classify the input group into two output groups, and compute the reduction in entropy using equations 5.7.2 and 5.7.3. Select the knowledge object that produces the largest reduction in entropy.
- Step 5.  
Use the selected knowledge object to create two new child nodes for the next level of the tree, and mark this object as the 'selected feature'. Assign the output groups of the current node as the input groups to the respective child nodes. Repeat steps 2 through 5 until done.

Fig. 5.7 Procedural steps of the EDT Algorithm.

numeric. To construct the knowledge objects we transformed the analytical features into pseudo-symbolic features. The pseudo-symbolic features were developed as discriminating values of the analytical features which were used to split the numeric feature into a feature which is either less than, greater than, or equal to an observed discriminating value. Thus a knowledge object can be defined as a tuple of two



components. One component designates feature identification and the other represents the value of that feature, i.e.,

Knowledge Object :  $(Y_i, v'_i)$

where  $Y_i$  is the label of the  $i$ -th feature, and  $v'_i$  is its discriminating value at a particular stage in the decision tree which is estimated as:

$$v'_{ri} = \begin{cases} \text{less} & \text{if } v'_i < \tau'_i \\ \text{equal} & \text{if } v'_i = \tau'_i \\ \text{greater} & \text{if } v'_i > \tau'_i \end{cases}$$

Each feature was examined to determine its discriminating value  $v'_i$ . This was done by finding minimum and maximum values of each feature in every pattern class. When the range of the feature in one class did not overlap the range of the same feature in another class, two candidate discriminators were identified, i.e.,

$$\text{Range } (\tau'_i) = \begin{matrix} \text{max } v' \\ \text{Val} \\ \text{min } v' \end{matrix} \left\{ \begin{array}{l} \text{Overlap } (v'_{ij}) \dots 5.7.1 \\ \text{for } i = 1, \dots, n \\ j = 1, \dots, N \end{array} \right.$$

where  $\tau'_i$  is a non-overlapping range of a feature's value between two or more classes and  $\text{Overlap } (v'_{ij})$  is a function that returns the limits of the feature value  $v'_{ij}$  which overlaps in two or more classes.

We observed that some features had no candidate discriminators, and hence they were eliminated; some other features had multiple candidate discriminators, and consequently appeared several times, with different ranges of course, in the list of candidate discriminators.

## 2. Entropy Computation

The entropy  $I_n$  of an input group is defined by:

$$I_n = \sum_{i=1}^N [-P(i) \log_2 P(i)] \quad \dots 5.7.2$$

and that  $P(i) = p_i / p_o$

where

$N$  : number of classes in the group  
 $P(i)$  : probability of occurrence of class  $i$   
 $p_i$  : number of patterns in the group from class  $i$ ,  
 $p_o$  : Total number of patterns in the group

The range of  $I_n$  values implies that

$I_n \geq 1$  if a group contains several classes, and each class having an equal number of samples,

$I_n < 0$  if a group contains several classes, and one class is dominant,

$I_n = 0$  if a group contains all samples from one class.

Notice that the entropy will be greater than or equal to 1.0 when a branch contains samples from several classes with each having equal number of samples. The magnitude also depends on the number of classes. When the samples in a group predominantly constitute one class, the entropy will be less than one. Finally, when all samples in a group are strictly from one class, the entropy is zero.

## 3. Best Object (Feature) Selection

For training, all samples in the design set form the initial learning group. The entropy for this group was computed. Then each available feature was used to dichotomize the group. The entropy of the resulting system of groups was found by using equation 5.7.2 and also computing the entropy

of each individual group. The entropy of the output system for each branch was evaluated using the relation:

$$I_o = \sum_{g=1}^G I_g (p_g / p_o) \quad \dots 5.7.3$$

where

G : number of output groups  
 $I_o$  : total entropy of the output group  
 $I_g$  : entropy of group g  
 $p_g$  : number of patterns in group g  
 $p_o$  : number of patterns in all output groups

The equation 5.7.3 reflects that the entropy of a group is weighed by the proportion of its membership. The feature which produces an output system with the greatest reduction in entropy is the best one to use for the node of the decision tree. Select feature for which the amount of decrease in amount of entropy is maximum, i.e.,

$$Y_1 = \text{Max} (I_n - I_o) \quad \dots 5.7.4$$

Then each of the output group becomes an input group to the descendant nodes of the tree.

#### 4. Building the tree

The initial input group consists of all the patterns in the learning set. As a node is built, each of its output groups become the input group of the descendant nodes. At any given node, the entropy of the input group is computed using equation 5.7.2. Each candidate discriminator is used to produce two descendant nodes. Then entropy of each descendant node is evaluated using equations 5.7.2. and 5.7.3. The reduction in entropy between two subsequent nodes is the new entropy subtracted from that of the input group's entropy and that reduction is saved for each candidate discriminator. When all

candidate discriminators have been tested, the one with the largest reduction in the entropy is saved as the discriminator at the current node. When an output group constitutes all samples of one class, i.e., entropy becomes 0, its node becomes a leaf node, and the identification of the class is saved. When there are no output groups with mixed classes in them, the tree is considered complete, and it was considered as the trained cognition subsystem. Notice that the initial entropy of the system has been reduced at every node of the tree, and the final result is a system with zero entropy.

## 5. Decision Function

The decision function at each non-terminal node is a simple binary decision which assigns a pattern to the left or right branch of the tree based on the entropy of the selected feature.

### 5.7.2 Computational Complexity and Problems

The EDT algorithm is an interesting example of how a restricted representation can simplify the learning process. Because of this restriction the algorithm is able to construct an effective tree in a highly efficient (polynomial time) fashion. The basic task is that of calculating the entropy for each branch. Assuming that there are  $b$  values for each of the  $n$  features, there will be  $b$  computations for each feature. At the next level for each of  $b$  branches,  $b$  values for each of  $(n - 1)$  features would be evaluated, out of which one best would be selected.

Thus the amount of computations would be  $\{b^2 \cdot (n-1) + n-2\}$ . As the tree grows, many branches would lead to one class membership. Assuming that  $L$ -level tree is as complex as  $L/2$  level tree, the computational complexity increases approxi-

mately as  $n.bL$ ; i.e., it increases exponentially with respect to  $L$ , but polynomially with respect to number of features,  $n$ , and their possible values,  $b$ .

One of the problems associated with the types of techniques however, is the complexity involved in the initial construction of the tree particularly when the initial feature set is large and have a large number of pattern samples. This usually would lead to a large size of the tree with multiple paths for the recognition of the same class. Furthermore the successive partitions may not necessarily be unique. In addition, any new knowledge or new samples would require a complete restructure of the system.

Despite the above stated problems involved in the initial construction of the tree, classifying an unknown becomes fairly simple. The unknown traverses the tree straight down one path which would be the order of  $\log L$ , where  $L$  is the level of the tree.

### 5.7.3 Merits of the EDT - Classifier

The EDT technique is more realistic in the sense it uses the features (or knowledge objects) based on their information content. It is practical as it can handle a large feature set and a fairly large number of classes without degrading the performance. Itemized list of its merits is given below:

#### 1. Classification Accuracy

By using only one feature that is the most useful for classification at a given node, the classification accuracy is improved. Also, the dimensionality problem is not as severe as in single layer classifiers which use more features at one time to perform classification.

## 2. Comparison sequence

The algorithm provides a natural and efficient knowledge-dependent order to the classification process, thus suggesting an information-based classification strategy without involving human biases.

## 3. Information accumulation during learning

While building the hierarchy (learning phase), important class/group characteristics can be stored at each node which can be efficiently utilized by the classification process.

## 4. Reduction in size of the problem

By hierarchical organization of classes and features a multi-class problem is reduced into a hierarchy of two-class problems with only one single feature determining the membership of a pattern. The membership assignment can be done by a simple binary decision without using a pattern classification algorithm.

## 5. Natural Grouping of classes

The algorithm provides a natural grouping of classes with the least amount of system designers' biases. Finer segregation of classes is followed as one moves down the hierarchical structure.

## 6. Reduction in computational complexity/cost

Since the decision process is organized hierarchically, the reduction in complexity and subsequent gain in processing efficiency is quite obvious.

### **5.8 Failure Control**

The failure control mechanism is an additional pattern classification scheme and uses the decision theoretic algorithms of the discrimination system in the sense of a single layer classifier. It has been developed to handle the failures of the discrimination system.

The discrimination system may fail at any intermediate node of PAH. In such case the Failure control scheme attempts to classify the unknown pattern one more time using the same parent classifier but against all classes above the PAH-node in question. The Failure control system learns its decision making capability through a supervised learning scheme and uses an unabridged feature set to performs the classification.

## Chapter 6

### Discrimination Subsystem

#### 6.1 Introduction

Discrimination Subsystem is the last major component developed in this research. Its function is to classify a pattern utilizing stored knowledge. To perform this task it uses a number of algorithms. To improve the recognition performance a hierarchical classification scheme, called PAH-classifier based on the PAH concept is introduced. The PAH-classifier primarily dichotomizes the classification process using variable feature sets and allows to train each node for a suitable classifier including the one introduced in this research. In this chapter, different classification methods are reviewed and categorized on the basis of their underlying operating principles. Several parametric and nonparametric classification methods are developed. A number of data dependent rules based on empirical knowledge have been designed for automatic selection of an appropriate classification algorithm. The expert, however, has the option to override the choices made by the system.

#### 6.2 Discrimination Subsystem

The last major component of the system is the Discrimination subsystem. The decision making process that the discrimination system uses can be formally stated as follows:

Let  $X = [x_1, x_2, \dots, x_n]^T$  be an unknown pattern vector of  $n$  characteristic measurements (features), and let  $\omega = (\omega_1, \omega_2, \dots, \omega_N)$  be the set of  $N$  classes. Based on the characteristic measurement values  $x$ 's, a decision making process either assigns the unknown pattern to one of the known classes  $(\omega_1, \omega_2,$



...,  $\omega_N$ ) or rejects it. The decision making process can be developed in a number of ways. We structured several commonly used algorithms and developed a procedural scheme to select an appropriate algorithm based on characteristics of the data, nature of the problem, and the performance objectives of the expert/user. In addition we developed a new classification algorithm, called PAH-classifier that can use any of the existing algorithms.

### 6.3 Classification Methodologies

The development of a good decision making process in the sense of minimum classification error has been one of the recurring topics of research in the field of pattern recognition. Methods from various theoretical and applied fields, such as mathematics, statistics, theory of formal languages, graph theory, heuristics, etc., have been explored and applied to solve pattern classification problems.

The selection of a classification method depends on the domain of an application, objectives of the designer, size of the problem and the types of features, numerical or structural, extracted from the pattern. For example, if the extracted features represent structural properties, and the application at hand requires both the description and classification, then the syntactic classification scheme, discussed in the forthcoming sections, appears to be a reasonable choice. However, when only discrimination between classes is the objective, pure numeric approaches are emphasized; although in some situations very poor performance is reported [FUKS-82]. Many different classification approaches have been reported in the PR literature. These approaches can be loosely placed in four categories, namely, decision-theoretic, information theoretic, syntactic, and graph-theoretic. These approaches are briefly reviewed in the sections to follow.

Although these methods are applicable to ar. general types of patterns, the aim of this review and the objectives behind the development of new methods are: 1) to highlight their applicability to the signal classification problems, 2) to justify the need for a classification scheme which should consider the knowledge parameters according to their discrimination power, 3) to select a classifier that produces the best training results, and, 4) to justify the need for a classification scheme which should partition the problem space to improve the performance and perhaps may update the knowledge base (adaptive learning) without interfering the decision making process.

### **6.3.1 Decision Theoretic Approaches**

The development of decision-theoretic models for pattern classification has been a major subject of research [FUKU-90]. Two prominent approaches of modeling have emerged as a result of these efforts, parametric and non-parametric approaches. Several methods in each of the categories were developed and their performances were tested on patterns from a large variety of applications [COHE-86b, HAYD-84, SIDD-91c]. These methods may be distinguished from each other on the basis of their underlying assumptions.

The basic concepts involved in the development of different forms of decision functions and the salient characteristics of the parametric and non-parametric methods are briefly reviewed in the following sections. The classification algorithms used by the Discrimination subsystem are primarily from this class of approaches.

#### **6.3.1.1 Parametric Approaches**

Parametric classification methods which are also known as probabilistic classification methods refer to the development of

statistically defined discriminant functions in which the underlying probability density functions are assumed known or may be evaluated from the given data. The most commonly used such assumption is the density function given by normal distribution. It then remains to simply estimate a set of parameters which will then approximate the functional form of the assumed distribution function for the pattern classes. One of the main reasons for selecting normal statistics is the relative ease with which analyses can be handled under such assumptions. However, the parametric PR machine will only be as useful as the validity of known (or estimated) underlying class densities.

If the prototypes and unknowns do not conform to the assumed statistics, the classification accuracy will suffer accordingly as the underlying distribution functions may not necessarily reflect the distribution of the characteristic measurement vectors obtained from the samples in the design set.

To implement these methods, each class, say,  $\omega_j$  is associated with a priori probability  $P(\omega_j)$  and conditional density  $P(X/\omega_j)$ , for  $j=1,2, \dots, N$ . There are several ways one can estimate a priori and conditional probabilities [DUDA-73, NADL-93, SIDD-81, TOU-74]. Based on  $P(\omega_j)$  and  $P(X/\omega_j)$ , the function of a classifier is to test  $N$  statistical hypotheses that an unknown pattern  $X$  belongs to the class  $\omega_j$  by defining a decision function  $D_j(X)$ . Bayes theorem is generally used in defining such decision functions [NILS-65]. Most commonly used form of this decision function  $D_j(X)$  is defined as:

$$D_j(X) = P(\omega_j) P(X/\omega_j) \quad \dots 6.3.1$$

This classification procedure is also known as Bayes optimum decision rule. Thus, using Bayes classification rule an unknown pattern  $X$  is accepted as a member of the class  $\omega_j$  if

$$D_j(X) > D_i(X), \quad \text{for } i, j=1, 2, \dots, N, \text{ and } i \neq j.$$

### 6.3.1.2 Non-Parametric Approaches

Non-parametric approaches in statistical decision models are often resorted to when underlying probability densities are unknown or the data do not follow any of the known distributions. Thus if a priori knowledge of the problem does not lend itself to a safe density assumption, a variety of non-parametric procedures may be utilized to develop the discriminant functions necessary for classification. These are simple and intuitive approaches and can be roughly categorized into probability estimation methods, direct decision methods, transformation methods, and adaptive decision methods. Direct and adaptive decision methods are the most commonly applied non-parametric methods and are dealt with in detail in the following sections. Other methods are as follows. Probability estimation methods require estimating the density functions  $P(X/\omega_j)$  from sample patterns. If these estimates are satisfactory, they are substituted for the true densities in designing the classifier. Another method in this area consists of directly estimating a posteriori probability  $P(\omega_j/X)$ . The latter method is closely related to direct decision methods. The transformation methods primarily transform feature space so that the parametric methods may be applied in the transformed space.

#### 6.3.1.2.1 Direct Decision Functions

These methods bypass the estimation of a posteriori probability and directly evaluate the decision function. The decision function is usually assumed known or decided by the designer. Nearest neighbor methods and minimum distance classifiers are some of the popular methods among direct decision methods. In general these methods work as follows.

Let  $\chi(p) = \{X_1, \dots, X_p\}$  be a set of  $p$  labeled samples, and let  $X_k$  is the  $k$ -th sample of  $\chi(p)$  and assume that it is nearest to  $X$ . The nearest neighbor rule for classifying  $X$  is to assign it the label associated with  $X_k$ . The number of nearest neighbors may be varied. When the decision is based on a single nearest neighbor the method is called minimum distance method, otherwise it is based on identity of the majority of the nearest neighbors. The nearest neighbor rule and minimum distance methods are suboptimal procedures; their use usually leads to an error rate greater than the minimum possible, the Bayes rate.

#### 6.3.1.2.2 Adaptive Decision Functions

In adaptive methods it is usually assumed that forms of the discriminant functions are known. The thresholds used to design the decision functions are deterministic and are usually based on the values obtained from the design set. In these methods a pattern classification problem is formulated in terms of one or more discriminant functions.

Suppose that  $n$  characteristic measurements  $X = [x_1, x_2, \dots, x_n]^T$  represent a pattern in  $n$ -dimensional feature space. Then, discriminant functions  $D_j(X)$ 's associated with pattern classes  $\omega_j$ 's, for  $j=1, 2, \dots, N$ , partition the measurement space into  $N$  mutually exclusive regions, where each region corresponds to a particular pattern class. This is an ideal situation and it may happen only when classes are distinctly apart. The classification procedure using discriminant analysis is as follows.

Given an unknown pattern  $X$  and the discriminant function  $D_j$ , the classification process is to assign the unknown to class  $\omega_i$ , if  $D_i(X) > D_j(X)$ , for  $i, j = 1, 2, \dots, N$ , and  $i \neq j$ . To classify  $N$  classes maximum  $N-1$  discriminant functions would be

required. Several functional forms of discriminant functions have been proposed in the literature [DUDA-73, SCHA-92]. Commonly used forms are linear, piece-wise linear and polynomial discriminant functions. A linear discriminant function  $D_j(X)$  is a linear combination of feature elements, that is,

$$D_j(X) = \sum_{k=1}^n w_{jk} \cdot x_k + w_{j0} \quad \dots 6.3.2$$

where  $W_j = [w_{j1}, w_{j2}, \dots, w_{jn}]$  is a weight vector estimated from the design set of the  $j$ -th class and  $w_{j0}$  is a constant. Several parametric and nonparametric approaches to estimate these weight vectors are described in Section 4.5. Linear discriminant functions have been used by Hay et al. [HAYD-84], Siddiqui et al. [SIDD-88, SIDD-89a], and several others for signal classification [COHE-86b, SETH-82].

Linear discriminant functions are relatively easy to implement. One of their major disadvantages is that they assume that individual patterns are separated by neat and clean class boundaries, which in general, is an unrealistic assumption. To achieve the maximum separability, use of piecewise linear discriminant function is usually suggested. In the piecewise linear discriminant approach more than one weight vector for each class is used. This approach is formally described as follows.

Suppose that  $W_1, W_2, \dots, W_N$ , are the  $N$  sets of weight vectors associated with  $N$  classes  $(\omega_1, \omega_2, \dots, \omega_N)$ , respectively, and that the weight vectors in a set  $W_j$  are denoted as  $W_j^k$ , for  $k = 1, 2, \dots, u_j$ , where  $u_j$  is the number of weight vectors in the set  $W_j$ . A piecewise linear discriminant function is defined as:

$$D_j(X) = \text{Max}_{(1 \leq k \leq u_j)} (D_j^k) \quad \dots 6.3.3$$

where  $D_j^k$  is the  $k$ -th discriminant function for the  $j$ -th class.

An alternate formulation to this approach considers  $N$  sets of weight vectors  $W_1, W_2, \dots, W_N$  as the set of reference vectors from  $N$  classes  $(\omega_1, \omega_2, \dots, \omega_N)$  respectively, and the distance  $d_j^k(X, W_j^k)$  between  $k$ -th reference vector of the  $j$ -th class and the unknown pattern  $X$  is used to define the discriminant function as follows:

$$D_j(X) = \min_{(1 \leq k \leq u_j)} (d_j^k(X, W_j^k)) \quad \dots 6.3.4$$

The nearest neighbor method discussed in [COVE-67] is an example of this approach. The application of this technique in signal processing and its suitability in dealing with the problems, such as, representative reference vector selection [ANDE-73, ANDR-58] and optimum decision-making criterion can be found in [DUDA-66, DEVI-82, NADL-93].

Another functional form which is termed as polynomial discriminant was developed to attain maximum separability through the nonlinear discriminant functions, especially when the classes are not linearly separable. Generally, a polynomial discriminant function is defined as:

$$D_i(X) = \sum_{j=1}^n w_{ij} \cdot \phi_j(X) \quad \dots 6.3.5$$

where  $\phi_j(X)$  is a function of the characteristic measurement vector  $X$ .

Several methods based on orthogonal expansion, least square approximation and stochastic approximation, etc., have been suggested [MEIS-68, SPEC-67, ULLM-73] to obtain the functional form of  $D_i(X)$ . The approaches presented above may perform well for non-overlapping classes, however, in practice, it is hardly the case. To deal with this problem other methods were

investigated [DUDA-73, FUKU-90, SCHA-92], some of which are presented in the following sections.

### 6.3.2 Information - Theoretic Methods

Entropy is a statistical measure of uncertainty and can be used to measure the intra-class dispersion and is given by:

$$H = - \sum_i \{ p_i \ln p_i \} \quad \dots 6.3.6$$

where  $p$  is the probability density of the pattern population, and  $E_p$  is the expectation operator with respect to  $P$  [TOU-74, KOSK-92]. This concept can be used to design a pattern classifier using features which minimize the entropy of the pattern classes under consideration. The EDT classification algorithm that we developed in Chapter 5 is a good example of this concept.

### 6.3.3 Syntactic Approaches

In various pattern recognition applications, along with the discrimination, the description of a pattern is also required. The syntactic approaches were primarily developed to fulfil these requirements. Considerable theoretical as well as applied studies on this subject have been reported in PR literature [ALI-77a, ALI-77b, FUKS-82, FUKS-86, KRAM-73, LINC-86, NARA-69, YOUN-86, ZHAN-80]. In this approach a pattern is viewed as complexes of primitive structural elements, usually called primitives. The relationships among the primitives are defined using syntactic rules. The primitive structural parts are perceptually higher level objects than scalar numerical measurements. In practice, structural approach involves a set of independent processes: 1) identification and extraction of primitives; 2) identification of relationship to be defined among primitives; 3) recognition of allowable structures in terms of the primitives; and, 4) the relationship among structures. These processes are jointly used to design a



syntactic classifier. The classifier is basically a syntax analyzer that classifies an unknown pattern into one of the known classes and provides an adequate description, implicitly or explicitly, of the underlying pattern. A number of structural methods have been reported in [FUKS-82, LINC-86, PAVL-77, SKOR-86].

#### **6.3.4 Other Decision Making Strategies**

In addition to the techniques described in previous sections there have been several other efforts which explore methods from other areas such as graph theory and heuristics. No clear boundaries exist between the graph-theoretic and heuristic approaches except that heuristic approaches may either apply ingenious ways to combine established PR methods or use pure heuristic decision rules such as production rules to capture the common sense decision making process whereas the graph-theoretic approaches are based on well postulated axioms from graph theory [FARI-83, KATS-69, NADL-93, SCHA-92, WATT-71].

##### **6.3.4.1 Graph Theoretic Approaches**

Graph theory has been applied to classify patterns in many different ways. These techniques differ from each other in respect of pattern representation and decision making procedures. Generally, patterns are represented as graphs with or without attributes. In a graph representation without attributes, the special points in a pattern, such as end points, junction points etc., are considered as vertices and lines joining these vertices are regarded as edges of a graph. On the other hand, in a graph representation using attributes, called attributed graph, rules are prescribed for assigning attributes to each vertex or edge. The decision making process used in this approach may be stated as follows:

Let  $U_x$  and  $U_r$  be graphs constructed from an unknown pattern and the reference pattern respectively. If  $U_x$  is isomorphic to  $U_r$  then  $X$  is assigned to the  $r$ -th class. In case  $U_x$  and  $U_r$  are attributed graphs and if there exists an isomorphism between  $U_x$  and  $U_r$  such that the attributes of corresponding vertices or edges differ only within a prescribed threshold, then  $X$  is still assigned to the  $r$ -th class.

It is evident that in order to establish the isomorphism between two graphs, vertices and edges needed to be compared which is a time consuming task. A special form of graph known as the decision tree may be used to minimize the classification time. Decision trees are constructed using features extracted from the training patterns. The features extracted from an unknown pattern  $X$  are compared with the features stored at each node of the decision tree. Depending upon the comparison result of a randomly selected feature of an unknown pattern at the root node, a path to an expected subclass is selected. Once the expected class is established, the features relevant to that subclass are tested and further decision is made about the unknown pattern. The possible decisions at each node include the correct classification of an unknown pattern, return to root node, or move one level up/down to a node for further processing.

This process continues until an unknown is either classified, misclassified, or rejected as an invalid pattern. This decision making process involves the search of the known pattern. Thus, the design of a decision tree can affect the classification speed. The optimal design of decision trees has been an important topic of research and several methods have been proposed in [DATT-80, LIN-80, NADL-93, SCHA-92, SWAI-77, WANG-84].

#### **6.3.4.2 Heuristic Approaches**

Heuristic approaches constitute either an adhoc solution or a composite of several decision-theoretic approaches or syntactic methods. Adhoc approaches usually apply AI methods in designing the decision rules. One such approach uses production rules to capture the common sense decision making process [CHAS-88]. Another set of approaches combine statistical and/or mathematical approaches with syntactic approaches [BLAC-74, DUER-80]. Several hybrid approaches combining decision-theoretic and syntactic methods in different orders have been proposed in the literature [FUKS-82, KANA-72, NADL-93, TSAI-80].

#### **6.4 Trends in Decision Making Process**

In general, the basic function of a decision making process is application dependent. To apply these algorithms in a general sense, however, it is required that every pattern recognition system should, 1) incorporate the knowledge provided by the statistical (numerical) and structural characteristic measurements, 2) be flexible in learning the variations in patterns, and, 3) be able to classify large class (size) problems without degrading the performance.

A solution to the first issue might help in developing a system which can perform satisfactorily in real life applications. One solution suggested by the researchers is the combinations of structural and statistical approaches [KANA-72, BLAC-74, TSAI-80, DUER-80]. The problem with these approaches lies in both feature extraction and classification since two different types of feature extraction schemes and two different classification schemes would be required. Another solution which has been emphasized throughout this thesis and originally suggested by the author in an earlier article [SIDD-87a] is the incorporation of the knowledge

provided by the physical observations and expert knowledge.

A solution to the second issue is usually addressed by considering a large training set of sample patterns [FUKU-89]. It is assumed that the sample patterns in a design set reflect all possible variations. A priori knowledge obtained from these samples is used in the form of weight vectors, probability distribution parameters, grammars, prototypes, graphs or decision rules, etc. A formal knowledge representation method which allows update and organization of knowledge, without any change in the decision logic, is required. It should be noted that the decision making approaches based on syntax or decision trees need reorganization of decision making process for every pattern not perceived during the training session of the classifier. The knowledge organization and representation schemes described in Chapters 2 through 4 present a formal method for the elimination of some of the problems in classification methods discussed above.

The solution to the third problem is usually suggested by using the decision trees [DAT-80,EAST-91,FUKU-75,QUIN-88]. These approaches perform the classification task which can be implemented either, 1) by successive identification of a feature that could partition the pattern space; successive partitions may not necessarily be unique, or, 2) by successive dichotomization of the pattern space. Both of these approaches are described in the following sections.

### **6.5 Classification (Search) Strategies**

The classification or the class search strategies of the existing pattern classification algorithms can be broadly grouped into two categories, hierarchical or multi-layered and non-hierarchical or single-layered. Most approaches discussed in Section 6.3 above, use single layer strategies which class-

ify a given set of patterns into a predetermined number of classes in one step (layer). Such approaches have significant drawbacks.

Among numerous drawbacks, the significant ones are, 1) only one of the possible combinations of pattern features is used for the classification of all the classes, 2) the same classification strategy needed to be used for all unknown patterns, 3) if rejection is to be incorporated, there is no way to establish rejection against which class, 4) each unknown sample is tested against all classes which may cause higher rejection or misrecognition, 5) for large problems (in terms of number of classes and number of features) such classifiers tend to become computationally more complex and expensive with proportionately degrading performance. All these factors lead to a relatively high degree of inefficiency, misclassification, and rejection and as such they are incapable of providing any reasonably general solution for a pattern classification problem in a given domain.

In single-layer strategies, the use of only one feature subset is inevitable and as a consequence some features which are pertinent for discriminating between some classes are not selected since they may not be useful in discriminating other classes, whereas a few other features which may be marginally contributory over a large set of classes may get selected. Therefore, this overall 'best feature set' selected for classifying input samples into classes over a problem space may not be the best features for discriminating between specific groups or pair of classes. The problem turns even more severe when there are a large number of classes, since a larger feature set would be required for classification. Another problem of using a large number of features is the need for corresponding increase in the number of training and testing samples so that, statistically speaking, the results

obtained may still be reliable. Furthermore, some patterns may not need all the features in order to arrive at correct classification, but a single layer classifier measures these features any way which usually decreases the efficiency.

On the other hand the hierarchical strategies, which have recently become a useful decision tool, usually reduce the search space by partitioning the single-layered decision making strategy into a hierarchy of decisions and as such, at least, guarantee the efficiency in decision making. As described in previous section these strategies can be implemented in several possible ways. The most popular approach has been the successive dichotomization of the pattern space through successive identification of a feature or features that could partition the pattern space; successive partitions may not necessarily be unique. For several reasons discussed below this approach was not used in designing the search strategy of the Discrimination system. Instead, we used the PAH concept to partition the pattern space.

## **6.6 Hierarchical Decision Approaches**

The graph theoretic methods and the methods based on tree are some of the well known examples of hierarchical methods. A number of examples of methods based on this approach are described in Section 6.3.4.1. A majority of these approaches are based on hierarchically selecting features and then using a decision function to determine which class or classes can be correctly recognized. The problem associated with these techniques is the large size of the tree and multiple paths for the recognition of the same class. Another problem is that successive partitions may not necessarily be unique. The hierarchical classification scheme developed for the Discrimination system not only solves most of the problems associated with single layered schemes but also those associated with

traditional hierarchical schemes. To the best of our knowledge, there is no hierarchical technique present which is capable of using more than one classifier or feature set at different nodes and layers of the tree and offers a unique classification path for each class. Furthermore, the technique is more realistic and practical as it can handle a large feature set and a fairly large number of classes without degrading the performance.

#### 6.6.1 The PAH - Classifier

The PAH-classifier is a tree classifier and consists of a root node, a number of non-terminal nodes and a number of terminal nodes. Figure 6.1 illustrates a hierarchical organization of

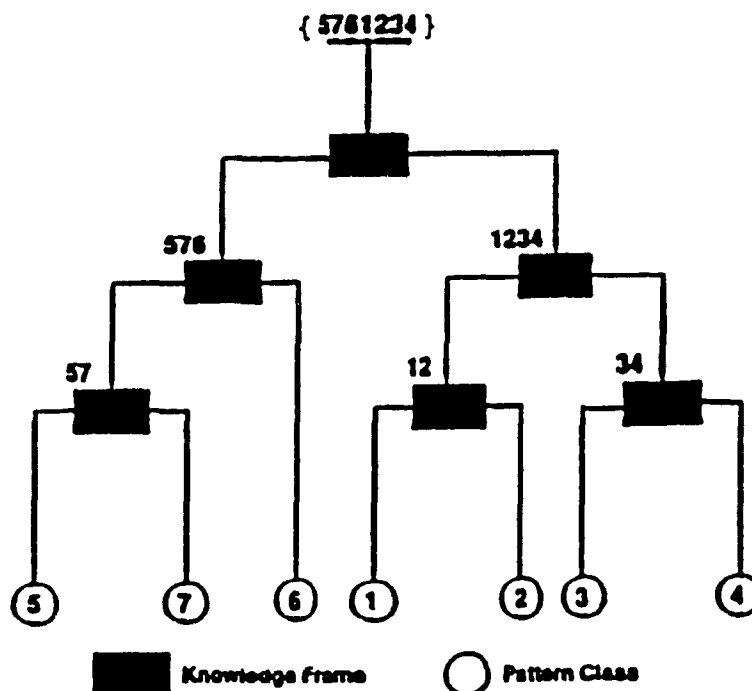


Fig. 6.1: PAH Classifier Design.

the classes which is the basic information utilized by the PAH-classifier. The nodes containing the information enclosed in braces represent a group (cluster) of classes. The nodes with bare numbers, 1, 2, ..., 7, representing identities of the classes, are the terminal nodes whereas the top node, having the entire set of classes, in braces, into which a sample could possibly be classified, is the root. A non-terminal node has both ascendant and descendant nodes, and as such represents an intermediate decision. The immediate descendant nodes of a non-terminal node represents the outcomes of an intermediate decision. A terminal node corresponds to a terminal decision, i.e., the decision-making procedure terminates and an unknown being classified at this stage is assigned the label of the corresponding class of the node.

At each node, in fact, any single-layered classifier can be used, which could either be the same for the entire tree or different for different nodes. Thus a tree classifier, using the appropriate features and a selected nodal classifier classifies an unknown by starting the unknown sample at the root and traversing a path of the tree, where each non-terminal node encountered invokes the nodal classifier and decides the subsequent path until the unknown ends at a terminal node, whose identity is the label assigned to the unknown sample.

#### **6.6.2 Design of the PAH-Classifier**

The PAH-classifier may be considered as consisting of five components, 1) a hierarchical ordering of the pattern classes, 2) the number of features used at each non-terminal node, 3) selection of features for corresponding classes at each non-terminal node, 4) the classifier to be used at each non-terminal node, and 5) the training of every node for respective nodal classifier.



The first step is accomplished by using hierarchical clustering algorithms and has already been described in Chapter 3. This step also determines the number of samples to be used at each node. Based on the tree structure obtained in Step 1, the number of features to be used at each internal node can be determined using theoretical rules described in Chapter 4 (see Section 4.2.1). This solves the second step. The third step is carried out using either the Fisher ranking algorithm or the algorithm proposed in [SIDD-90a]. Both of these algorithms are described in Chapter 4 (see Sections 4.3 and 4.4). The choice of the classification algorithm at the fourth step is based on the problem specifications which can be established by applying the rules described in Section 6.9, however, the classification procedure, in general, is described below. The fifth step requires one to select appropriate size of the training set to justify the practicality of the system and this has already been discussed in Chapter 3. Additional training issues and algorithms are already described in Chapter 5.

The mathematical formulation of the PAH classification procedure is as follows. Assume that  $D$  is the optimal decision function, with equal a priori probabilities (and a zero-one loss function) for testing class (group) pair  $G_i, G_j$ , at each internal node and  $\Omega$  is the decision of  $D$ , for all  $i, j = 1, \dots, 2N-1$ , and  $i \neq j$ , we have,

$$\Omega = D(G_i, G_j) \quad \dots 6.6.1$$

with

$$\Omega = \begin{cases} G_i & \text{if } r_{ij} \geq \tau \\ G_j & \text{otherwise} \end{cases} \quad \dots 6.6.2$$

where  $r_{ij}$  is the nodal classifier-dependent decision criterion and  $\tau$  is the corresponding threshold evaluated on the design set. With  $\Omega$  and  $D$  defined above, the PAH classification procedure can be defined in recursive form:

$$\Omega_i = D (G_i, \Omega_{i-1}) \quad \text{for } i=2, \dots, N \quad \dots 6.6.3$$

with  $\Omega_1 = G_1$ ,

where  $N$  is the number of classes. The recursive formula of  $\Omega_i$  starts with  $\Omega_2$ ; and  $\Omega_N$  is the final decision which determines to which class the unknown sample belongs.

### 6.6.3 Computational Complexity

Each of the steps of the PAH-classifier discussed above involves its own level of complexity. The complexity of hierarchical ordering depends on the number of pattern classes and the number of samples in each class; how the tree is built, bottom-up or top-down; and also on the number of features and the procedure used for ordering the classes. The second function that adds on to the complexity of the classifier is the training of the system. This factor is primarily based on the number of features, feature weighing mechanism, and the kind of classifier used at each non-terminal node of the tree. Finally, the classification of an unknown involves additional complexity which is based on the size of the path the process has to traverse in reaching a leaf-node. Thus, the computational complexity of a PAH-classifier, i.e.,  $C(\text{PAH})$  will be:

$$C(\text{PAH}) = C \{ \text{Ord} (\text{PAH}) \} + C \{ \text{Train} (\text{PAH}) \} + \sum_{i=1}^{n-1} C \{ \text{Class} (\text{node}) \} \quad \dots 6.6.4$$

where  $C \{ \text{Ord} (\text{PAH}) \}$  is the amount of complexity involved in ordering the initial pattern association hierarchy,  $C \{ \text{Train} (\text{PAH}) \}$  is the complexity involved in training the classifier, and  $C \{ \text{Class} (\text{node}) \}$  is the amount of complexity of the classifier selected at a node. The complexity  $C \{ \text{Ord} (\text{PAH}) \}$ , for example, for a bottom-up tree building process would require  $T_b$  computations.

$$\begin{aligned}
 T_b &= \sum_{i=2}^N i(i-1)/2 + C_f & \dots 6.6.5 \\
 &= \{N(N-1)(N-2)\}/6 + C_f
 \end{aligned}$$

In above the first term on the right hand side gives the number of similarity indices the clustering method,  $f$  would compute to organize  $N$  pattern classes whereas  $C_f$  is the complexity of the clustering method,  $f$  itself. Assuming that there are  $n$  features in a pattern vector, the centroid clustering procedure just using Euclidean distance would require  $n$  multiplications and  $n$  additions, i.e.,

$$C_f = O(n) \quad \dots 6.6.6$$

The complexity of the classification process depends on the shape of the pattern association hierarchy, the association may generate a balanced tree or a skewed one. If  $N$  is the number of classes, then PAH-classifier makes at most  $N-1$  comparisons (classifications) for a successful search for a pattern class and  $N-1$  comparisons for an unsuccessful search. In other words the time for a successful search or for an unsuccessful search is close to  $O(n)$ . In the best case for both a successful search and for an unsuccessful search number of comparisons is one. To determine the average behavior we need to look more closely at the PAH tree, equating its size to the number of element comparisons in the algorithm. The distance of a node from the root is one less than its level. The internal path length,  $I$ , is the sum of the distances of all internal nodes from the root. The external path length,  $E$ , is defined analogously as the sum of the distances of all external nodes from the root. It is easy to show by induction that for any binary tree with  $q$  internal nodes, and path lengths  $E$  and  $I$ , the quantities are related by the formula,

$$E = I + 2q \quad \dots 6.6.7$$

It turns out that there is a simple relationship between  $E$ ,  $I$  and the average number of comparisons in a binary tree. Let  $S(q)$  be the average number of comparisons in a successful search and  $U(q)$  the average number of comparisons in an unsuccessful search. The number of comparisons needed to classify at an internal node is one more than the distance of this node from the root. Hence,

$$S(q) = 1 + I/q \quad \dots 6.6.8$$

The number of comparisons on any path from the root to an external node is equal to the distance between the root and the external node. Since every binary tree with  $q$  internal nodes has  $q+1$  external nodes, it follows that:

$$U(q) = E/(q+1) \quad \dots 6.6.9$$

Using these formulas for  $E$ ,  $S(q)$ , and  $U(q)$  we find that,

$$S(q) = (1 + 1/q) U(q) - 1 \quad \dots 6.6.10$$

The minimum value of  $S(q)$  and  $U(q)$  is achieved by an algorithm whose binary decision tree has minimum external and internal path length. This minimum is achieved by the binary tree whose external nodes are  $o$ . adjacent levels.

It follows that  $E$  is proportional to  $q \log q$ . Using this in the preceding formulas, we conclude that  $S(q)$  and  $U(q)$  are both proportional to  $\log q$ . Therefore, the complexity of the classification process for traversing the PAH-tree is approximately proportional to  $\log q$ . However, the complexity of an algorithm in branching a pattern to left or right of the tree depends upon the classification algorithm used at various nodes. As described earlier any classification algorithm can be used at each non-terminal node of the PAH-tree. The complexity of various methods used in this thesis for classifying a pattern for an  $N$  pattern class problem is given in

Table 6.1. In the table  $A_i$ 's denote the approximate average complexity of the corresponding method for evaluating the decision function. In these computations a single pattern for each of the reference class is assumed, however, for KNN method  $p_i$  samples for each of the  $N$  classes are considered.

Table 6.1  
Complexity of Unweighed Pattern Classifiers

Method	Classifier	Complexity	Comments
MDC	$NA_M + N/2$	$A_M$ is the complexity of the decision criterion used; for e.g., for Euclidean distance it is equal to $O(n)$ .	
KNN	$N \cdot p_i A_k + O(n-k)$	$A_k$ is the complexity of evaluating the neighborhood; for e.g., for Euclidean distance it is equal to $O(n)$ .	
LDC	$NA_L + n$	$A_L$ is the complexity of evaluating the linear discriminant function; for e.g., using the pooled covariance $n+1$ coefficients of the function require $O(n)$ computations.	
QDC	$NA_Q + On^2$	$A_Q$ is the complexity of evaluating the quadratic discriminant function; for e.g., using the individual class covariances $n+1$ coefficients of the function require $O(n^2)$ computations.	
BYC	$NA_B + P_B$	$A_B$ is the complexity of Bayesian classifier, if log probabilities are used it is equal to $O(n)$ . $P_B$ is additional one time cost of evaluating the a posteriori probabilities and approximately requires $O(n^2)$ computations.	

#### 6.6.4 Merits of the PAH - Classifier

The five steps of the PAH-classifier bring about several performance advantages, the important ones are described below:

##### 1. Classification Accuracy

Since any number of features can be used at a non-terminal node; by using only features that are pertinent for classification at a given layer, the classification accuracy will be improved. Also, since a small number of features are used at one stage, the dimensionality of the problem is not as severe as in single layer classifiers which use more features at one time to perform classification.

##### 2. Natural sequence of comparison

It provides a natural class-dependent order to the classification process, thus suggesting a problem-based classification strategy.

##### 3. Information accumulation during learning

While building the hierarchy (learning phase), important class or group characteristics can be stored at each node which can be efficiently utilized by the nodal classifier and thus supplementing the classification process and pertinent information collection with ease.

##### 4. Reduction in size of the problem

By hierarchical organization of classes and features a multi-class problem is reduced into a hierarchy of two-class problems. A two-class problem is much simpler problem and generally does not require a complex classifier or a large feature set. Thus the selection of an appropriate classification algorithm and the locally optimal feature subset at every node is realistically possible.

##### 5. Natural Grouping of classes

The algorithm provides a natural grouping of classes with the least amount of system designers' biases. Finer segregation of classes is followed as one moves down the hierarchical structure.

## 6. Reduction in computational complexity/cost

Since the decision process is organized hierarchically, the reduction in complexity and subsequent gain in processing efficiency is quite obvious.

### 6.7 Nodal Classifiers

The decision-theoretic type of pattern classifiers were used at different nodes of the PAH-tree to meet a wide range of decision objectives. The following classifiers represent linear, parametric, and non-parametric approaches:

- Empirical Bayesian classifier
- K-nearest neighbor classifier
- Minimum distance classifier
- Linear discriminant function classifier
- Quadratic discriminant function classifier

The basic principles underlying these methods, in general, have been discussed earlier in this chapter. In the following sections the methodology used to develop these methods is discussed.

#### 6.7.1 Empirical Bayesian classifier

Empirically a Bayesian classifier could be designed using a variety of approaches for the estimation of probabilities. To design a Bayesian classifier assume that:

$P(r)$  = a priori probability of input pattern belonging to a class  $r$

$P(X)$  = probability of input pattern being the particular pattern  $X$

$P(X|r)$  = conditional probability of the input pattern being the particular pattern  $X$ , given that it belongs to class  $r$

$P(r|X)$  = conditional probability of the input pattern belonging to class  $r$ , given that it is the particular pattern  $X$ .

To minimize the probability of misrecognition, an unknown pattern  $X$  should be assigned to class  $s$  such that,

$$P(s|X) > P(r|X) \quad \text{for all } r, \text{ and } r \neq s.$$

According to the Bayesian rule,

$$\{ P(s) P(X|s) / P(X) \} > \{ P(r) P(X|r) / P(X) \}$$

Since  $P(X)$  is independent of recognition class, it can be omitted, thus,

$$P(s) P(X|s) > P(r) P(X|r) \quad \text{for all } r, \text{ and } r \neq s.$$

The assumption of statistical independence of features leads to a practical method of estimating  $P(X|r)$ . If there are  $n$  features in each pattern,  $P(X|r)$  can be expressed as:

$$P(X|r) = \prod_{i=1}^n P(x_i | r) \quad \dots 6.7.1$$

Each  $P(x_i | r)$  can be estimated individually, and the maximum likelihood classification rule can be expressed as:

$$P(s) \prod_{i=1}^n P(x_i | s) > P(r) \prod_{i=1}^n P(x_i | r) \quad \dots 6.7.2$$

for all  $r$ , and  $r \neq s$ .

For further simplification, we assume that a priori probabilities of all pattern classes are equal, then we have:

$$\prod_{i=1}^n P(x_i | s) > \prod_{i=1}^n P(x_i | r) \quad \dots 6.7.3$$

for all  $r$ , and  $r \neq s$ .

Using monotonic function of the above, we have:

$$\sum_{i=1}^n \log P(x_i | s) > \sum_{i=1}^n \log P(x_i | r) \quad \dots 6.7.4$$

for all  $r$ , and  $r \neq s$ .



The expression 6.7.4 is a general rule and can be approximated in several possible ways. We approximated the distribution of  $P(x_i | r)$  by an empirical method which required a training set of patterns for each class.

Let  $x_{ir}$  denote the  $i$ -th feature of a member in class  $r$ . Also assume that the value of  $x_{ir}$  varies in the range of  $v_{ir}$  which can be split into  $b$  non-overlapping regions of variable sizes. Determine the number of samples in each region. Using the labeled patterns of the design set a training matrix MAT was computed, each element of which is given by:

$$\text{MAT} (ijr) = \log P (x_{ij} = v_j | r) \quad \dots 6.7.5$$

where  $x_{ij}$  is the  $j$ -th region of feature  $x_j$ , with  $1 \leq j \leq b$ ,  $1 \leq i \leq n$ , and class  $r$ . Therefore,

$$P(x_i | r) = \sum_{j=1}^b \log P (x_{ij} | r) \quad \dots 6.7.6$$

From the training matrix,  $\log P(X | r)$  can be computed for any given feature vector  $X = [x_1, x_2, \dots, x_n]^T$  whose value is, say,  $[v_1, v_2, \dots, v_n]$ . Assuming class conditional independence:

$$\begin{aligned} \log P (X | r) &= \sum_{i=1}^n \log P (x_i | r) \\ &= \sum_{i=1}^n \text{MAT} (ij_i r) \quad \dots 6.7.7 \end{aligned}$$

Using this method,  $\log P (X | r)$  was estimated for all the pattern samples in the design set. Substituting the expression 6.7.7 in 6.7.4 empirical Bayesian classification rule is obtained. The probability for any unknown belonging to a particular class can be estimated simply by adding the log probabilities of feature  $x_i$  having certain value  $v_i$ . Using

this rule  $X$  is assigned to class  $s$  such that

$$\sum_{i=1}^n \text{MAT} (ij,s) > \sum_{i=1}^n \text{MAT} (ij,r) \quad \dots 6.7.8$$

for  $i,j = 1, \dots, N$ , and  $i \neq j$ .

### 6.7.2 K-Nearest Neighbor Classifier

The K-Nearest Neighbor (KNN) classification is a standard method and is especially marked out by its simplicity [COVE-67, MEIS-72]. Let  $\chi(p_o) = \{X_1, \dots, X_{p_o}\}$  be a set of  $p_o = \sum_{i=1}^N p_i$  labeled samples, where  $p_i$  is the number of samples

in class  $i$  and that there are  $N$  pattern classes. Let also that  $X_k$  (a member of  $\chi(p_o)$ ) be the sample nearest to  $X$ . The nearest neighbor rule for classifying  $X$  is to assign it the label associated with  $X_k$ . In order to find the nearest neighbors of the unknown it is necessary to compute the distances between  $X$  and all other samples of the design set.

The number of neighbors which are considered for classification is usually denoted by  $K$ . When  $K$  is greater than one a voting scheme (majority rule) is applied to determine the class identity of the unknown. Formally we define a KNN classification rule which assigns an unknown pattern  $X$  to the class of its majority of  $K$  nearest neighbors, i.e.,

$$D(C_i, X) = \min_{i=1}^K \{ D(C_m, X) \} \quad \dots 6.7.9$$

for  $m = 1, \dots, N$

where  $D$  is any distance measure definable over the pattern space. Different distance measuring schemes are already described in Chapter 3. We introduced a reject option. Under this mode of operation, the classification decision is made

when one class  $C_i$  receives a number of votes which is at least equal to the qualifying majority level  $\theta_m$ , otherwise the pattern is rejected.

The KNN-method does not require linearly separable classes and no training is necessary because the recognition performance is not dependent on the training set. New patterns may be added to the data set without difficulties. The main disadvantage of the original KNN-method is the fact that no data compression is possible; all pattern vectors must be stored and many computations are necessary to find the nearest neighbor. The nearest neighbor rule is a suboptimal procedure; its use will usually lead to an error rate greater than the minimum possible, the Bayes rate.

### 6.7.3 Minimum Distance Classifier

Applying the assumption of statistical independence, the probability  $P(X|r)$  can also be estimated from vector  $X$  and parameters  $M_r = [M_{r1}, M_{r2}, \dots, M_{rn}]^T$ , where the  $M_{ri}$ 's are the estimates of the  $i$ -th feature conditional probability  $P(x_i | r)$ . Assuming that the patterns are normally distributed, then:

$$P(X|r) = \frac{\exp [-1/2 (X-M_r)^T S_r^{-1} (X-M_r)]}{(2 \pi)^{n/2} (|S_r|)^{1/2}} \quad \dots 6.7.10$$

Where  $X$  and  $M_r$  are column vectors

$$M_r = [M_{r1}, M_{r2}, \dots, M_{rn}]^T$$

$$X = [x_1, x_2, \dots, x_n]^T$$

and  $S_r$  is an estimate of the  $r$ -th class covariance matrix with  $S_{rij}$  as its elements. For  $i, j = 1, 2, \dots, n$ ,  $S_{rij}$  is the average of  $(x_i - M_{ri})(x_j - M_{rj})^T$  over the whole training set.

Thus, if the  $p$  patterns constituting the  $r$ -th class training set are  $X_1, X_2, \dots, X_p$  and

$$S_{rij} = 1/p \sum_{h=1}^p (x_{hi} - M_{ri}) (x_{hj} - M_{rj})^T \dots 6.7.11$$

$S_{rij}$  is a measure of correlation between  $x_i$  and  $x_j$  in patterns belonging to the  $r$ -th class.  $S_r^{-1}$  and  $|S_r|$  denote the inverse and determinant of  $S_r$  respectively. Assuming that a priori probabilities of classes are equal, equation 6.7.11 becomes:

$$\begin{aligned} \log P(X|r) = & 1/2 [-n \log (2 \pi) - \log (|S_r|) - X^T S_r^{-1} X] \\ & + 1/2 [ 2 X^T S_r^{-1} M_r - M_r^T S_r^{-1} M_r] \dots 6.7.12 \end{aligned}$$

In equation 6.7.12 above,  $-n/2 \log (2\pi)$  is common to all decision functions and can be omitted from maximization. If the covariance matrices of all classes are equal, the terms  $-\log (|S_r|)$  and  $-1/2 X^T S_r^{-1} X$  can also be omitted since they are independent of  $r$ . The maximum likelihood decision rule which is equivalent to Mahalanobis distance then becomes:

$$X^T S_s^{-1} M_s - 0.5 M_s^T S_s^{-1} M_s > X^T S_r^{-1} M_r - 0.5 M_r^T S_r^{-1} M_r \dots 6.7.13$$

for every  $r$ , and  $r \neq s$ , also, for  $i, j = 1, 2, \dots, n$ ,  $S_{rij} = 0$  unless  $i=j$ , and  $S_{rii} = S_{rjj}$  which implies that the features are statistically independent, and the statistical variability of all features is equal. In this case  $S_r^{-1}$  is scalar and can be omitted from expression 6.7.13. Thus,  $X$  will be assigned to the class for which:

$$X^T M_r - 1/2 M_r^T M_r \dots 6.7.14$$

is maximum. This rule is equivalent to minimum Euclidean rule.

Considering every member in the training set as a representation point, the algorithm can determine the distance of an unknown  $X$  from every pattern in the training set. The distances given by equations 6.7.12, through 6.7.14 can also be used to compute the distances for KNN-method. For an arbitrary constant  $K$ , these equations can be used to find the  $K$  nearest patterns to the unknown  $X$ . The pattern  $X$  is thus assigned to the class to which the majority of the  $K$  nearest neighbors belong.

#### 6.7.4 Linear Discriminant Classifier

A discriminant function is a function  $d(X)$  which defines the decision surface. This classifier is a linear combination of feature element which defines a hyperplane to separate one class of signals from another in the feature space. The conditional probability for a given value of  $x_i$  can be found by assuming that the probability distribution is highly likely to peak at the mean value  $m_{ir}$ . This can be approximated by the normal distribution:

$$P(x_i|r) = \exp [-(x_i - m_i)^2] \quad \dots 6.7.15$$

Taking the logarithm,

$$\log P(x_i|r) = -(x_i - m_i)^2 = 2 x_i m_i - x_i^2 - m_i^2 \quad \dots 6.7.16$$

Then for  $n$  features,

$$\sum_{i=1}^n [2 m_{is} x_i - m_{is}^2 - x_i^2] > \sum_{i=1}^n [2 m_{ir} x_i - m_{ir}^2 - x_i^2]$$

Since  $x_i^2$  is independent of a recognition class, omitting this from maximization, we have

$$\sum_{i=1}^n x_i w_{is} + w_{os} > \sum_{i=1}^n x_i w_{ir} + w_{or} \quad \dots 6.7.18$$

$$\text{where } w_{ij} = 2 m_{ij}, \text{ and } w_{oj} = \sum_{i=1}^n -m_{ij}^2$$

The maximum likelihood rule has been reduced to a set of linear functions of the features for each class k.

$$G(X) = w_{kn} x_n + \dots + w_{ki} x_i + \dots + w_{k2} x_2 + w_{k1} x_1 + w_{ko} \quad \dots 6.7.19$$

such that the pattern X is said to belong to class s if

$$G_s(X) > G_r(X) \quad \text{for all } r, \text{ and } r \neq s \quad \dots 6.7.20$$

and this set of linear functions G are known as linear discriminant functions.

#### 6.7.5 Quadratic Discriminant Classifier

A simpler approach is to ignore the problem of estimating the class densities, and concentrate on the problem of estimating the decision surfaces. We may express the discriminant function for class  $\omega_i$  as:

$$g_i(X) = \sum_{j=1}^p g_{ij}(X) \quad \dots 6.7.21$$

where  $g_{ij}(X)$  is the discriminant function associated with the j-th sample of class i.

The Mahalanobis distance of sample  $X_j$  from class i is given by:

$$g_{ij}(X) = (X_j - M_i)^T S_i^{-1} (X_j - M_i)$$

The following nonlinear decision rule was used to design the pattern classifier that discriminates between the classes.

The decision rule employed is called the Quadratic Discriminant Classifier (QDC). The QDC classifier is preferred when the pattern observations either lack the information or it is required to provide data for the loss matrix and a priori probabilities required for minimizing the Bayes risk. This classification scheme implements a quadratic decision boundary to separate the pattern classes. Let  $X$  be a sample pattern. Also, let  $N$  be the total number of classes. Let  $M_i$  and  $S_i$  be the mean vector and covariance matrix for class  $i$ ,  $i=1, \dots, N$ . The QDC rule with the assumption of multivariate normal distributions, the QDC assigns sample  $X$  to class  $k$ , if,

$$(X - M_i)^T S_i^{-1} (X - M_i) \dots 6.7.22$$

is minimum for  $i = k$ .

The expression 6.7.22 is biased by a constant  $e$  which is chosen such that the number of samples misclassified in the design set is minimized, thus 6.7.21 can be written as:

$$d_i^2 = (X - M_i)^T S_i^{-1} (X - M_i) + e \dots 6.7.23$$

The decision rule 6.6.22 represents the general form of a minimum distance classifier with distance metric  $d_i^2$  and also could be interpreted as Bayes' classifier or maximum likelihood classifier with a Gaussian assumption of the distribution of features. For Bayes classifier,  $e$  in 6.7.23 is  $\log (|S_1|/|S_2|) - 2 \log (P(\omega_1) / P(\omega_2))$ . For maximum likelihood classifier,  $e$  in 6.7.23 is  $\log (|S_1|/|S_2|)$ . In these expressions  $P(\omega_i)$  is the a priori probability of class  $\omega_i$ , and  $|S_i|$  is the determinant of the covariance matrix  $S_i$  of class  $i$ ,  $i=1,2$ .

The classifier is easy to implement since to determine uniquely, only  $M_i$ ,  $S_i$  and  $P(\omega_i)$  have to be determined. If the fea-

tures are Gaussian distributed and  $S_1 \neq S_2$  then QDC gives a better classification accuracy than an LDC.

## 6.8 Classification Process

As described in Section 5.5, the unknown sample to be classified iterates from the root to the leaves in a hypothesis and test fashion. It is first assigned to one of the groups at the nodes in the first level and then proceeds to the next level. This procedure continues at each hierarchical layer of the tree and the classification becomes finer and finer while the sample goes higher and higher up to the leaves of the tree, until finally it arrives at one of the terminal nodes containing only one designated class. Using one of the most suitable pattern classification algorithms, the unknown input is sequentially classified according to the performance index among groups at each level of the tree. This is done under very strict range of decision parameters determined from the design set. The classification process may in fact terminate here, if the system is in the executive mode and the input pattern closely agrees with the characteristics of the reference pattern. In case the discrimination system fails the control is passed over to the Failure Control system where one final attempt is made to classify the unknown. If the Failure Control system is unable to identify the unknown, it may either be rejected or the control is handed over to the expert depending upon the mode of operation. The expert then can change a number of parameters in an attempt to classify the event.

The discrimination system includes two basic types of classifiers, parametric and nonparametric classification algorithms. Among parametric classifiers the Bayesian classifier (BYC), Linear discriminant classifier (LDC), and quadratic discriminant classifier (QDC) are included. Minimum distance classi-



fier (MDC) and K-nearest neighbor (KNN) were selected as non-parametric classifiers. For each of the five algorithms the system/user has the option of selecting three different decision (similarity measure) criteria. Based on the nature of the design data set or the user/expert choice, the system at every non-terminal node selects a suitable feature set and an appropriate classification algorithm while traversing (inferencing) through the knowledge tree.

### **6.9 Parametric Selection of a Classifier**

The choice of a classification procedure best suited to a specific problem is influenced by four factors of practical interest. First of these is the factor of imprecision of the design data where simply because of the testing conditions, the test equipment, or the test object itself, we are constrained to accept a given vector representation of the physical world. These issues have been addressed in Chapter 2.

The second factor is the number of dimensions of the vector space. Although it is within our power to choose whatever we think is useful information, the exact kind and number of different features of patterns we must describe in order to gain a complete characterization is beyond the capabilities of any machine. These questions, to the extent of practicality, are answered in Chapter 3. The IRS system provides wide choices of features that may be generated and does, for the purpose of discrimination between classes, sufficiently represent all classes. From the chosen features the system or the user can select the optimal features.

Once the dimensions of the pattern space are chosen, the third factor is to select a number of representative samples for each class. The exact number of samples is not significant, although different classification algorithms are influenced to

different extent by the sample size on which they operate. More accurate estimates of the required sample size can be made by statistical methods, here, however, only practical, "rule of thumb" is considered.

Generally speaking, the number of samples required by the decision procedures is related to the number of undetermined coefficients that must be established by the learning process. Linear discriminant classifier requires the smallest number of given samples, since only lower order statistics are needed to establish the decision rules. Decision rules that use various distance measures must estimate covariance matrices with a corresponding increase of the required sample size. From a mathematical point of view, a sample size several times the number of dimensions of the space is desirable. The number of samples on which the recognition is learned should exceed  $(n+pc_p)$ , where  $n$  is the number of dimensions and  $p$  is the degree of the polynomial (decision function).

After the selection of the vector space and the sample space, the fourth factor is to determine the method of classification to be employed. In the solution of practical physical problems this choice can be made relatively easily, for physical arguments can often be advanced in support of the adequacy of one procedure or another. This simplest solution is always the most desirable one, since it is usually the one that can be implemented most readily. Experimental evidence may indicate that members of a class lie close to one another and classes are well separated. If this is the case, LDC will, in general result in decision making with a sufficiently low error probability. If classes are believed to consist of subclasses, and groups are believed to possess different properties, it is best not to consider the employment of any of the linear methods. If the number of dimensions is low,

nonlinear methods are promising. If little or nothing is known about the data, the unsupervised or adaptive method, is particularly useful. With it, an intuitive notion of the nature of the probability density of a given class can be gained. This choice in most instances is difficult and can be made only intuitively by experts.

To select a classifier, theoretically one should seek the answer to these questions: 1) Does the method create optimal boundaries between classes? 2) How does it resolve overlapping boundaries? 3) How reliable and efficient the algorithm is? and, 4) Whether the effectiveness of the algorithm in regards to both complexity of the data and timeliness of the classifier is important? Whichever the classification rules one selects, it should produce as correct as possible classification decisions, and it should be easy to apply.

By answering these questions we attempt to distinguish between different methods. The general aim of supervised PR is to develop rules for classification of samples of unknown origin, on the basis of a design set with known classification which have been characterized by a number of features. The success of classification algorithm depends on whether the classification rules are optimal for the problem at hand. Optimal rules imply optimal class boundaries. If the variables used for the classification are appropriately chosen, then objects belonging to different classes are situated in separate regions of the pattern space and the classification rules correspond to boundaries of those regions. Optimal boundaries can be obtained only if the distribution of the population in pattern space is exactly known, and so is its parameters [DERD-86]. Theoretically, then, optimal boundaries can be obtained only if each class is represented by an infinite number of samples. In practice, population distribution and its parameters are estimated from the samples of restricted

size. It is imperative to remind the rules of sampling theory in order to draw representative samples. Thus, one distinction between techniques can be made based on the information on their underlying population distribution. The non-parametric techniques make no assumptions on the population distribution while parametric techniques do. The parametric techniques are based on a well defined distribution. LDC and QDC, for instance, are based on the assumption that the population distributions are multivariate normally distributed. Consequently they yield optimal boundaries only if the populations are indeed normally distributed. The efficiency of parametric techniques is greater than that of non-parametric techniques, especially when small samples are used. If no such information is available then non-parametric techniques should be used. Deviations from these assumptions about the distribution may result in boundaries far from optimal.

Often closely similar classes create overlapping regions and it usually becomes impossible to find a combination of parameters that allow complete distribution, in such situations piecewise LDC or QDC may be useful.

A third possible distinction between techniques can be developed on the basis of their degree of reliability. Algorithms such as KNN and MDC can be considered deterministic in character whereas modeling and BYC methods are probabilistic methods. With a deterministic technique an object is classified in one and only one of the training classes and the degree of reliability of this decision is not measured. In probabilistic techniques, the boundaries of the classes correspond to confidence limits defined on a statistical basis.

The distinction based on effectiveness is concerned with the timeliness of the decision. In certain situations the time

constraints are so absolute that even a correct classification is of no use after a given time. For example, in a vision system the robot must see (recognize) the object before it moves its hands to pick up the object. Similarly, in medical monitoring systems the signals received from a terminally ill patient must immediately be recognized to save his/her life.

The theoretical aspects pertaining to the selection of a classifier can then be summarized in Table 6.2. Referring to Section 3.4.4, similar to clusters, we considered the pattern classes that are optimally compact in the sense of minimum intra-class variations as homostats and the ones that are not compact, as segregates. All the thoughts described above can be structured into a set of meta rules shown in Table 6.3. These rules, based on the characteristics of the data, would guide the expert/user in selecting an appropriate classification procedure.

Table 6.2  
Properties of Classification Algorithms

Method	Advantages	Disadvantages
MDC	Distribution free, Simple and fast, work well if classes are homostat.	Small amount of noise can significantly lower the performance.
KNN	Distribution free, Only assumption is high correlations between nearby features. No training necessary.	Limited if classification speed is important - though could be modified.
LDC	Parameters can be inter- preted in terms of main effect and interactions.	Small/large n Performance drops if classes are segregates.
QDC	No particular assumptions.	Small n or very large n. Computationally expensive.
BYC	Only n parameters/class. Good performance if dist- ribution is normal.	Assumes independent vari- ables. Usual assumptions of parametric methods. Performance drops if data deviates from assumption.
PAH-U*	Easy to implement and carries the advantages of the classifier used.	Carries the disadvantages of the nodal classifier used.
PAH-V+	More reliable and robust.	Difficult to implement and carries the multiple of disadvantages pertain- ing to every classifier used.
-----		
* PAH Classifier with same classifier at each node		
+ PAH Classifier with different classifiers at each node		

Table 6.3

Set of rules used in the selection of Classification  
Procedure and Feature Weighing Criterion

- 
1. Is the number of pattern classes known?  
[2. Yes, 3. No]
  2. Are the pattern class distributions known?  
[4. Yes, 5. No]
  3. Classes are unknown; expert input is required.  
3.1 Unsupervised learning is required.  
/\* The system does not support this situation as yet,  
Exit. \*/
  4. Use Parametric Classifier.  
/\* To identify a suitable parametric classifier, first  
decide whether classes are homostats or segregates by  
calling the procedure Structure. \*/  
  
**Structure**  
  
4.1 Enter the pattern classes in order.  
  
4.2 Enter the number of samples in each class in the same  
order.  
  
4.3 /\* processing by the system \*/  
  
Arbitrarily pick 20% (minimum 2 classes) of pattern  
classes and the system will read all samples belonging to  
the pattern classes chosen, compute the mean and  
variance.  
  
intra-class variation: \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_  
threshold: \_\_\_\_\_  
  
structure end  
  
Is the intra-class variation  $\geq$  threshold?  
[6. Yes, 7. No]
  5. Can you estimate the distribution parameters?  
[8. Yes, 9. No]

contd.

---

Table 6.3 (Contd.)

Set of rules used in the selection of Classification  
Procedure and Feature Weighing Criterion

---

6. Sample patterns in a class are segregates.
- 6.1 /\* This step is determined automatically by evaluating the inter-class variations. If the inter-class variations are small; select either the QDC or the BYC method. \*/
- Are the variations in the feature values significant?  
[10: Yes, 11. No]
7. Sample patterns in a class may be homostats.
- 7.1 /\* If the intra-class variations are < threshold, select LDC. The system would show the inter-class variations for the selected classes \*/
- Can you call the pattern a homostat?  
[12. Yes, 10. No]
8. Go to Step 4
9. Use Non-parametric Methods.  
/\* To choose an appropriate method one basically has to decide whether the pattern classes are homostats or segregates \*/
- call procedure **Structure**
- Is the intra-class variation  $\geq$  threshold?  
[13. Yes, 14. No]
10. Select the QDC algorithm.  
Go to Rule 15
11. Select the BYC algorithm.  
/\* Apply the algorithm and exit \*/
12. Select the LDC algorithm.  
Go to Rule 15
13. Select the KNN algorithm.  
Go to Rule 15

contd.

---



Table 6.3 (Contd.)

Set of rules used in the selection of Classification  
Procedure and Feature Weighing Criterion

- 
14. Select the MDC algorithm.  
Go to Rule 15
15. /\* Classification algorithm has been selected, now select  
the feature weighing criterion. \*/
- Determine the appropriate weights of the features.  
Do you want the variables to be weighted?  
[16 Yes, 17. No]
16. Select the appropriate weight of features.
- 16.1  $w_i = 1$  /\* equal weight to all features \*/
- 16.2  $w_i = 1/\sigma_i$  /\* if features have smaller intra-class  
variations \*/
- 16.3  $w_i = s_i/|m_i|$  /\* Hsia weight [HSIA-81] \*/
- 16.4  $w_i = \left\{ 1/[s_i \sum_{j=1}^N 1/s_{ij}] \right\}^2$  /\* Tou and Gonzalez weight  
features \*/
- 16.5  $w_i = s_i/\sigma_i$  /\* if features have smaller intra-class  
variations \*/
- 16.6  $w_{ijk} = \{m_{i,j} - mk_{k,j}\}^2 / \{p_i s_{i,j}^2 + p_k s_{i,k}^2\}$   
/\* if features have smaller intra-class \*/
- 16.7  $w_i = 2 \left\{ \sum_{j=1}^{N-1} \sum_{k=1}^N w_{ij} / N(N-1) \right\}$  /\* Tou and Gonzalez  
features \*/
- Go to Rule 18
17. Select the Classification Algorithm specified.  
/\* Apply the algorithm to classify the unknown and Exit.  
\*/
18. Select the algorithm specified in the descendent rule  
along with the weight function.  
/\* Apply the algorithm using the selected weight,  
classify the unknown and Exit. \*/
-

## Chapter 7

### Classification Experiments and Results

#### 7.1 Introduction

The recognition methods we developed were put to practical use to evaluate their performance on real-life data sets. The data set signals were taken from non-destructive testing (NDT) and non-invasive testing (NIT) generated from known/unknown materials. These data sets will be referred to as NDT-data, EEG signals or EEG-data, genetic cell data or CEL-data, and petroleum oils (polynuclear aromatic hydrocarbons) data or PNA-data. The algorithms constituting the system discussed in previous chapters were implemented and several classification experiments were conducted.

The experiments reported in this chapter include: 1) Classification of NDT data set with full (unabridged) and abridged feature sets using the single layer classifiers, 2) Classification of NDT data with optimal feature sets using single layer classifiers, 3) Classification of NDT data with an overall (global) optimal feature set and the same classifier at each node of the PAH tree, 4) Classification of NDT data with locally optimal feature sets and the same classifier at each node of the PAH tree, 5) Classification of NDT data with different nodal classifiers at each node of the PAH tree - PAH-V classifier, 6) Repeating experiments conducted in 1 to 5 on EEG data (3 class problem), 7) Repeating some of these experiments on PNA data (20 class problem), and, 8) Repeating several of the experiments on CEL data (19 class problem).

Before we report any experiment, we recap the structure of the methods so that the reader can visualize the complete picture without flipping through the previous chapters.

## 7.2 The Functional View of the Recognition Components

To demonstrate the feasibility of the concepts and the performance of various algorithms presented in previous chapters a working model of the components described was developed. As described in Chapter 2 these components are composed of two major subsystems, 1) Knowledge Acquisition, Representation, and Organization (KARO) subsystem, and, 2) Inference Engine. The building blocks and operational details of these components are presented in previous chapters. We present here a functional and operational overview of the system.

The KARO subsystem consists of three phases: namely, fact gathering, knowledge base, and knowledge formalization and organization. The fact gathering phase performs three main tasks, 1) acquisition of input data (waveform signals), 2) data preprocessing, and, 3) pattern measurements. Each of these tasks further entails a series of operations. The pattern measurements which have been referred to as analytical features (or knowledge), constitute a major component of the knowledge base. Other components of the knowledge base include empirical knowledge and the meta knowledge regarding the requisite problem domain. Meta knowledge comprises of, 1) a set of rules constructed to represent statistical knowledge present in a data set, 2) a set of procedures that an expert may apply on physical observations, and 3) a set of empirical parameters to monitor the classification process. The meta knowledge is used to formulate an expert's judgement and objectives.

The next phase of the KARO is knowledge formalization and organization. The knowledge was organized using a new concept of 'pattern association hierarchy (PAH)' developed in this thesis. Analytical knowledge formalized as feature vectors, takes advantage of the natural association that exists among

pattern classes to build their association hierarchy and hence the concept of PAH was introduced. The empirical and meta knowledge pertaining to each set of associated pattern classes is organized in structures called knowledge frames. Each knowledge frame comprises procedures and a list of decision parameters described in Section 4.6.

A knowledge frame with appropriate node-dependent knowledge is placed at each intermediate node of the PAH. The same pattern association hierarchy is used by the Discrimination subsystem (a component of the inference engine) to classify patterns (see Section 3.5). The classification process is a two-tier system and includes the Discrimination System and Cognition System, which are described below.

#### **7.2.1. The Function of The Discrimination Subsystem**

The discrimination subsystem is a procedure-based pattern classification system and uses several decision-theoretic classification algorithms. These algorithms include two basic types of pattern classifiers, parametric and non-parametric. Among parametric classifiers, linear discriminant classifier (LDC), quadratic discriminant classifier (QDC), and Bayesian classifier (BYC) are developed, whereas minimum distance classifier (MDC) and K-nearest neighbor (KNN) were selected as non-parametric classifiers. Each of these classifiers again can select one among several decision criteria available, any size of feature subset, and a weighing function among various provided. Based on the nature of the design data set or the user/expert choices the system at every non-terminal node can select an appropriate classification algorithm while traversing (inferencing) through the knowledge tree.

Each of the classification algorithm can function on a general PR problem with 2 to 21 pattern classes, 2 to 100 samples in

a pattern class, and each class may have a variable number of samples, and, 1 to 112 features in a pattern vector. Note that these parametric values are merely the practical limits, the algorithm itself does not care for such values. These algorithms are tested on various data sets using abridged feature set (Feat-A), i.e., feature set selected by Successive Feature Elimination Process, the Fisher ranked feature set (Feat-F), and the features ranked by the pseudo-similarity method, i.e., feature set Feat-S. The feature sets Feat-F and Feat-S are selected after eliminating the poor performers. Feat-F and Feat-S are sets of ranked features from which any number of features can be selected.

### The Discrimination Process

The decision process uses two types of classifiers - single layer and hierarchical. A single layer classifier uses a common feature set and evaluates a decision function through all pattern classes simultaneously and selects the one giving the highest value (discrimination). This is a well understood phenomenon and questions about its merits have been raised in the literature [FUKU-90] and at several places in this thesis, it will not be discussed any further. However, to demonstrate some practical insights several results using this kind of search strategy are described in the sections to follow.

The PAH classifier is a hierarchical classifier. This, in fact, is an inference tree and can, 1). use the same classifier at each internal node of the tree, or 2). different classifiers at different non-terminal nodes of the tree. Thus to distinguish N classes N-1 different decision functions or classifiers can be used. In the first case it will be called PAH-U -- stands for PAH-Uniform, and for the latter case it will be called PAH-V -- stands for PAH-Variable. Using any of these classifiers, the unknown sample to be classified

iterates from the root to the leaves in a hypothesis and test fashion. It is first assigned to one of the groups at the first level, each of which contains several classes and then to the next group at one of its child nodes in the second level. This procedure continues at each hierarchical layer of the tree and the classification becomes finer and finer as the sample goes higher and higher up (towards the leaves) the tree, until finally it arrives at one of the terminal nodes containing only one original class whose label is assigned to the unknown.

### **7.2.2 The Function of The Cognition Subsystem**

The cognition system implements an information-theoretic algorithm for pattern classification. It is an alternate classifier and performs classification independent of other classification algorithms. It uses the transformed analytical features (pseudo-symbolic features). This algorithm, called entropy-based decision tree (EDT) algorithm learns its classification capability by selecting one best feature at a time based on its (feature) information content. The feature selected is used to split the available patterns into two groups. The process is continued hierarchically until all samples at a given node belong to the same pattern class. At this point such node becomes a terminal node and the label of the pattern class is assigned to this node and a search (classification) path is established for the pattern class. The feature selection, hierarchical splitting and class labelling process is continued until the entire design set is assigned to some terminal node. Note that such decision tree organization eliminates the need for any classification algorithm.

The tree so constructed is saved and later used for classification. To classify, the unknown entering from the root, iterates through the tree and is assigned the label of a

terminal node based on maximally matching its information content.

### **7.2.3 The Function of The Failure Control Subsystem**

The failure control mechanism which also uses the decision theoretic algorithms of the PAH-U and PAH-V classifiers is an additional pattern classification system which has been developed to handle the failures of the PAH-U and PAH-V classifiers. Both of these algorithms may fail at any intermediate node of the hierarchy. In such case the Failure control system attempts to classify the unknown pattern one more time using the same parent classifier (in their traditional perspective) but against all classes above the current PAH-node. The Failure control system learns its decision-making capability through the scheme described earlier (see Chapter 5) and uses an overall optimal feature set determined by the Fisher discriminant function of equation 4.4.1 to perform the classification.

### **7.3 System's Training**

For every classification problem all the classifiers were available at each non-terminal node of the PAH. However, the classifier selected by the rules was tagged for a particular node and the selected classifier was individually trained for the classes already assigned (by the tree building process) to that node, so that the system while classifying an unknown in the expert mode simply selects the tagged classifier. Available to each node of the hierarchy is its optimal feature set and other empirical knowledge components. The classifier and the training information for each intermediate node is then stored in the respective knowledge frames. The system can now be used to characterize any set of input signals (in the respective problem domain, of course). The following sections

summarize the results obtained from the different classification experiments.

#### **7.4 Performance of the Recognition Components**

The recognition and interpretation components were individually trained on four different data sets using several classification algorithms. The implementation details of these algorithms are described in following sections. With a few exceptions, the performance of the system on the training sets from all four data sets was very impressive, ranging from 82.5% to 100% for a majority of pattern classes from various algorithms. Hence these results are reported here only if we found it necessary. However, the results on the testing sets will be discussed in detail since they are indicative of performance potentials of the methods in a realistic environment. The performances of individual feature selection schemes and classifiers are observed and several interesting results are reported in sections to follow.

##### **7.4.1 Implementation of MDC**

The Minimum Distance Classifier (MDC) was the first classifier we implemented using two different decision criteria - Euclidean distance and Mahalanobis distance. The MDC classifier with two decision criteria will be referred to as MDC-E and MDC-M in the following discussion implying the MDC classifier using Euclidean distance and the classifier using Mahalanobis distance, respectively. Each of these classifiers can be trained individually to function as a single-layer stand-alone classifier. In addition, they are designed to function with any given number of samples, feature sets and decision criteria.



These classifiers can also be used on one or more non-terminal nodes of the PAH. A set of unique distance thresholds are evaluated from the design set and placed at each decision node of the tree. These thresholds were used to implement the reject option on the PAH to eliminate further classification process.

#### 7.4.2 Implementation of KNN

The K-nearest neighbor (see Section 6.7.2) was implemented for  $k=1, 3, 5$ . Euclidean distance with various weights, and Mahalanobis distance were used to evaluate the neighborhood criterion and the KNN classifier using each of the distances will be referred to as KNN-E and KNN-M in the following discussion. To improve the performance, k-nearest samples to the unknown from each class were selected from which the final k neighbors were examined. Since the Mahalanobis distance involves heavy computations, we evaluated the distance from four arbitrary samples and the mean of the concerned class only.

#### 7.4.3 Implementation of LDC

The linear discriminant classifier with discriminant function given in equation 6.7.19 was implemented. The coefficients of the discriminant function were defined as:

$$c_k = S^{-1} M_k \quad \dots 7.4.1$$

where  $M_k$  is the mean vector of the k-th group (class) and S is the pooled variance-covariance matrix of the groups (classes). To account for lack of equality in the covariance matrices each component of the function can be weighed using one of the weighing functions described in Section 4.5.

As reported in Chapter 6 and for several other reasons discussed in previous sections this classifier performs well on homostat data only and one should not apply this scheme as soon as the within-class variations exceed certain data-dependent threshold.

#### **7.4.4 Implementation of QDC**

In fact, the computer program for Quadratic Discriminant Classifier (QDC) was implemented by enhancing the program for LDC. The program for QDC computes the discriminant differently; instead of computing the pooled variance/covariance matrix, individual group (class) covariance matrices were used to compute the coefficients of the discriminant function. To account for non-normality in data, we included the provision to multiply each discriminant function by one of the several weights described earlier.

The quadratic and linear discriminant classifiers are useful for a wide range of distributions. LDC performs as well as QDC unless there is a great difference in the covariance matrices of different classes.

#### **7.4.5 Implementation of BYC**

The Bayesian classifier (BYC) was implemented using the algorithm described in Section 6.7.1. As mentioned in that section, the Bayes rule is optimal if minimum overall error of classification is required. To achieve this objective we carefully examined the feature variations both within classes and between classes and estimated the posteriori class probabilities using various ranges of features describing each pattern class. For this purpose one needs a larger feature set and a larger sample size. Since we did not have much choice on the size of data, we decided to use a larger feature

set. As such we used 62 features for the NDT data, i.e., the features obtained after processing through the first two steps of the Successive Elimination Process. For EEG data we had a sufficient size of the data and as such the abridged set was used to estimate the posteriori probabilities for this problem.

#### 7.4.6 Implementation of PAH

Various algorithms comprising the PAH classifier have been described in Section 6.6. We implemented this classifier using a composite of two procedures consisting of all those algorithms. One procedure reads the appropriate tree constructed by one of the selected clustering procedures, and computes and stores the empirical and statistical knowledge pertaining to cluster of classes for every non-terminal node of the tree in the form of an indexed sequential storage. This knowledge is utilized by the classifier (tagged one, or the one selected by the user) to recognize an unknown. The other procedure is the driver which implements the two modes of system's operations, i.e., executive (expert), and consultant (assistant) modes. The driver, in either of its modes lets a pattern (known or unknown) run through the tree, retrieving the appropriate knowledge and applying the user-indicated nodal classifier or the tagged one to identify the pattern in question. The decision of the classifier could either be in favor of one of two groups (classes), or rejecting both. In case of a favorable decision the classification will continue to the subsequent node until a terminal node is reached, whose identity becomes that of the unknown. In case of rejection the Failure Control process is invoked whose task is to make one more attempt to classify the rejected pattern using the selected classifier and all the classes above the current node.

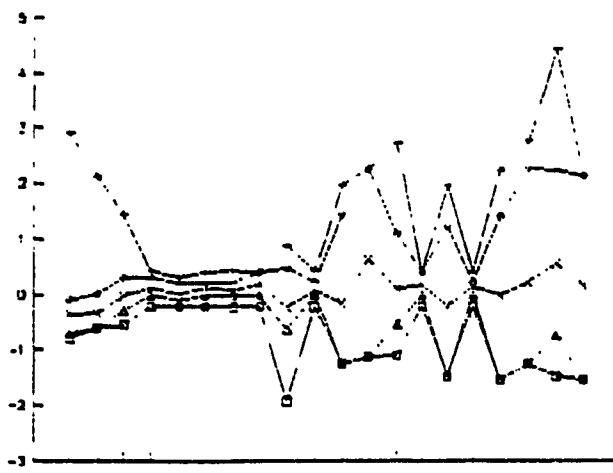
The classifier was implemented in two ways: PAH-U and PAH-V. In PAH-U, 'U' stands for uniformity, implying that the same classifier is used at each non-terminal node of the tree. If different classifiers are used at various nodes, the classifier is called as PAH-V, i.e., PAH Variable.

#### 7.4.7 Implementation of EDT

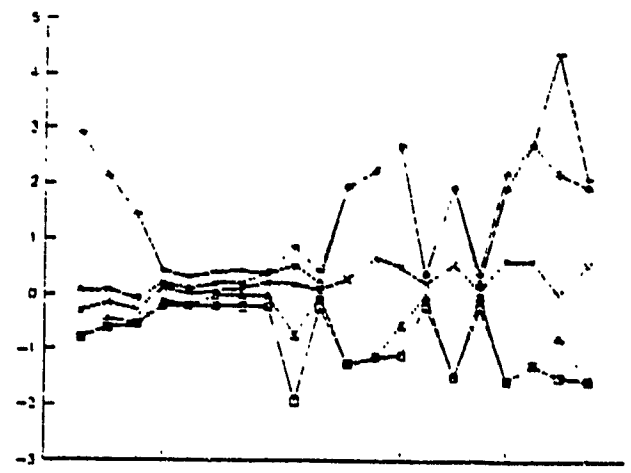
Because of the size of the decision tree the Entropy-based Discrimination Tree (EDT) algorithm (see Section 5.7) was implemented on an AT&T 3B2 minicomputer. The algorithm selects one feature at a time based on its capability to split the entire design set into two groups. A specific range of a feature is considered most capable if it has the maximum entropy at the current node. The selected feature is assigned to the node. At the next step a feature that brings maximum reduction between entropies of an input group and a corresponding output group is selected. The process is continued until each path emanating from the root of the tree ends up with zero entropy. In that case the node becomes a terminal node, the label of the pattern class is assigned to it and the process terminates.

#### 7.5 Performance on NDT Data

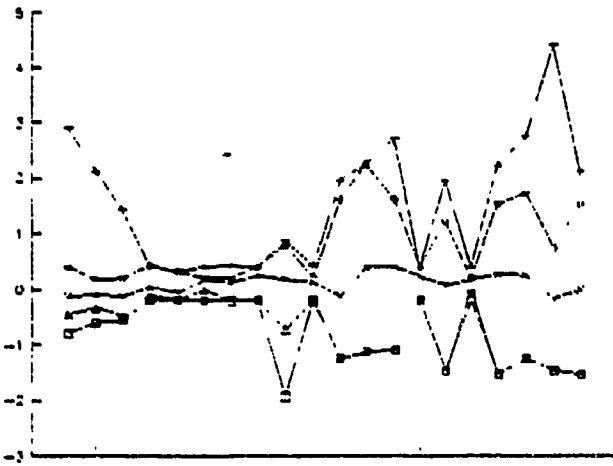
Forty samples of NDT data were collected from acousto-ultrasonic tests of each of 9 artificially induced defects in a steel bar. The same number of samples were also collected from a flawless bar. The characteristics of the data are already described in Section 3.2.1 and the nomenclature of the 10 pattern classes is listed in Table 3.1. Since NDT data was the most noisy data and had all sorts of problems (see Fig. 7.1) a PR designer can imagine, it was used as a test case for developing all the components of the system. In addition, we had most of the required information available including the



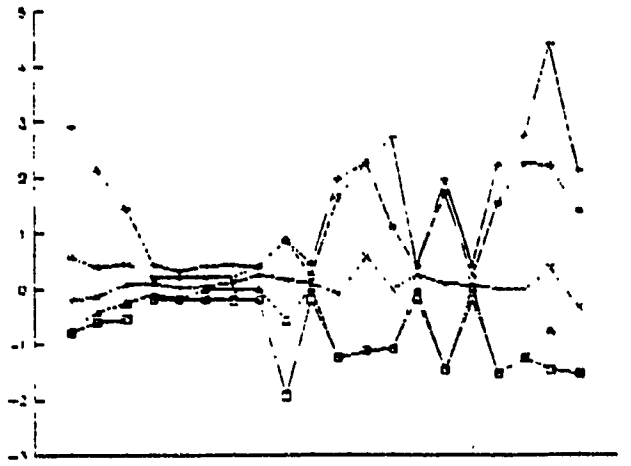
a: Variations in class A



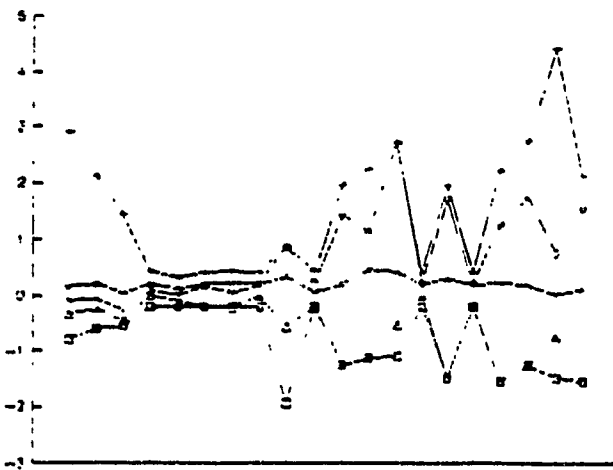
b: Variations in class B



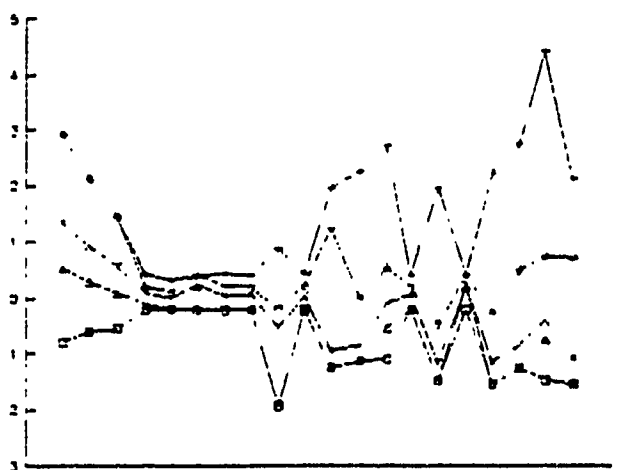
c: Variations in class C



d: Variations in class D



e: Variations in class E



f: Variations in class F

Legend: Horizontal Scale: Features  
Vertical Scale: Feature Value in Thousand

Fig. 7.1: Variations between different NDT Pattern Classes

raw signals. The complexity of this problem can be examined by the amount of variation within different classes (see Fig. 7.1). The amount complexity of this problem can be examined by the amount of variation within different classes (see Fig. 7.1). The amount of variation within different patterns of the same class was in the order of  $10^{+5}$ . We stabilized the variations by using the log transformation (see Section 3.2.2.2). The results obtained during different intermediary steps, such as feature nomenclature, features selected at different phases of "Successive Elimination Process", and various pattern association hierarchies, i.e., PAH's constructed, have been reported in previous chapters. The pattern classification experiments conducted on this data are reported in this section.

#### 7.5.1 Experiment - A: MDC

Initially, we used the unabridged feature set (Feat-U). The performance of the classifier was very low, recognizing only 0% to 60% of samples for various classes giving an impression that features might have been just a bunch of random numbers. Upon using the abridged feature set Feat-A and ranked feature sets Feat-F and Feat-S the performance was improved significantly.

Using an overall best feature set (abridged) of 40 features, Feat-A, that is the features obtained by applying the Successive Elimination Process, the classifiers MDC-E and MDC-M were applied to all 10 classes simultaneously, the results of which are shown in Table 7.5.A1, and Table 7.5.A2 respectively. These results were only slightly better giving a peak performance of only 82.5% correct recognition with MDC-M classifier. Major failure was encountered on the recognition of 'small' defect classes, particularly, small-medium and small-deep classes. This observation can be interpreted as, 'small

defects far inside the material are difficult to identify', and for all practical purposes their misrecognition or rejection may not be severe to cause any catastrophic loss. However, (see Table 7.5.A2) only two defective patterns were misclassified as non-defective and the no-defect class was itself 90% correct which implies that the system is capable of successfully discriminating, at least, between defects and non-defects.

The above experiments were repeated using optimal feature sets, Feat-F, and Feat-S, the results of which are shown in Table 7.5.A3, and Table 7.5.A4. It appears that Fisher ranked features were more effective than Pseudo-similarity ranked features giving an overall performance of 70%. However, in case of Feat-F we have to weigh each feature with a weight of  $1/s_i$ , i.e., scale the feature with the standard deviation of feature  $i$  for the class comparing with, whereas the Feat-S was weighed using  $1/s_o$ , i.e., scaling the feature with the overall standard deviation of feature  $i$ . The first weight is computationally more expensive to evaluate than the latter one. Another interesting observation to note is that the performance of both feature sets Feat-F and Feat-S was comparable for medium, large and no-defect classes, however, Feat-F was more sensitive than Feat-S in recognizing 'small defect' classes.

A similar experiment was conducted using MDC-M classifier and even better results were obtained (see Table 7.5.A5) suggesting to use Mahalanobis distance instead of weighted Euclidean for cases where the pattern classes are 'segregates', i.e., have large within class variations.

Table 7.5.A1

Classification Results on NDT Data using  
 Linear Organization of Pattern Classes  
 (MDC - Euclidean Distance)  
 (Feature Set: Feat-A = 40,  $wt=1/sd_0$ )

Class output input	1	2	3	4	5	6	7	8	9	10	Recog. %
1	12		1	4			3				60
2		8	4	3	3			2			40
3	1	2	10	1				3	3		50
4	1	2	2	10		1	3	1			50
5		3	1	1	11		1		3		55
6						19			1		95
7	1		1	2			15	1			75
8				3	2		1	13		1	65
9				1	1				18		90
10	1		1	1			1			16	80
Total Recog./No. Misrecog./ Av. Recog.							132	68			66

Table 7.5.A2

Classification Results on NDT Data using  
 Linear Organization of Pattern Classes  
 (MDC - Mahalanobis Distance)  
 (Feature Set: Feat-A = 40)

Class output input	1	2	3	4	5	6	7	8	9	10	Recog. %
1	18									2	90
2		15	2	1	1			1			75
3	1	1	12	2	1	1		2			60
4	3	1	2	13		1					65
5		2	1	1	16						80
6						20					100
7	1	1					18				90
8		1		1			1	17			85
9		1					1		18		90
10	2									18	90
Total Recog./No. Misrecog./ Av. Recog.							165	35			82.5



Table 7.5.A3

Classification Results on NDT Data using  
 Linear Organization of Pattern Classes  
 (MDC - Euclidean Distance)  
 (Feature Set: Feat-F = 20,  $w=1/sd_i$ )

Class output input	1	2	3	4	5	6	7	8	9	10	Recog. %
1	14		1	4						1	70
2		10	3	1	4			2			50
3	1	1	10	2	1		1	2	2		50
4	2	2	2	8		1	2	1	2		40
5		2	2	2	11			1	2		55
6						20					100
7		1		1			18				90
8			1	2	1		1	14		1	70
9				1	1				18		90
10	1		1				1			17	85
Total Recog./No. Misrecog./ Av. Recog.								140	60		70

Table 7.5.A4

Classification Results on NDT Data using  
 Linear Organization of Pattern Classes  
 (MDC - Euclidean Distance)  
 (Feature Set: Feat-S = 25,  $w=1/sd_o$ )

Class output input	1	2	3	4	5	6	7	8	9	10	Recog. %
1	10	1		2	3	1	3				50
2	1	11	1	1	4			1	1		55
3	3	2	5	1	1	1		4	3		25
4	4	2		7		1	4	1	1		35
5	1	4			10	2	2	1			50
6						20					100
7					1		18		1		90
8		1	1	1			1	14	1	1	70
9		1				1			18		90
10			1	1			1	1		16	80
Total Recog./No. Misrecog./ Av. Recog.								129	71		64.5

Table 7.5.A5

Classification Results on NDT Data using  
Linear Organization of Pattern Classes  
(MDC - Mahalanobis Distance)  
(Feature Set: Feat-F = 20)

Class output input	1	2	3	4	5	6	7	8	9	10	Recog. %
1	17	2		1							85
2	3	14					2	1			70
3		3	15							2	75
4				17		1	1		1		85
5				6	8	3	2		1		40
6				7	2	11					55
7	1	2		3		1	13				65
8	4	1					2	13			65
9									20		100
10		2								18	90
Total Recog./No. Misrecog./ Av. Recog.							146	54			73

### 7.5.2 Experiment - B: KNN

Using KNN the performance of the classifier on the design sets for NDT data was in the range of 40% to 60% for various feature sets. Table 7.5.B1 shows the performance of KNN-E for Feat-A=40. We reported the number of rejects for each class as well. In some cases more than 50% of samples in a class were rejected. This observation can be attributed to large variations between patterns of those classes but also shows the reliability of the classifier. Similar experiment was repeated using Mahalanobis distance to determine the neighborhood, the results of which are shown in Table 7.5.B2. The results reported in this table were much better than those reported in Table 7.5.B1. This observation shows that, at the expense of heavy computations, satisfactory results may be obtained. Theoretically, since KNN classifiers are useful for homostat type data, it was understandable to obtain such a poor performance from the classifier. Hence, we did not

perform any additional experiments with this classifier on NDT-data.

Table 7.5.B1

Classification Results on NDT Data using  
Linear Organization of Pattern Classes  
(3NN - Euclidean)  
(Feature Set: Feat-A = 40,  $wt=1/sd_i$ )

Class input	output 1	2	3	4	5	6	7	8	9	10	Rej.	Recog. %
1	10	2		1							7	76.92
2		13			1	1					5	86.67
3	1	1	2	1					1		14	33.33
4				4	3	1				1	11	44.44
5		1	1	1	8	1		1	1		6	57.14
6						15	5				0	75.00
7		1		1	1		5	3			9	45.45
8			1		1			5	4		9	45.45
9			1			2			6	7	4	37.50
10	1	1		1	1			3	1	8	4	50.00
Total Recog./No. Misrecog./No. Reject/Av. Recog.											76 55 69	58.02

Table 7.5.B2

Classification Results on NDT Data using  
Linear Organization of Pattern Classes  
(3NN - Mahalanobis Distance)  
(Feature Set: Feat-A = 40)

Class input	output 1	2	3	4	5	6	7	8	9	10	Rej.	Recog. %
1	11	1	1	1						1	5	73.33
2		15	1								4	93.75
3	1		5	1		1					12	62.50
4				7	1	1				1	10	70.00
5		1		1	10	1		1	1		5	66.67
6						20					0	100.00
7					1		10	1	1		7	76.92
8					1			8	2		9	72.73
9									12	5	3	70.59
10	1			1				1	1	8	8	66.67
Total Recog./No. Misrecog./No. Reject/Av. Recog.											106 31 63	77.37

### 7.5.3 Experiment - C: LDC

The performance of linear discriminant classifier (LDC) on NDT data was examined and as expected the results were very poor and were not worth reporting here. As explained in Sections 7.4.3 and 7.4.4 and as the rules described in Table 6.3 dictate it, this classifier performs well on homostat data only and hence no further experiments on NDT data were conducted using this classifier.

### 7.5.4 Experiment - D: QDC

The performance of the quadratic discriminant classifier (QDC) on NDT data was also examined and the results were in the range of 71% to 84% for various feature sets. As explained in Section 7.4.4 and also supported by the rules described in Table 6.3, this classifier performs well on "segregates" data since the classifier uses the individual within class co-variances to determine the likelihood. Table 7.5.D1 shows the performance of QDC for feature set Feat-A=40. The results shown vary between 55% and 90% giving an overall average of 71%. Similar experiment was repeated using the weight of  $1/sd_i$  to determine the best proximity, the results of which are shown in Table 7.5.D2. The results reported in this table were comparatively better than those reported in Table 7.5.D1 and an average performance of 74% was achieved. This observation shows that the variations between and within classes may be inconsistent, however, at the expense of slightly added computations satisfactory results may be obtained. The last experiment was repeated using optimal feature sets Feat-F and Feat-S, giving an average performance of 84% and 81.5%, respectively. The results are reported in Table 7.5.D3 and Table 7.5.D4. These results show that comparatively better results can be obtained if best features are used, however, the features ranked by Fisher index appear to give slightly

better results than the one ranked by the pseudo-similarity algorithm. The reason for the difference in the performance is due to the way two algorithms rank the features. Since the NDT data had large amounts of variations the Fisher's ranking performed slightly better as it uses both within and between class variations in ranking the features. By reviewing the confusion matrix (Table 7.5.D3) another observation we made is that Fisher index clustered the classes around a narrow diagonal band, implying that a majority of the misrecognized samples were confused with their close neighbors. Theoretically, since QDC classifiers are useful for "segregates" type data, it was understandable to obtain better performance from this classifier as compared with that obtained from MDC-E, KNN and LDC classifiers.

Table 7.5.D1

Classification Results on NDT Data using  
Linear Organization of Pattern Classes  
(Classifier: QDC)  
(Feature Set: Feat-A = 40)

Class output	1	2	3	4	5	6	7	8	9	10	Recog. %
input											
1	12	3	1	1		2				1	60.00
2	2	15	1	2							75.00
3	1	2	13	3						1	65.00
4		1	2	11	4	1				1	55.00
5		1		1	13	1		1	1	2	65.00
6					1	14	5				70.00
7						2	15	3			75.00
8					1	1	2	16			80.00
9							2	3	15		75.00
10	1								1	18	90.00
Total Recog./No. Misrecog./ Av. Recog.								142	58		71.00

Table 7.5.D2

Classification Results on NDT Data using  
 Linear Organization of Pattern Classes  
 (Classifier: QDC)  
 (Feature Set: Feat-A = 40,  $wt=1/sd_i$ )

Class output input	1	2	3	4	5	6	7	8	9	10	Recog. %
1	12	2	2	1		2				1	60.00
2	2	16	1	1							80.00
3	1	2	14	3							70.00
4		1	2	13	3	1					65.00
5		1		2	12	2		1	1	1	60.00
6					1	15	4				75.00
7						3	15	2			75.00
8					1		2	17			85.00
9							2	2	16		80.00
10	1								1	18	90.00
Total Recog./No. Misrecog./ Av. Recog.											148 52 74.00

Table 7.5.D3

Classification Results on NDT Data using  
 Linear Organization of Pattern Classes  
 (Classifier: QDC)  
 (Feature Set: Feat-F = 20,  $wt=1/sd_i$ )

Class output input	1	2	3	4	5	6	7	8	9	10	Recog. %
1	13	3	2			1				1	65.00
2	2	16	2								80.00
3		3	15	2							75.00
4			2	16	2						80.00
5			1	2	15	2					75.00
6					2	17	1				85.00
7							18	2			90.00
8							1	19			95.00
9								1	19		95.00
10										20	100.00
Total Recog./No. Misrecog./ Av. Recog.											168 32 84.00

Table 7.5.D4

Classification Results on NDT Data using  
 Linear Organization of Pattern Classes  
 (Classifier: QDC)  
 (Feature Set: Feat-S = 20, wt=1/sd<sub>i</sub>)

Class input	output 1	2	3	4	5	6	7	8	9	10	Recog. %
1	13	3	3	1							65.00
2	1	16	2	1							80.00
3	1	3	15	1							75.00
4	1			15	3	1					75.00
5		1		2	14	2		1			70.00
6					1	17	2				85.00
7							19	1			95.00
8							2	18			90.00
9							1	2	17		85.00
10	1									19	95.00
Total Recog./No. Misrecog./ Av. Recog.											163 37 81.50

#### 7.5.5 Experiment - E: BYC

The Bayesian classifier was also applied to all 10 classes simultaneously and much better performance with an overall average of 78% correct was obtained. These results as compared to the two earlier experiments (Tables 7.5.A1, 7.5.A2) are much better and can be attributed to more accurate evaluation of the class probabilities. These results are reported in Table 7.5.E1. Again the BYC was able to distinguish defects from non-defects 100% of the times, whereas only one non-defect out of 20 samples was recognized as defect, thus correctly recognizing non-defects at 95%. In fact had the data been of a larger size, the performance ratio might have been better. The large defects were far better recognizable than small defects. It is suspected that the physical observations from smaller defect classes may not be representative. This observation may be attributed to the poor quality of the data acquisition system wherein the quality of the transducer/

receiver system may be questioned. The probing system may not be powerful enough to transfer certain meager information pertaining to smaller defects.

Table 7.5.E1

Classification Results on NDT Data using Linear  
Organization of Pattern Classes  
(Bayesian Classifier, Feature Set: Feat-A = 62)

Class Output Input	1	2	3	4	5	6	7	8	9	10	Recog. %
1	15			1		4					75
2	2	13	1	2				1	1		65
3	1	2	14	1				1			70
4		3		11		4			2		55
5	1			1	13	4		1	1		65
6						18	1	1			90
7							17	3			85
8				1				18	1		90
9				1		1			18		90
10				1						19	95
<hr/>											
Total Recog./No. Misrecog./ Av. Recog.							156	44			78

#### 7.5.6 Experiment - F: PAH

The MDC classifier was applied on a PAH tree constructed by different clustering algorithms. Using the single linkage algorithm for building the tree and MDC-M classifier at each node of the tree a classification experiment was conducted. Table 3.4 (see Chapter 3) lists the hierarchical organization of the pattern classes. The classification results obtained are shown in Table 7.5.F1. These results giving 76.5% are slightly better than those reported in Table 7.5.A5 which are the results of applying the same classifier on all 10 classes simultaneously. This observation supports our claim that by hierarchically classifying the two associated classes (groups) at a time, misclassification can be reduced. Another experi-



ment was conducted by transforming the 10 class problem into a 4 class problem. In this experiment all three classes involving each of 'deep', 'medium' and 'shallow' defects were combined into one class each. Table 7.5.1 shows the way these 4 classes construct their hierarchy (PAH). Using this tree much better results, with an overall average of 91.50% correct performance, were obtained. The no-defect class remains separate from other classes. Table 7.5.F2 shows the results of this experiment on the testing set. These results i.e., 91.5% correct recognition, show performance superior to any other algorithm of the Discrimination Subsystem. These results indicate that excellent performance can be achieved by reducing the size of the problem; in fact, by grouping the similar classes together.

Table 7.5.1

Hierarchical Organization of Pattern Classes  
(Single Linkage Method)

Class		Node	Left	Right
Shallow	(A)			
Medium	(B)	1	AD	BC
Deep	(C)	2	B	C
No Defect	(D)	3	A	D

Table 7.5.F1

Classification Results on NDT Data using Hierarchical  
Organization (Single Linkage) of Pattern Classes  
(MDC - Mahalanobis Distance)  
(10 Class Problem)

Class output input	1	2	3	4	5	6	7	8	9	10	Recog. %
1	20										100
2		14	1		5						70
3	2	1	13	1		1			1	1	65
4		2	4	14							70
5		5	4		10						50
6		5	4			11					55
7							20				100
8		7			1			12			60
9									20		100
10				1						19	95
Total Recog./No. Misrecog./ Av. Recog.											153 47 76.5

Table 7.5.F2

Classification Results on NDT Data using Hierarchical  
Organization (Single Linkage) of Pattern Classes  
(MDC - Mahalanobis Distance)  
(Four Class Problem)

Class output Input		1	2	3	4	Recog. %
Shallow	(1)	54	2	4		90.0
Medium	(2)		55	5		91.7
Deep	(3)	1	4	54	1	90.0
No Defect	(4)				20	100.0
Total Recog./No. Misrecog./Av. Recog.						183, 17 91.50

#### 7.5.7 Experiment - G: EDT

The EDT algorithm was independently applied to the NDT data and perfect performance of 100% was obtained on the design

set, however, when it was applied to the testing set 76% of 200 patterns were correctly recognized. The results of this experiment are shown in Table 7.5.G1.

Notice that all patterns of the no-defect class were correctly recognized, and only one defect pattern, 'small-deep' was confused as no-defect. An even better result is obtained when the 10 class problem is transformed into a 4 class problem with classes Shallow, Medium, and Deep by combining all three of shallow, medium, and deep defect categories from 10 classes. The no-defect class remains separate and an overall average of 95.50% correct performance was obtained. Table 7.5.G1 shows the results of this experiment on the testing set.

Table 7.5.G1  
Classification Results on NDT Data using EDT  
Algorithm  
(10 Class Problem)

Class output input	1	2	3	4	5	6	7	8	9	10	Recog. %
1	16			4							80
2		10	1		5			4			50
3	2		9	3	1			2	2	1	45
4	4		1	15							75
5	4		1		15						75
6			2	2		15			1		75
7							20				100
8		5			2			13			65
9			1						19		95
10										20	100
Total Recog./No. Misrecog./Av. Recog.								152	48		76

Table 7.5.G2

Classification Results on NDT Data using EDT  
Algorithm  
(Four Class Problem)

Class Output Input	1	2	3	4	Recog. %
Shallow (1)	59		1		98.33
Medium (2)		59	1		98.33
Deep (3)	3	4	52	1	86.70
No Defect (4)				20	100.00
Total Recog./No. Misrecog./Av. Recog.	190	10			95.00

#### 7.5.8 Comments: Performance on NDT Data

From the engineering point of view the performance of the system on the design sets from NDT data was not very impressive. The recognition results were in the range of 25% to 100% for various pattern classes from different algorithms and several of those results are reported in Sections 7.5.1 through 7.5.7. Because of the large variations within and among classes several classifiers, e.g., KNN, and MDC-E failed to give even a half decent performance. The performance of MDC-M, BYC, EDT, and PAH classifiers was marginal. QDC with ranked features produced satisfactory results. However, high performance was obtained when the 10 class problem was reduced to a 4 class problem. In particular PAH-U classifier produced close to perfect performance (87% to 100%).

In majority of the cases lower performance (35% to 65%) is caused by the three smaller defect classes, namely, small/shallow (smsh), small/medium (smme), and small/deep (smde).

By removing these three classes the performance from most of the classifiers reached between 80% and 95%. For example, the performance of MDC-M on classes 5 through 10 (see Table 7.5.A2) reached 89.1%; the performance of MDC-E on the same classes reached 81.7% (see Table 7.5.A3); the performance of QDC reached 90% mark (see Table 7.5.D3); and the performance of BYC reached 85.8 percentage point (see Table 7.5.E1). Lacasse et al. [LACA-88] reported the similar observations on smaller defect classes and showed only 60% performance on a 4 class problem (3 smaller defect classes and one no-defect class) using hand-picked features. Another interesting observation was that, though KNN did not perform well (see Table 7.5.B1 and Table 7.5.B2) it treated all classes equally. The performance on large defect classes was generally very high (90% to 100%) which was equally comparable to other results reported elsewhere [LACA-88, LAMB-89]. This observation should not undermine our approach as we have presented a generic solution and did not care for the size of the problem, characteristics and representation of data, and source and the nature of problem. In addition, all these steps were performed automatically.

We tried to further verify our results with current industry standards and found that the attempts to solve large multi-class defect classification problems are scarce. Major concern on reliability of the NDT techniques comes from the lack of dependable scanning probes and data acquisition equipment to collect data in a noisy or hostile environment. As described in Sections 1.4 and 2.2 a vast majority of industrial NDT problems are solved using human-operator-based signal display systems [STAL-82, SING-92] and as such there is no direct one-to-one match to our solution. There are only a handful of systems which solely perform automated signal classification or used PR based methods. Even these systems, as much we are aware of them, can solve problems with much

smaller dimensions: usually 2 to 4 class problems and using a few manually selected features only.

Industrial systems usually do not rely on one individual NDT test. They generally perform a comprehensive failure analysis. For example, Silvus [SILV-92] reported one such system at Southwest Research Institute and suggested the NDT industry to use specialist, "Failure Analyst" who is placed remotely from the failure site and has full array of communication with the testing staff including extensive photographic and textual documentation on the failure characteristics. A failure analyst performs NDT operations first; these usually include multiples of visual inspections, two/three dimensional view x-radiography, electrical and mechanical checks to confirm reported failure mode and one or more electromagnetic (NDT) tests, and perhaps, other tests that are relevant to a particular type of component. Disassembly, visual inspection and photography at each step come next; during this process other techniques such as scanning-electron micrography and energy dispersion X-ray analysis are employed to provide additional insight into the failure mechanism or to document particular observations. After disassembly is complete, selected parts of the failed device may be cross-sectioned to reveal features that are not visible from their surface. From this comprehensive point of view the routine NDT tests have become a meager component in an integrated system environment where even a marginal performance is acceptable. Several researchers have reported only 65% to 75% performance on primarily two classes from NDT tests [ALDR-92, MOWR-88, SILV-92]. In addition these classes were major defects which were considered potentially dangerous.

Aldrich [ALDR-92] and Singh [SING-92] have also supported developing computer aided design (CAD) approach to disassemble the components so that the operator can inspect the component

from a variety of angles and positions and be able to make a well informed decision. These comprehensive systems are normally supported by huge data bases and the categorization problem is treated as relational data base search (pattern matching) problem. Several large vendors such as Siemens [NPJ-92, NPJ-93], General Electric (GE) [ALDR-92] and EPRI [ROBE-92] have developed and are promoting a variety of hardware and software devices to support the operator-based testing environment. Examples of such devices include Video-mapping, Videodisc, RVT (remote visual testing) and GE's GERIS (GE's Remote Inspection System) which allow the inspection crew to collect data in a hostile environment and receive/transmit from/to a remote site [NPJ-92, NPJ-93, ALDR-92].

Recapping the discussion we can safely conclude that the experiments we conducted have demonstrated that the non-destructive testing and monitoring can be performed in an efficient and cost effective manner without resorting to comprehensive system supported manual testing and disassembly. The PAH-based classification approach supplemented with data dependent rules for the selection of an appropriate algorithm at various stages certainly can substantially improve the classification performance. If nothing at all, at least, our algorithms can be used to identify the defective material from large piles of questionable material and in such situations one would need to simply analyze (test) the defective material only. Some of these results and observations have been reported in [SIDD-94a,b].

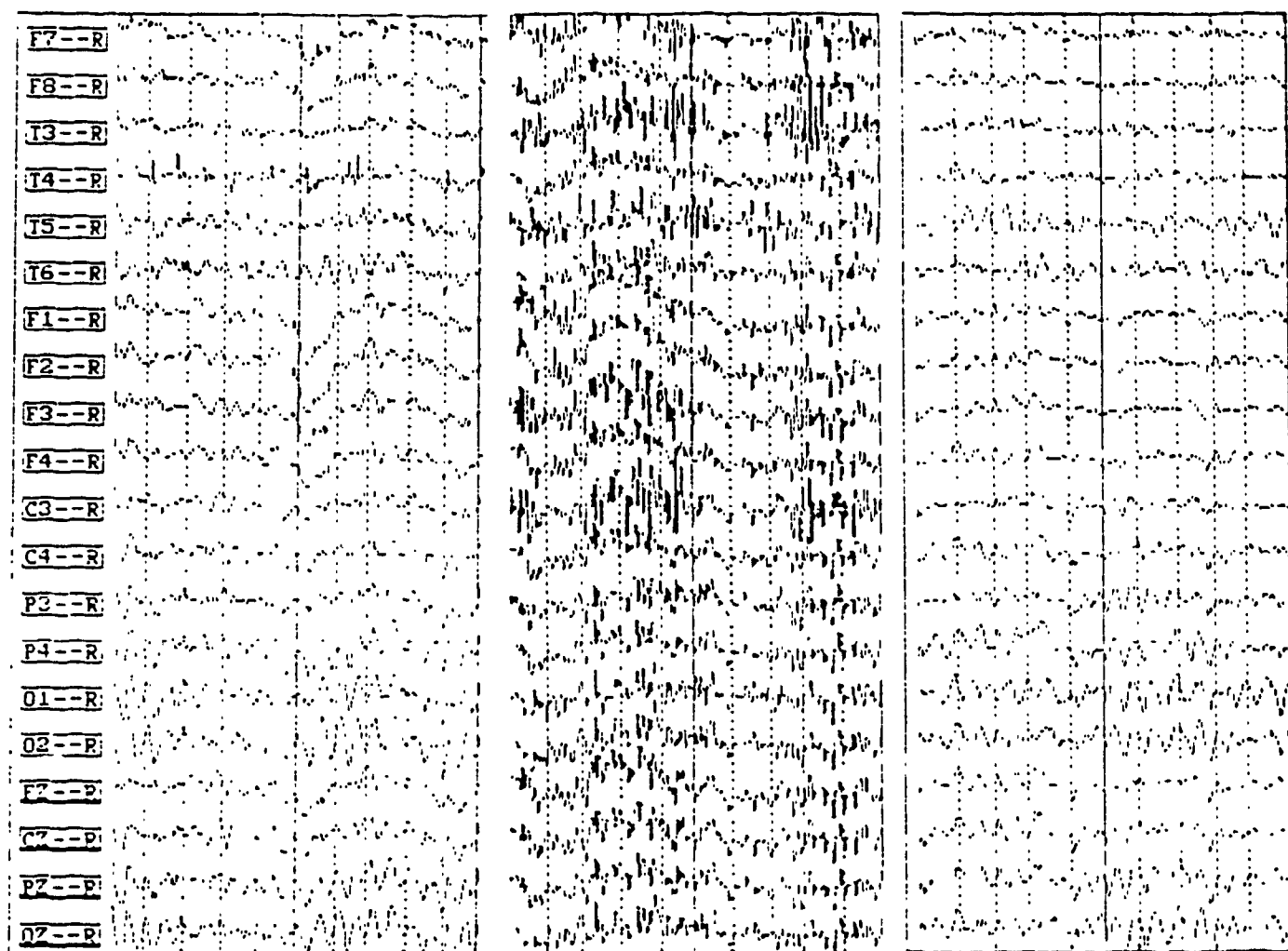
## 7.6 Performance on EEG Data

The data on EEG signals were provided by the Department of Psychiatry, McMaster University, Hamilton, Canada upon the request of Tektrend. The data were collected on three classes, namely, eye-artifacts, muscle-artifacts, and non-artifacts. These classes are respectively called "eye", "mus", and "cle" in the tables reporting the results. Ten EEG channels corresponding to the standard 10-20 system points were applied in the frontal (F4 and F8), central (C4), temporal (T4, T6), parietal (P4), occipital (O2), and corresponding left sided locations on a patient's skull. A plot of each of these classes from various channels is shown in Fig. 7.2. Each class was represented by 100 samples for both the design set and the testing set. Each sample pattern was represented by 112 features, listed in Table 7.6.1. The procedure for data collection and nomenclature of the features are described in Appendix A. The data were one of the cleanest data and hence the problem turned out to be relatively simple. The features from the design data set were processed through the Successive Elimination Process, which was able to delete 90 features listed in Table 7.6.2.

The remaining 22 features constitute the abridged feature set, Feat-A for this data set. Using Feat-A several classification experiments were conducted. The rankings of these features based on Fisher's discriminant index and Pseudo-similarity algorithm are shown in Table 7.6.3. Using these feature sets several additional classification experiments were performed and are reported in sections to follow.



# Electroencephalogram (EEG) Signals from various Channels



## Legend:

X- axis : Time in Unit Second  
Y- axis : One Unit = 55  $\mu$ V (micro volt)

Fig. 7.2. A few typical samples from EEG signals.

Table 7.6.1

Features used for EEG Problem  
(see Appendix A to decode the abbreviations)

F e a t u r e      D o m a i n s				
Statistical	Zero Cross.	Hjorth Slope Descr.	Time/Pulse Shape	Derived from Raw Signal
1: SMV	6: AVF	12: MOB	14: NPK	25: AIN
2: SSD	7: AFD	13: CPX	15: PK1	26: PKD
3: SKF	8: AF1		16: PK2	27: DR1
4: KUR	9: AF2		17: APR	28: DR2
5: CVR	10: AF3		18: APF	29: DR3
	11: AF4		19: PRT	30: DR4
			20: PRS	31: DR5
			21: PFT	
			22: PFS	
			23: PPW	
			24: HPW	

F e a t u r e      D o m a i n s				
Frequency Power Distribution		Auto - Correlation		
Low	High	Shape	Spectra-Dist.	
32: P01, 50: P19, 68: P37	84: PH1	95: APK	105: PP1	
33: P02, 51: P20, 69: P38	85: PH2	96: AP1	106: PP2	
34: P03, 52: P21, 70: P39	86: PH3	97: AP2	107: PP3	
35: P04, 53: P22, 71: P40	87: PH4	98: GPK	108: PP4	
36: P05, 54: P23, 72: P41	88: PH5	99: GPL	109: PP5	
37: P06, 55: P24, 73: P42	89: PH6	100: 2PP	110: PP6	
38: P07, 56: P25, 74: P43	90: PH7	101: 2PA	111: PP7	
39: P08, 57: P26, 75: P44	91: PH8	102: PKA	112: PP8	
40: P09, 58: P27, 76: P45	92: PH9	103: TAR		
41: P10, 59: P28, 77: P46	93: XPH	104: PDS		
42: P11, 60: P29, 78: P47	94: NPH			
43: P12, 61: P30, 79: P48				
44: P13, 62: P31, 80: P49				
45: P14, 63: P32, 81: P50				
46: P15, 64: P33, 82: MXP				
47: P16, 65: P34, 83: MNP				
48: P17, 66: P35				
49: P18, 67: P36				

Table 7.6.2

EEG - Problem: Features Deleted  
Using Successive Elimination Process

---

A. Stationary Features

86	87	88	89	90
91	92	94		

B. Features Deleted with Discordance Test

17	20	22	107	108
109	110	111		

C. Highly Correlated Features  
(Correlation  $\geq 0.3$ , Frequency  $\geq 4$ )

5	7	10	11	12
13	14	15	16	19
23	25	26	27	28
29	30	31	32	33
34	35	36	37	38
39	40	43	44	45
47	48	49	50	51
52	53	54	55	56
57	59	61	63	64
66	68	69	70	71
72	73	74	75	76
77	78	79	80	81
82	83	84	85	93
95	96	97	99	101
102	103	104	106	

D. Features Merged

None

---

Table 7.6.3

EEG - Problem: Feat -F and Feat-S  
Feature Ranking (Fisher and Pseudo-Similarity)

Fisher's Rank		Pseudo-Similarity Rank	
Rank	Feature Id	Rank	Feature Id.
1	8	1	6
2	21	2	1
3	2	3	8
4	1	4	16
5	6	5	5
6	7	6	20
7	4	7	22
8	5	8	10
9	14	9	9
10	18	10	21
11	15	11	19
12	17	12	17
13	9	13	3
14	22	14	15
15	16	15	18
16	13	16	4
17	10	17	14
18	20	18	2
19	19	19	12
20	11	20	13
21	12	21	11
22	3	22	7

## 7.6.1 Experiment A - MDC

The performance on the design set using feature set Feat-A and different weights is shown in Table 7.6.A1. It turned out that if weight, i.e.,  $w = 1/s_i$ , is used, a simple MDC with Euclidean distance can achieve an overall performance of 87.33%. However, reviewing the performance individually, Fisher weights turned out to be the best in identifying eye-

artifacts and muscle-artifacts each with 96%, however, its performance sharply dropped when applied to the 'Clean' signals. The performance of MDC-E classifier on the testing set is shown in Table 7.6.A2. The weight  $1/s_i$  has produced the best overall results recognizing 81% of 300 samples from 3 classes. Table 7.6.A3 shows the results of a similar experiment using MDC-M giving an overall recognition of 93.67% on the training set and 86.67% on the testing set respectively. The performance using this classifier was the best obtained among all MDC classifiers, perhaps for the reason that all the features were appropriately weighed.

We repeated the experiment by selecting 8, 10, 12, 15 and 20 ranked features from both Feat-F and Feat-S sets to determine the best feature size empirically. Their overall performance is shown in graphs of Fig. 7.3. The graph shows that a much smaller feature set can achieve satisfactory results unless the features selected are the most discriminatory ones. The last experiment was repeated on feature sets Feat-F and Feat-S, their performance on testing sets for the best performing features (15 features) are respectively shown in Table 7.6.A4 and Table 7.6.A5. Both of these tables show that the system was able to successfully identify the most discriminatory features, 15 in this case, to achieve the peak performances of 84.33% and 91.33% from the classifiers using feature sets Feat-F and Feat-S respectively. This experiment also shows that Pseudo-similarity algorithm is a better ranking algorithm when the classes are homogeneous.

The experiments reported in Table 7.6.A4 and 7.6.A5 also show the comparative performance of various weights. It turned out that weight  $1/s_i$  was overall best weighing scheme for feature set Feat-S and was able to recognize 88% of 'eye', 100% of 'mus', and 86% of 'cle' signals giving an overall average of 91.33%. The weight  $sd_i/m_i$  is a poor performer implying that

Table 7.6.A1

EEG Problem: Design Set  
 Classification Results  
 (Classifier - MDC-E)  
 (Feature Set: Feat-A = 22)

 $w = 1$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	89	5	6	89
mus		94	6	94
cle	19	26	55	55
Overall Average:				79.33

 $w = 1/sd_i$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	88	2	10	88
mus		93	7	93
cle	2	17	81	81
Overall Average:				87.33

 $w = m_i / sd_i$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	94		6	94
mus		95	5	95
cle	20	24	56	56
Overall Average:				81.67

 $w = m_i / sd_i$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	91	3	6	91
mus		95	5	95
cle	8	27	65	65
Overall Average:				83.67

 $w = f_i$  (Fisher weight)

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	96		4	96
mus		96	4	96
cle	32	15	53	53
Overall Average:				81.67

 $w = sd_i / m_i$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	88	8	4	88
mus	6	89	5	89
cle	22	27	51	51
Overall Average:				76.00

Table 7.6.A2

EEG Problem: Testing Set  
 Classification Results  
 (Classifier - MDC-E)  
 (Feature Set: Feat-A = 22)

 $w = 1$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	94	1	5	94
mus	2	91	7	91
cle	52	23	25	25
Overall Average:				70

 $w = 1/sd_i$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	93	0	7	93
mus		91	9	91
cle	30	11	59	59
Overall Average:				81

 $w = m_i / sd_i$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	92	1	7	92
mus		93	7	93
cle	49	18	33	33
Overall Average:				72.67

 $w = m_i / sd_i$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	94	1	5	94
mus		93	7	93
cle	39	26	35	35
Overall Average:				74

 $w = f_i$  (Fisher weight)

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	94		6	94
mus	1	93	6	93
cle	63	8	29	29
Overall Average:				72

 $w = sd_i / m_i$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	88	8	4	88
mus	6	88	6	88
cle	52	25	23	23
Overall Average:				63.33

mean is not a good scaling factor whereas standard deviation is.

Table 7.6.A3

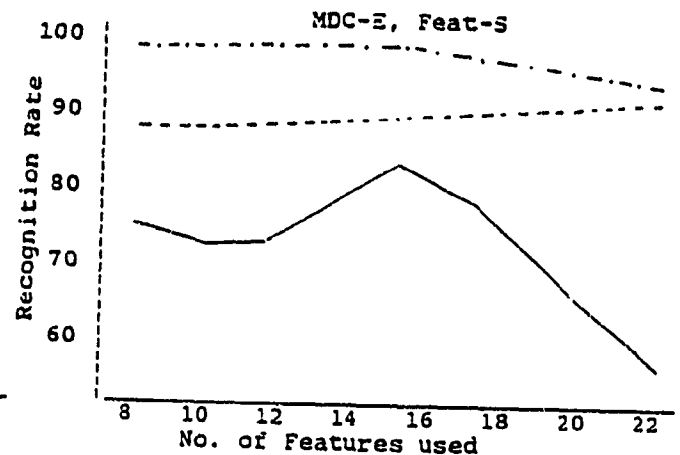
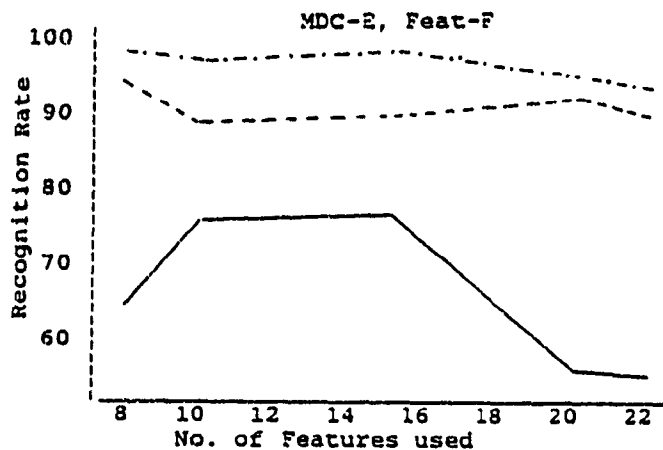
EEG Problem: Design / Testing Set  
Classification Results  
(Classifier - MDC-M)  
(Feature Set: Feat-A = 22)

Design Set

input Class	Output eye	Class mus	Recog. cle	Recog. %
eye	94	1	5	94
mus	2	97	1	97
cle	9	1	90	90
Overall Average:				93.67

Testing Set

input Class	Output eye	Class mus	Recog. cle	Recog. %
eye	93		7	93
mus	1	98	1	98
cle	30	1	69	69
Overall Average:				86.67



Legend:

eye = .....  
mus = .....  
cle = ———

Fig. 7.3: Recognition Performance versus Number of Features.



Table 7.6.A4

EEG Problem: Testing Set  
 Classification Results  
 (Classifier - MDC-E)  
 (Feature Set: Feat-F = 15)

 $w = 1$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	87		13	87
mus	2	96	2	96
cle	37	3	60	60
Overall Average:				81

 $w = 1/sd_1$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	85	1	14	85
mus		100		100
cle	28	3	69	69
Overall Average:				84.33

 $w = m_1 / sd_1$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	94		6	94
mus	2	96	2	96
cle	41	3	56	56
Overall Average:				82

 $w = 1 / sd_0$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	89	1	10	89
mus	2	96	2	96
cle	36	2	62	62
Overall Average:				82.33

 $w = f_1$  (Fisher weight)

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	79		21	79
mus		88	12	88
cle	43	2	55	55
Overall Average:				74

 $w = sd_1 / m_1$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	83		17	83
mus	6	92	2	92
cle	36	1	63	63
Overall Average:				79.33

Table 7.6.A5

EEG Problem: Testing Set  
 Classification Results  
 (Classifier - MDC-E)  
 (Feature Set: Feat-S = 15)

 $w = 1$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	72		28	72
mus	1	97	2	97
cle	5	6	89	89
Overall Average:				86

 $w = 1/sd_i$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	88		12	88
mus		100		100
cle	9	5	86	86
Overall Average:				91.33

 $w = m_i / sd_i$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	79		21	79
mus	1	96	3	96
cle	27	15	64	64
Overall Average:				79.33

 $w = 1 / sd_o$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	78		22	78
mus		99	1	99
cle	5	2	93	93
Overall Average:				90

 $w = f_i$  (Fisher weight)

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	78		22	78
mus		87	13	87
cle	45	2	53	53
Overall Average:				72.67

 $w = sd_i / m_i$ 

input Class	Output Class			Recog. %
	eye	mus	cle	
eye	63		37	63
mus	11	61	28	61
cle	12	2	86	86
Overall Average:				70

### 7.6.2 Experiment B - KNN

Using KNN the performance of the classifier on the design sets for EEG data was in the range of 70% to 100% for various feature sets. Table 7.6.B1 shows the performance of KNN-E ( $k = 3$ ) for Feat-A=22 on the design set whereas the Table 7.6.B2 shows its performance on the testing set. On the design set the highest average performance of 89.2% was achieved using Fisher's weight. The KNN-E with weight  $1/s_i$  was able to recognize overall 85.9% of 300 samples in the testing set. We reported the number of rejects for each class as well. In some cases, particularly, for the 'clean' class several samples were rejected. This observation, while indicating the presence of a few outliers, also shows the reliability of the classifier. A similar experiment was repeated using Mahalanobis distance to determine the neighborhood, the results of which are shown in Table 7.6.B3. The results reported in this table were even better than the those reported in Table 7.6.B2 achieving a high recognition rate of 92.2%. The classes 'eye', 'mus' and 'cle' were correctly recognized respectively with 95%, 96%, and 85.6%. This observation shows that Mahalanobis distance which takes into account variations in the data may be a useful alternative to weighed Euclidean distance. Since the KNN classifiers are useful for homostat type data, it may be appropriate to conclude that EEG data might fit the definition of homostat data.

Table 7.6.B1

Classification Results on EEG Data using  
 Linear Organization of Pattern Classes  
 (3NN - Euclidean)  
 (Feature Set: Feat-F = 22)  
 (Design Set)

 $w = 1$ 

input Class	Output Class			Rej.	Recog. %
	eye	mus	cle		
eye	93	5	2	0	93
mus	6	86	8	0	86
cle	7	29	57	7	61.3
Overall Average:					80.5

 $w = 1/sd_1$ 

input Class	Output Class			Rej.	Recog. %
	eye	mus	cle		
eye	99	1	0	0	99
mus	3	90	7	0	90
cle	3	26	70	1	70.71
Overall Average:					86.62

 $w = m_1 / sd_1$ 

input Class	Output Class			Rej.	Recog. %
	eye	mus	cle		
eye	93	6	1	0	93
mus	8	85	6	1	85.9
cle	9	34	53	4	55.2
Overall Average:					78.3

 $w = 1 / sd_0$ 

input Class	Output Class			Rej.	Recog. %
	eye	mus	cle		
eye	95	5	0	0	95
mus	3	89	7	1	89.9
cle	6	32	61	1	61.6
Overall Average:					82.2

 $w = f_1$  (Fisher weight)

input Class	Output Class			Rej.	Recog. %
	eye	mus	cle		
eye	97	2	1	0	97
mus		97	3	0	97
cle	8	18	71	3	73.2
Overall Average:					89.2

 $w = sd_1 / m_1$ 

input Class	Output Class			Rej.	Recog. %
	eye	mus	cle		
eye	94	5	1	0	94
mus	7	85	8	0	85
cle	5	25	59	11	66.3
Overall Average:					82.4

Table 7.6.B2

Classification Results on EEG Data using  
 Linear Organization of Pattern Classes  
 (3NN - Euclidean)  
 (Feature Set: Feat-F = 22)  
 (Testing Set)

 $w = 1$ 

input Class	Output Class			Rej.	Recog. %
	eye	mus	cle		
eye	92		8	0	92
mus	6	88	5	1	88.9
cle	26	8	60	6	63.8
Overall Average:					81.9

 $w = 1/sd_1$ 

input Class	Output Class			Rej.	Recog. %
	eye	mus	cle		
eye	94		6	0	94
mus	4	93	3	0	93
cle	29		68	3	70.1
Overall Average:					85.9

 $w = m_1 / sd_1$ 

input Class	Output Class			Rej.	Recog. %
	eye	mus	cle		
eye	88	2	10	0	88
mus	6	89	5	0	89
cle	39	3	50	8	54.4
Overall Average:					77.7

 $w = 1 / sd_0$ 

input Class	Output Class			Rej.	Recog. %
	eye	mus	cle		
eye	91	1	7	1	91.9
mus	3	93	3	1	93.9
cle	27	3	64	6	68.1
Overall Average:					84.9

 $w = f_1$  (Fisher weight)

input Class	Output Class			Rej.	Recog. %
	eye	mus	cle		
eye	94		6	0	94
mus	2	92	4	2	93.9
cle	31	2	64	3	66
Overall Average:					84.8

 $w = sd_1 / m_1$ 

input Class	Output Class			Rej.	Recog. %
	eye	mus	cle		
eye	90		8	2	91.8
mus	5	88	6	1	88.9
cle	20	6	62	12	70.5
Overall Average:					84.2

Table 7.6.B3

Classification Results on EEG Data using  
 Linear Organization of Pattern Classes  
 (3NN - Mahalanobis)  
 (Feature Set: Feat-A = 22)

Design Set						Testing set					
input Class	Output Class			Rej.	Recog. %	input Class	Output Class			Rej.	Recog. %
	eye	mus	cle				eye	mus	cle		
eye	98		2	0	98	eye	95	1	4	0	95
mus	2	95	2	1	96	mus	1	95	3	1	96
cle	4	2	92	2	93.9	cle	12	2	83	3	85.6
Overall Average:					95.95	Overall Average:					92.2

### 7.6.3 Experiment - C: LDC

The performance of linear discriminant classifier (LDC) on EEG data was examined and the results were very encouraging. The results on the design set and the testing set are reported in Table 7.6.C1. On the design set the classifier was able to obtain 93.67% correct recognition. However, on testing, over 86% of unknown samples were correctly recognized. This shows that if the data quality is better and most useful features are selected a simple classifier like LDC can do the job.

Table 7.6.C1

EEG Problem  
Classification Results - LDC  
(Feature Set: Feat-A = 22)

Design Set					Testing Set				
input Class	Output Class			Recog. %	input Class	Output Class			Recog. %
	eye	mus	cle			eye	mus	cle	
eye	94	1	5	94	eye	93		7	93
mus	2	97	1	97	mus	1	98	1	98
cle	9	1	90	90	cle	30	1	69	69
Overall Average:				93.67	Overall Average:				86.67

#### 7.6.4 Experiment - D: QDC

The performance of the quadratic discriminant classifier (QDC) on EEG data was examined as well and the results were very impressive producing a range of 86% to 100% for various feature sets. Using the feature set Feat-A=22, the classification results on the design set and the testing set are reported in Table 7.6.D1. On the design set the classifier was able to obtain 94.33% correct recognition. However, on the testing set over 88% of unknown samples were correctly recognized. Although the data appears to be "homostat", the "cle" class still seems to have large variations between several of its samples and as a result the performance of this class improved significantly using QDC. It reached 86% mark - the highest among all classifiers reported in sections 7.6.1 through 7.6.3. Similar experiment was repeated using the optimal feature sets Feat-F and Feat-S (15 features each), giving an average performance of 88.67% and 90%, respectively. These results are reported in Table 7.6.D2 and Table 7.6.D3. The

results reported in Table 7.6.D2 were comparatively similar to those reported in Table 7.6.D1 except that the performance on class 'eye' was dropped whereas the performance on other two classes improved. This observation shows that satisfactory results may be obtained if best features are used, however, the features ranked by Pseudo-similarity algorithm appear to give slightly better results than the one ranked by the Fisher index. The reason for the difference in the performance is due to the way two algorithms rank the features and perhaps due to the characteristics of the data. Since the EEG data had small amounts of variations within classes (nearly homostat) the Pseudo-similarity ranking performed slightly better. By reviewing the confusion matrices of these tables (Table 7.6.D2 and Table 7.6.D3) another observation we made is that the two classes, i.e., 'eye' and 'mus' were mainly confused with 'cle' and there was little or no confusion among themselves. Theoretically, since QDC classifiers are useful for "segregates" type data, and although it is computationally more expensive, it is always safer to use QDC when within class variations are detected.



Table 7.6.D1

Classification Results on EEG Data using  
 Linear Organization of Pattern Classes  
 (Classifier: QDC)  
 (Feature Set: Feat-A = 22)

## Design Set

input Class	Output eye	Class mus	Recog. cle %	
eye	88	1	11	88
mus	0	95	5	95
cle	0	0	100	100
Overall Average:			94.33	

## Testing Set

input Class	Output eye	Class mus	Recog. cle %	
eye	89	0	11	89
mus	0	91	9	91
cle	14	0	86	86
Overall Average:			88.67	

Table 7.6.D2

Classification Results on EEG Data using  
 Linear Organization of Pattern Classes  
 (Classifier: QDC)  
 (Feature Set: Feat-F = 15)

## Design Set

input Class	Output eye	Class mus	Recog. cle %	
eye	84	2	14	84
mus	0	97	3	97
cle	1	3	96	96
Overall Average:			92.33	

## Testing Set

input Class	Output eye	Class mus	Recog. cle %	
eye	80	2	18	80
mus	0	97	3	97
cle	11	0	89	89
Overall Average:			88.67	

Table 7.6.D3

Classification Results on EEG Data using  
 Linear Organization of Pattern Classes  
 (Classifier: QDC)  
 (Feature Set: Feat-S = 15)

Design Set					Testing Set				
input Class	Output Class			Recog. %	input Class	Output Class			Recog. %
	eye	mus	cle			eye	mus	cle	
eye	88	1	11	88	eye	84	1	15	84
mus	0	98	2	98	mus	0	97	3	97
cle	1	2	97	97	cle	10	1	89	89
Overall Average:				94.33	Overall Average:				90

#### 7.6.5 Experiment - E: BYC

The Bayesian classifier was also applied to all 3 classes simultaneously and an overall average of 90.67% correct was obtained. Detailed results are reported in Table 7.6.E1. These results as compared to the four earlier experiments reported in sections 7.6.1 through 7.6.4 (see Tables 7.6.A2 through 7.6.A5, Tables 7.6.B2/3, Table 7.6.C1, and Tables 7.6.D1 through 7.6.D3) are much better and can be attributed to more accurate evaluation of class probabilities since each class was represented by 100 samples. The BYC algorithm was able to distinguish 'eye', 'mus', 'cle' classes respectively with 87%, 95% and 90% which is the best performance we obtained among all the algorithms reported in sections 7.6.1 through 7.6.4. It can be concluded that BYC is always a better choice when sample size is sufficiently large.

Table 7.6.E1

Classification Results on EEG Data using Linear  
 Organization of Pattern Classes  
 (Classifier: BYC)  
 (Feature Set: Feat-S = 15)

Design Set					Testing Set				
input Class	Output Class			Recog. %	input Class	Output Class			Recog. %
	eye	mus	cle			eye	mus	cle	
eye	91	1	8	91	eye	87	2	11	87
mus	0	99	1	99	mus	1	95	4	95
cle	1	2	97	97	cle	9	1	90	90
Overall Average:				95.67	Overall Average:				90.67

#### 7.6.6 Experiment - F: PAH

The MDC classifier was applied on a PAH tree constructed by the single linkage algorithm (SLA). By applying MDC-M classifier at each node of the tree a classification experiment using feature set Feat-A=22 was conducted. Table 7.6.4 lists the hierarchical organization of the pattern classes obtained by SLA. The classification results obtained are shown in Table 7.6.F1. These results giving 95.67% are better than those reported in Table 7.6.A3 which are the results of applying the same classifier on all 3 classes simultaneously. This observation again confirms that by hierarchically classifying the two associated classes (groups) at a time misclassification can be reduced. These results i.e., 90.67% correct recognition show performance superior to any other algorithm of the Discrimination Subsystem. These results indicate that excellent performance can be achieved by reducing the size of the problem. Another experiment using PAH-V with MDC-M

classifier at the root node and QDC classifier at node 2 was conducted and a perfect performance was obtained.

A third experiment was conducted using MDC-E classifier and 15 Fisher ranked features, i.e., Feat-F at the first node and 15 Pseudo-similarity ranked features, i.e., Feat-S at the second node and an average performance of 90.67% was obtained, correctly classifying 'eye', 'mus', and 'cle' classes respectively with 92%, 96% and 84%. These results are reported in Table 7.6.F2.

Table 7.6.4

Hierarchical Organization of Pattern Classes  
(Single Linkage Method)

Class		Node	Left	Right
eye	(A)			
mus	(B)	1	A	BC
cle	(C)	2	B	C

Table 7.6.F1

Classification Results on EEG Data using Hierarchical  
Organization (Single Linkage) of Pattern Classes  
(MDC - Mahalanobis Distance)

Class output		1	2	3	Recog. %
Input					
eye	(1)	95	2	3	95
mus	(2)	0	98	2	98
cle	(3)	1	5	94	94
Total Recog./Misrecog./Av. Recog.					287, 13 95.67

Table 7.6.F2

Classification Results on EEG Data using Hierarchical  
 Organization (Single Linkage) of Pattern Classes  
 (MDC - Euclidean Distance)  
 (Multiple Feature Sets: Node 1: Feat-F, Node 2: Feat-S)

Class output		1	2	3	Recog.
Input					%
-----					
eye	(1)	92	0	8	92
mus	(2)	0	96	4	96
cle	(3)	16	0	84	84
-----					
Total	recog./Misrecog./Av. Recog.	272, 28			90.67
-----					

#### 7.6.7 Comments: Performance on EEG Data

As described earlier, the data was the cleanest and was sufficient in size to apply any classifier. However, the performance is enhanced and the amount of computation was significantly reduced by selecting a smaller set of the most discriminatory features, appropriately weighing them and automatically choosing the most suitable classifier.

Consequently, the recognition performance of the system on the EEG data was very high and the recognition results were in the range of 80% to 100% for various pattern classes from different algorithms and some of those results are reported in Sections 7.6.1 through 7.6.6. Because the data was less noisy all classifiers, including MDC-E, KNN, and LDC were successful in achieving over 80% performance. The performance of MDC-M, BYC, QDC, and PAH classifiers was superior. However, different classifiers performed differently on various classes.

In some of the cases lower performance (51% to 80%) is caused by the 'cle' class. By removing this class the performance

from most of the classifiers tops out to 90% to 100%. For example, the performance of MDC-E on two classes (see Table 7.6.A2, tablets 3 and 4) reached 93.5%; the performance of MDC-M on the same classes reached 95.5% (see Table 7.6.A3); KNN ( $k=3$ ) was able to produce 94% correct results (see Table 7.6.B2); LDC achieved the performance of 95.5%. QDC treated all classes more fairly, recognizing the three classes with 89%, 91%, and 86% respectively giving an average performance of 88.67% (see Table 7.6.D1) using the feature set Feat-A=22. Almost similar performance was obtained using the set Feat-F=15, however, using the set Feat-S the performance reached 90% mark (see Table 7.5.D3). These results show that classes 'eye' and 'mus' have less noise than the 'cle' class. Clinically it can be interpreted that the signals from the 'cle' class may have been corrupted with the patient's eye blinks since in majority of cases this class is confused with the 'eye' class. Slight confusion occurred with the 'mus' class implying that the interference from muscles was very little. The performance of BYC reached over 90% correctly recognizing 87% of 'eye', 95% of 'mus' and 90% of 'cle' signals (see Table 7.6.E1). By applying MDC-M and QDC classifiers on a PAH tree (see Section 7.6.6) the PAH-V was able to achieve perfect results. Interestingly, using different feature sets at various nodes MDC-E was able to achieve above 90% performance (see Table 7.6.F2). This experiment, thus strongly supports the idea of applying different feature sets at different nodes of the tree in order to obtain a higher performance.

We tried to verify our results with current clinical standards and found that the attempts to solve EEG signal classification problems using PR methodology are scarce. It is only recently, researchers reported a few efforts on simple two class problems [GEVI-86, TATS-88, SIDD-90c]. It is partly due to the reason that the instantaneous sources of brain electric and magnetic fields are unknown in number, position, and orientat-

ion. Physicians suspect that at any instant only a small fraction of the brain's hundreds of simultaneously active major systems might be performing processing related to the sensory, motor, or cognitive functions being studied [GEVI-87]. Major concern on reliability of the EEG signals comes from the difficulty in isolating the neurological effects of a disease that are buried in pathological and unrelated physiological indications (see Fig. 2.2 in Chapter 2). In the case of brain electrical sources, irregular resistive tissue enclosing the sources spread and distort the resulting pattern of signals. Apart from the complexity of the problem, since human life is directly involved, the factor of disbelief on part of the physicians has played a significant role in discouraging automation efforts. Consequently, EEG classification is entirely done by physicians or technically certified medical staff. Major focus of these efforts has been towards the development of enhanced display of EEG waveforms and analysis and extraction of parameters so that a physician can make a well informed decision.

Surprisingly, we found that Grajski et al. [GRAJ-86] developed a decision tree based classification approach only conceptually related to the EDT algorithm and it was applied to a five class problem giving 97% correct recognition. In their approach the tree was constructed by successively selecting a feature using partitioning type clustering algorithm picking a pair of groups which maximizes the split between two groups and was specifically developed for the problem at their hand. Computationally their approach makes the tree construction process an exponential one. In addition they resampled the data at every non-terminal node to maximize the characteristics of the pattern classes and as such they collected the cleanest possible samples. In our opinion and due to several concerns the medical community has, such controlled way of data acquisition has no practical value [GEVI-87].

The experiments we conducted here have clearly demonstrated a superior performance on EEG signals. Perfect performance from classifiers such as PAH-U and PAH-V have clearly surpassed any of their counterparts. Any classification or monitoring experiment can be performed in an efficient and cost effective manner without resorting to comprehensive system supported visual display and manual interpretation.

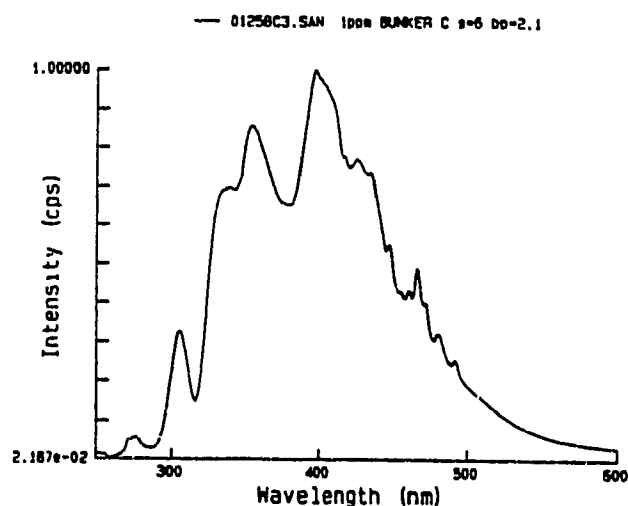
#### **7.7 Performance on the PNA Data**

Using the same system with the PNA-knowledge base 20 ultra-violet visual (UV-vis) synchronous fluorescence spectra of petroleum oils (polynuclear aromatic hydrocarbons) of various origins were used to train the system (see Table 7.7.1). The data on this problem were provided in the form of spectra with 351 data points between 260 to 610 nanometer on frequency scale by the Lockheed Engineering and Sciences Company, Las Vegas, NV. A few spectra of different types of oil are shown in Fig. 7.4. Additional details on the data are available in Appendix B.

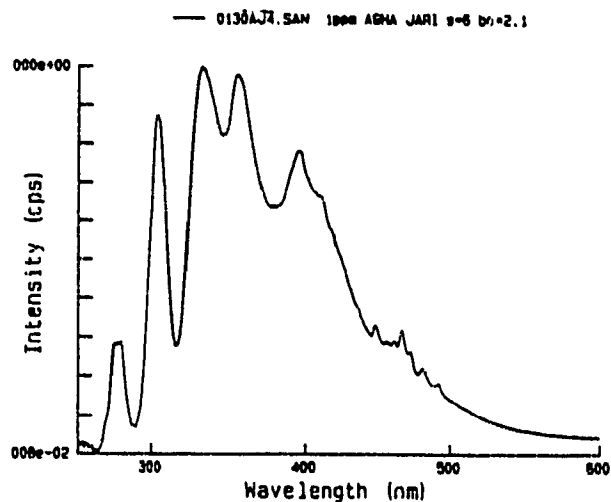
The PNA problem is basically a spectral classification problem and requires the algorithm to identify the structure of the spectra which in turn is associated to a particular compound. Traditionally, this is done by identifying the peaks, their amplitudes and peak positions which are compared with reference patterns or searched through large data base libraries. The extent of applying PR methods was to use grouping or elastic matching. Eastwood et al. [EAST-91] and Siddiqui et al. [SIDD-91a] introduced advanced PR methods to this area and several results on different chemometric problems are reported in [EAST-91, SIDD-89e, SIDD-91a]. Classification of PNA compounds is important from both scientific and environmental points of view. Scientifically we can identify the chemical structure of liquid compounds non-invasively and thus can



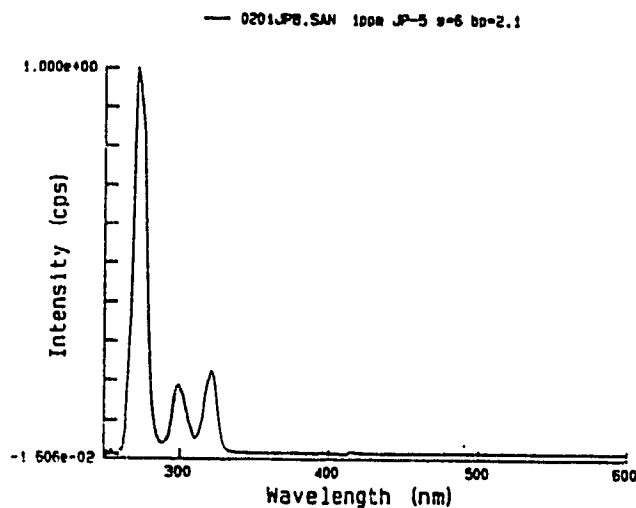
# Polynuclear Aromatic Hydrocarbons (PNA) Spectra for various Oils



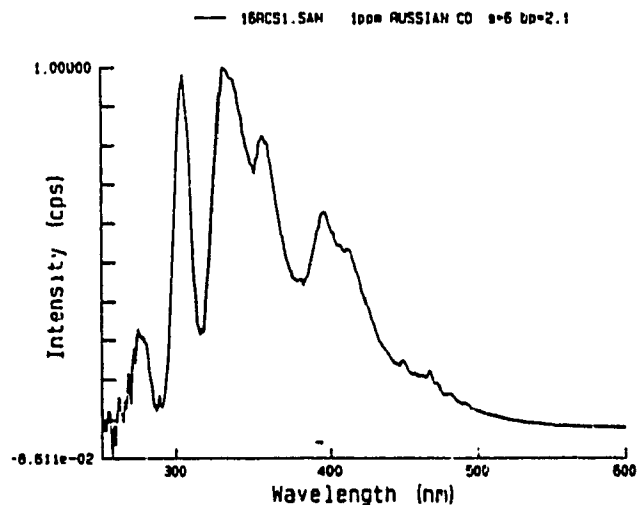
a: Oil 2 (Bunker Crude)



b: Oil 5 (Agha Jari)



c: Oil 7 (JP-5)



d: Oil 13 (Russian Crude Oil)

## Legend:

X- axis : Wavelegth in 100 Units  
Y- axis : Intensity in concentration per second,  
1 Unit = .1 cps

Fig. 7.4. A few sample spectra of Petroleum Oils (PNA's)

eliminate the need for expensive and environmentally hazardous chemical testing laboratories. A majority of PNA compounds are hazardous to life and environment. Spectral classification methods can be efficiently applied to control the level of pollutants in air and waters, and to control and monitor the industrial waste and emission.

Extracting the envelopes from the spectra we identified the number of visible peaks. For the present problem we identified 3 to 10 peaks in each spectrum. From each envelope we extracted 10 features which are listed in Table 7.7.2.

The organization of the hierarchical association between different groups of oils using a centroid method is shown in Table 7.7.3. The Mahalanobis distances between the oils ranged from 0.514 for the most similar pair of oils to 6.463 for the two least similar groups. The design set was then tested using MDC-M classifier and a correct performance of 100% was obtained within a variable threshold ranging from 0.50 to 0.75. Using the trained classifier six unknown oil spectra were also correctly classified. The distances of the unknown samples from the closest matched oil in the design set and the values of the expert knowledge parameters used for the classification are shown in Table 7.7.4. As seen from the table the system was able to correctly classify oils unk2, unk3, and unk6 using the distance alone, whereas for other oils, it had to use additional features identified by a practicing expert [EAST-91] and included features such as wavelength, peak positions, and relative amplitudes.

Table 7.7.1

PNA Pattern Classes, their Labels and  
number of Samples

Label	Class Name	Number of Samples
A	South Carolina Crude Oil	5
B	Bunker Crude	5
C	Arabian Light Crude Oil	5
D	Energy Coop	5
E	Agha Jari	5
F	Sarir	5
G	JP-5	5
H	DFM Petroleum Equivalent	5
I	Marathon 6	5
J	Eastern Coop 6H5	5
K	Exxon 6 Fuel Oil	5
L	2 Fuel Oil	5
M	Russian Crude Oil	5
N	Prudho Bay	5
O	Kern River	5
P	Mesa Crude	5
Q	Arabian Heavy Oil	5
R	DFM Product	5
S	Venezuelan	5
T	5 Fuel Oil	5
Total		100

Table 7.7.2

## PNA Problem - Features Extracted

Feature Id	Feature Name
1	Number of Peaks
2	Ascending span of peak
3	Descending Span of peak
4	Ascending Slope
5	Descending Slope
6	Peak Location
7	Base Width
8	Top Width
9	Width of half rise
10	Width of half descend

Table 7.7.3

Pattern Association Hierarchy using  
Centroid Method (PNA Data)

node	cluster-q	cluster-r	distance
1	ASIHLTECQD	MNPBFKJOG	2.7245
2	ASI	HLTECQD	6.4632
3	AS	I	2.7481
4	MNPBFKJO	GR	1.5494
5	MNPBFKJ	O	1.4719
6	MNPBFK	J	2.0091
7	G	R	1.5553
8	MNPB	FK	1.5494
9	HLTE	CQD	1.5493
10	HLT	E	1.0344
11	HL	T	1.4725
12	CQ	D	1.3680
13	MNP	B	0.7282
14	MN	P	1.2619
15	F	K	1.2352
16	A	S	0.9449
17	C	Q	0.6744
18	M	N	0.5713
19	H	L	0.5142

Table 7.7.4  
Classification Results Using Spectral and Heuristic Parameters

Values of Features and Heuristic Parameters														
Test Oil	Training Oil	Distance		No of Peaks		Peak Location		Peak Amplitude						
		Ts	Threshold	Tr	Ts	Tr	Ts	Tr	Ts					
unk1	oil1	0.813	0.75	6	6	1	274	274	0.266	0.266	0.266			
												:	:	:
												6	433	432
unk2	oil8	0.427	0.50	3	3	1	277	277	0.192	0.192	0.191			
												:	:	:
unk3	oil12	0.250	0.50	3	3	1	278	278	0.274	0.274	0.186			
												:	:	:
unk4	oil13	0.801	0.50	8	7	1	274	267	0.205	0.205	0.223			
												:	:	:
												6	411	409
unk5	oil14	1.989	0.50	6	6	1	274	276	0.184	0.184	0.190			
												:	:	:
												6	419	418
unk6	oil19	0.030	0.50	4	4	1	274	276	0.183	0.183	0.182			
												:	:	:

Tr = Training set data  
Ts = Testing set data  
\* = Nanometers  
\*\* = Normalized Intensity

## 7.8 Performance on the CEL Data

Physiology, the science of natural phenomena in living matter such as organs, tissues, chromosomes, and cells, has been the key field for understanding internal communication and control in the biological system. Physiological research heavily relies on our ability to measure chemical and electrochemical activities taking place in the cells. Many functions of cells (e.g., neural, muscular, etc.) are chemical in nature. These functions, however, produce changes in the electric field which can be monitored by electrodes [COHE-86a]. The bioelectric potentials help physiologists study cell functions.

Usually, the source of the bioelectric signal is a group of cells. The accumulated effects of all active cells in the vicinity produce an electric field which propagates in the volume conductor consisting of various tissues of the body. The activity of a group of cells such as muscle, or some biological system, can thus indirectly be measured by means of electrodes placing on the skin. Fig. 7.5 schematically depicts the basic structure of a nerve cell, called neuron. The important parts of the neuron are the cell body (soma), the dendrites, and the axon. The cell body consists of the intracellular fluid with the various bodies required for the functioning of a cell [SPEK-87]. It is surrounded by an excitable membrane. The cell membrane is extended in various places to generate root-like structures called dendrites. These extensions are used for interconnections with other nerve cells. The axon serves as the output of the nerve unit. Some membranes have excitability characteristics. When the membrane is excited by means of electrical stimulus, the permeabilities of the membrane to ionic transfer undergo some changes. These changes cause the resting potential of the membrane to increase, become positive for a short period of time, and later, when the membrane repolarizes, to return to

its normal resting potential. The time course of the action potential, is shown in Fig. 7.6. CEL data, in fact, represent such action potentials for some undisclosed kind of cells. The data were collected on 19 classes of those cells, and 85 features were extracted from their action potentials. These features were furnished by Tektrend in the form of feature vectors. For proprietary reasons the identity of features and the pattern classes were never disclosed to the author. Therefore, to identify various classes we simply used the coded information to label the classes and the order of the features was used to identify them (features). This problem was most difficult in the sense each class had a different number of samples varying from 7 to 32 and that the class size was large. Table 7.8.1 lists this information. The data was arbitrarily split into two sets; number of samples included in each set is also listed in Table 7.8.1. This varied nature of the problem made the training slightly cumbersome. For several classes the data was too small to have any recognizable effect in the classification process, particularly when the single layer classification algorithms were used. The Successive Feature Elimination processor was able to remove 45 features listed in Table 7.8.2. Removing these features an abridged feature set of 40 features was obtained and it will be referred to as Feat-A=40. The feature set Feat-A was submitted to the ranking algorithms. Table 7.8.3 presents the results obtained from Fisher's discriminant index and Pseudo-Similarity ranking algorithms.

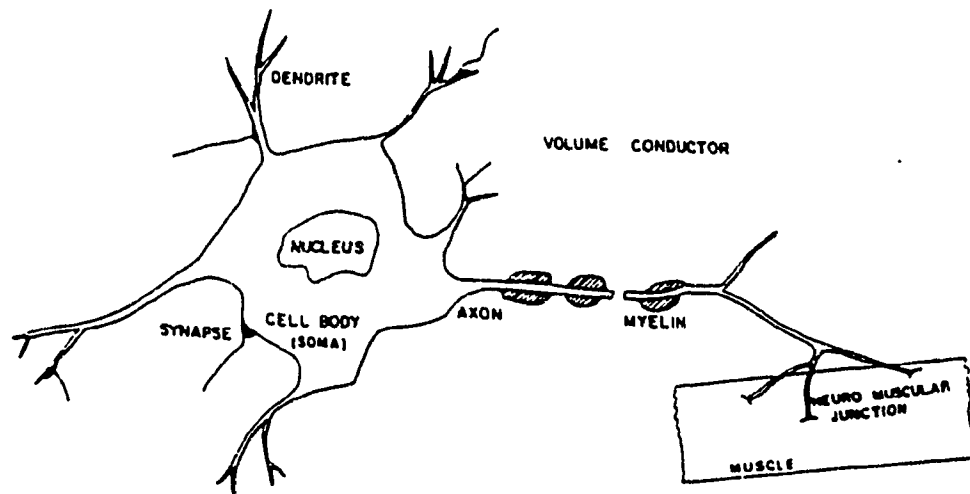


Fig. 7.5: Schematic structure of a nerve cell (neuron).

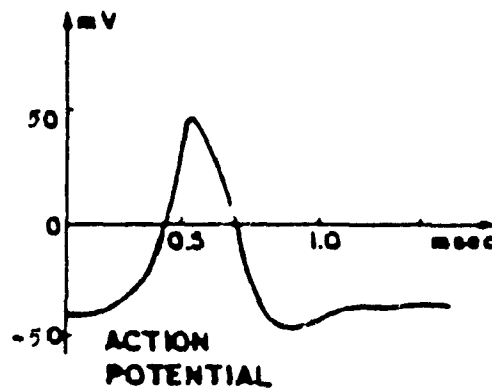


Fig. 7.6: The time course of the action potential.



Table 7.8.1

## CEL Data Pattern Classes

Class Label	Class Name	No. of Samples		
		Total	Design Set	Test. Set
A	Class AA	29	15	14
B	Class AB	30	15	15
C	Class AC	30	15	15
D	Class AD	30	15	15
E	Class AE	12	7	5
F	Class AF	16	10	6
G	Class AG	32	16	16
H	Class AH	30	15	15
I	Class AI	7	5	2
J	Class AJ	19	10	9
K	Class AK	30	15	14
L	Class AL	29	15	15
M	Class AM	30	15	15
N	Class AN	19	10	9
O	Class AO	29	15	14
P	Class AP	30	15	15
Q	Class AQ	20	10	10
R	Class AR	19	10	9
S	Class AS	20	10	10
Total		461	238	223

Table 7.8.2

CEL - Problem : Features Deleted  
Using Successive Elimination Process

---

A. Stationary Features

6	23	24	25	26
27	63			

B. Features Deleted with Discordance Test

3	11	15	16	17
28	29	31	39	44
64	82	85		

C. Highly Correlated Features  
(Correlation  $\geq 0.45$ , Frequency  $\geq 4$ )

19	20	21	22	30
35	47	54	56	58
68	69	74	75	79
83	84			

D. Features Merged

31, 9	(55, 12)*	30, 18	(57, 41)
34, 33	(61, 60)	35, 33	(67, 65)
40, 38	(72, 70)	34, 33	(67*, 66)
10, 9	(14, 13)	4, 3	(5, 4)

+ Corresponding mapping to original features id's.

\* The feature was new modified feature.

---

Table 7.8.3

CEL - Problem: Feat -F and Feat-S  
Feature Ranking (Fisher and Pseudo Similarity)

Feat - F				Feat - S			
Fisher's		Rank		Pseudo Similarity Rank			
Rank	Feat. Id	Rank	Feat. Id	Rank	Feat. Id	Rank	Feat. Id
1	33	21	4	1	36	21	2
2	36	22	13	2	27	22	13
3	34	23	31	3	15	23	4
4	27	24	35	4	16	24	37
5	16	25	37	5	5	25	31
6	25	26	2	6	34	26	14
7	15	27	28	7	24	27	7
8	1	28	32	8	33	28	35
9	19	29	12	9	19	29	28
10	26	30	7	10	1	30	8
11	20	31	14	11	26	31	40
12	5	32	22	12	38	32	30
13	3	33	40	13	3	33	11
14	38	34	9	14	25	34	22
15	24	35	29	15	18	35	9
16	23	36	8	16	23	36	39
17	21	37	30	17	10	37	17
18	6	38	39	18	20	38	6
19	18	39	17	19	12	39	32
20	10	40	11	20	21	40	29

### 7.8.1 Experiment A - MDC

The performance of MDC on the CEL data design set using its abridged feature set, Feat-A = 40 was evaluated. Table 7.8.A1 shows the performance using MDC-E unweighed classifier. Using the weight of  $1/s_i$  the performance of classifier on the same data reached above 91% mark. These results are shown in Table 7.8.A2. The last experiment was repeated on the testing set and over 79.3% of the data set was correctly recognized (see Table 7.8.A3). We repeated the experiment using MDC-M classifier and over 82% performance with classes, 'sae', 'sai', and 'sas' reaching 100% performance and the classes 'sac', 'sag',

'sah', 'sal', 'san', 'sao', 'sap, and 'sar' giving above 85% correct recognition.

Table 7.8.A1

CEL Problem : Design Set

Classification Results - MDC-E  
Euclidean Distance with Feat-A = 40

Input Class	O u t p u t					C l a s s					Recog.
	saa sak	sab sal	sac sam	sad san	sae sao	saf sap	sag saq	sah sar	sai sas	saj %	
saa	8	3	1	0	0	0	0	0	0	0	53.33
	0	0	1	0	0	0	0	1	1		
sab	0	7	2	0	4	0	0	0	0	0	46.67
	0	0	0	0	0	0	0	0	2		
sac	0	3	9	0	0	0	0	0	0	0	60.00
	0	0	0	0	0	0	0	0	3		
sad	0	0	2	11	0	0	0	0	0	0	73.33
	0	0	0	0	0	1	0	0	1		
sae	0	3	0	1	0	0	0	0	0	0	0.00
	0	0	1	1	0	0	1	0	0		
saf	0	0	0	0	0	3	2	0	1	0	30.00
	1	0	0	0	0	3	0	0	0		
sag	0	0	0	0	0	0	14	0	0	0	87.50
	0	0	0	1	0	0	0	0	1		
sah	0	0	0	0	0	0	0	9	0	3	60.00
	2	0	0	0	0	1	0	0	0		
sai	0	0	0	0	0	0	0	0	4	0	80.00
	1	0	0	0	0	0	0	0	0		
saj	0	0	0	0	0	0	0	1	1	6	60.00
	1	0	0	0	0	1	0	0	0		
sak	0	0	0	0	0	0	0	0	4	1	66.67
	10	0	0	0	0	0	0	0	0		
sal	0	0	0	0	0	1	0	0	1	0	86.67
	0	13	0	0	0	0	0	0	0		
sam	0	4	3	1	3	0	0	0	0	0	26.67
	0	0	4	0	0	0	0	0	0		
san	0	0	0	0	0	0	0	0	0	0	90.00
	1	0	0	9	0	0	0	0	0		
sao	0	0	0	0	0	0	0	0	1	1	80.00
	0	0	0	0	12	1	0	0	0		
sap	0	0	0	0	0	0	0	0	1	0	66.67
	2	0	0	1	1	10	0	0	0		
saq	0	0	0	0	2	0	0	0	0	0	50.00
	0	0	0	3	0	0	5	0	0		
sar	0	0	0	0	0	0	0	0	0	0	100.00
	0	0	0	0	0	0	0	10	0		
sas	0	0	0	0	0	1	1	0	0	0	70.00
	0	0	0	1	0	0	0	0	7		
Total Recog./Misrecog./Overall Ave.						151,	87				63.45

Table 7.8.A2

CEL Problem : Design Set

Classification Results - MDC-E  
 Weighted Euclidean Distance with Feat-A = 40  
 Weight Used:1/(sd of feat for class i)

Input Class	O u t p u t						C l a s s			R e c o g .	
	saa sak	sab sal	sac sam	sad san	sae sao	saf sap	sag saq	sah sar	sai sas	saj t	
saa	14	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	1	0		93.33
sab	0	13	0	0	2	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0		86.67
sac	0	0	13	2	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0		86.67
sad	0	0	3	12	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0		80.00
sae	0	0	0	0	7	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0		100.00
saf	0	0	0	0	0	8	0	0	0	0	
	0	0	0	0	0	2	0	0	0		80.00
sag	0	0	0	0	0	0	15	0	0	0	
	0	0	0	0	0	0	0	0	1		93.75
sah	0	0	0	0	0	0	0	15	0	0	
	0	0	0	0	0	0	0	0	0		100.00
sai	0	0	0	0	0	0	0	0	5	0	
	0	0	0	0	0	0	0	0	0		100.00
saj	0	0	0	0	0	0	0	0	0	9	
	0	0	0	0	0	1	0	0	0		90.00
sak	0	0	0	0	0	0	0	0	0	1	
	12	2	0	0	0	0	0	0	0		80.00
sal	0	0	0	0	0	0	0	0	0	0	
	0	14	0	0	0	1	0	0	0		93.33
sam	0	0	0	0	0	0	0	0	0	0	
	0	0	14	0	0	0	1	0	0		93.33
san	0	0	0	0	0	0	0	0	0	0	
	1	0	0	8	0	0	1	0	0		80.00
sao	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	15	0	0	0	0		100.00
sap	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	15	0	0	0		100.00
saq	0	0	0	0	1	0	0	0	0	0	
	0	0	0	0	0	0	9	0	0		90.00
sar	1	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	9	0		90.00
sas	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	10		100.00
Total Recog./Misrecog./Overall Ave.						217,	21				91.18

Table 7.8.A3

CEL Problem : Testing Set

Classification Results - MDC-E  
 Weighted Euclidean Distance with Feat-A = 40  
 Weight Used:1/(sd of feat for class i)

Input Class	O u t p u t						C l a s s				Recog.	
	saa sak	sab sal	sac sam	sad san	sae sao	saf sap	sag saq	sah sar	sai sas	saj	%	
saa	10	2	1	0	0	0	0	0	0	0	71.43	
	0	0	0	0	0	0	0	1	0	0		
sab	0	12	1	0	2	0	0	0	0	0	80.00	
	0	0	0	0	0	0	0	0	0	0		
sac	0	0	13	1	0	0	0	0	0	0	86.67	
	0	0	0	0	0	0	0	0	1	0		
sad	0	0	3	11	0	0	0	0	0	0	73.33	
	0	0	0	0	0	0	0	0	1	0		
sae	0	2	0	0	3	0	0	0	0	0	60.00	
	0	0	0	0	0	0	0	0	0	0		
saf	0	0	0	0	0	4	0	0	0	0	66.67	
	0	0	0	0	0	2	0	0	0	0		
sag	0	0	0	0	0	0	15	0	0	0	93.75	
	0	0	0	0	0	0	0	0	1	0		
sah	0	0	0	0	0	0	0	13	0	0	86.67	
	2	0	0	0	0	0	0	0	0	0		
sai	0	0	0	0	0	0	0	0	2	0	100.00	
	0	0	0	0	0	0	0	0	0	0		
saj	0	0	0	0	0	1	0	0	0	7	77.78	
	0	0	0	0	0	1	0	0	0	0		
sak	0	0	0	0	0	0	0	0	2	1	71.43	
	10	1	0	0	0	0	0	0	0	0		
sal	0	0	0	0	0	1	0	0	1	0	80.00	
	0	12	0	0	0	1	0	0	0	0		
sam	0	2	1	1	0	0	0	0	0	0	60.00	
	0	0	9	0	0	0	2	0	0	0		
san	0	0	0	0	0	0	0	0	0	0	77.78	
	1	0	0	7	0	0	1	0	0	0		
sao	0	0	0	0	0	0	0	0	1	0	85.71	
	0	0	0	0	12	1	0	0	0	0		
sap	0	0	0	0	0	0	0	0	0	0	86.67	
	1	0	0	1	0	13	0	0	0	0		
saq	0	0	0	0	1	0	0	0	0	0	60.00	
	0	0	0	3	0	0	6	0	0	0		
sar	1	0	0	0	0	0	0	0	0	0	88.89	
	0	0	0	0	0	0	0	8	0	0		
sas	0	0	0	0	0	0	0	0	0	0	100.00	
	0	0	0	0	0	0	0	0	10	0		
Total Recog./Misrecog./Overall Ave.						177,	46					79.37

Table 7.8.A4

CEL Problem : Testing Set

Classification Results - MDC-M

Feature Set Feat-A = 40

Input Class	O u t p u t					C l a s s				Recog. %	
	saa sak	sab sal	sac sam	sad san	sae sao	saf sap	sag saq	sah sar	sai sas	saj	
saa	11	2	1	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	78.57
sab	0	12	1	0	2	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	80.00
sac	0	0	13	2	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	86.67
sad	0	1	2	11	1	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	73.33
sae	0	0	0	0	5	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	100.00
saf	0	0	0	0	0	5	0	0	0	0	
	0	0	0	0	0	1	0	0	0	0	83.33
sag	0	0	0	1	0	0	14	0	0	0	
	0	0	0	0	0	0	0	0	1	0	87.50
sah	0	0	0	0	0	0	0	13	0	0	
	2	0	0	0	0	0	0	0	0	0	86.67
sai	0	0	0	0	0	0	0	0	2	0	
	0	0	0	0	0	0	0	0	0	0	100.00
saj	0	0	0	0	0	1	0	0	0	7	
	0	0	0	0	0	1	0	0	0	0	77.78
sak	0	0	0	0	0	0	0	0	2	1	
	10	1	0	0	0	0	0	0	0	0	71.43
sal	0	0	0	0	0	0	0	0	1	0	
	0	13	0	0	0	1	0	0	0	0	86.67
sam	0	2	1	1	0	0	0	0	0	0	
	0	0	9	0	0	0	2	0	0	0	60.00
san	0	0	0	0	0	0	0	0	0	0	
	0	0	0	8	0	0	1	0	0	0	88.89
sao	0	0	0	0	0	0	0	0	1	0	
	0	0	0	1	12	0	0	0	0	0	85.71
sap	0	0	0	0	0	0	0	0	0	0	
	1	0	0	0	1	13	0	0	0	0	86.67
saq	0	0	0	0	0	0	0	0	0	0	
	0	0	0	2	0	0	7	0	0	0	70.00
sar	1	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	8	1	0	88.89
sas	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	10	0	100.00
Total Recog./Misrecog./Overall Ave.						183,	40				82.06

### 7.8.2 Experiment B - QDC

The performance of LDC on the CEL data was evaluated as well, average recognition of which was below 65% and hence the results are not reported here. Obviously QDC was an alternate choice. The performance on testing set using the abridged feature set, Feat-A = 40 was evaluated. Table 7.8.B1 shows the performance of this classifier. The classes 'sag', 'sai', 'sar', and 'sas' were recognizable 100% whereas other classes reached as high as 93.33% producing an overall average of 86.1%. The lower performance is reported by 'saj', 'sam', and 'saq', respectively giving 66.67%, 73.33%, and 70%. Using the feature set Feat-F=21 the performance of classifier on the same data reached above 88.34% mark. These results are shown in Table 7.8.B2. Note that better performance is achieved using only 21 ranked features. The last experiment was repeated on the testing set using ranked features Feat-S=21 producing an average of over 86.55%. These results are reported in Table 7.8.B3. Reviewing the results of Table 7.8.B2 and Table 7.8.B3, it is obvious that Fisher ranked features produced slightly better results as compared with the Pseudo-Similarity ranked features. These observations can be explained with the fact that the CEL-data had 19 classes of variable representation and that significant amount of variations exist between classes, hence Fisher's discriminant index performed a little better.



Table 7.8.B1

CEL Problem : Testing Set

Classification Results - QDC  
(Feature Set Feat-A = 40)

Input Class	O u t p u t					C l a s s			R e c o g .	
	saa sak	sab sal	sac sam	sad san	sae sao	saf sap	sag saq	sah sar	sai sas	saj %
saa	13	1	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	92.86
sab	0	13	1	0	1	0	0	0	0	
	0	0	0	0	0	0	0	0	0	86.67
sac	0	1	14	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	93.33
sad	0	1	2	12	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	80.00
sae	0	1	0	0	4	0	0	0	0	
	0	0	0	0	0	0	0	0	0	80.00
saf	0	0	0	0	0	5	1	0	0	
	0	0	0	0	0	0	0	0	0	83.33
sag	0	0	0	0	0	0	16	0	0	
	0	0	0	0	0	0	0	0	0	100.00
sah	0	0	0	0	0	0	0	14	0	
	1	0	0	0	0	0	0	0	0	93.33
sai	0	0	0	0	0	0	0	0	2	
	0	0	0	0	0	0	0	0	0	100.00
saj	0	0	0	0	0	2	0	0	0	6
	0	0	0	0	0	1	0	0	0	66.67
sak	0	0	0	0	0	1	0	0	1	1
	11	0	0	0	0	0	0	0	0	78.57
sal	0	0	0	0	0	0	0	0	1	0
	0	13	0	0	0	1	0	0	0	86.67
sam	0	1	1	0	0	0	0	0	0	0
	0	0	11	0	0	0	2	0	0	73.33
san	0	0	0	0	0	0	0	0	0	0
	0	0	0	8	0	0	1	0	0	88.89
sao	0	0	0	0	0	0	0	0	1	0
	0	0	0	1	12	0	0	0	0	85.71
sap	0	0	0	0	0	0	0	0	0	0
	1	0	0	1	0	13	0	0	0	86.67
saq	0	0	0	0	1	0	0	0	0	0
	0	0	0	2	0	0	7	0	0	70.00
sar	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	9	0	100.00
sas	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	10	100.00
Total Recog./Misrecog./Overall Ave.						192,	31			86.10

Table 7.8.B2

CEL Problem : Testing Set

Classification Results - QDC  
(Feature Set Feat-F = 21)

Input Class	O u t p u t						C l a s s				Recog. %
	saa sak	sab sal	sac sam	sad san	sae sao	saf sap	sag saq	sah sar	sai sas	saj	
saa	14	0	1	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	100.00
sab	0	13	1	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	1	0	86.67
sac	0	0	15	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	100.00
sad	0	1	1	12	1	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	80.00
sae	0	0	0	0	5	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	100.00
saf	0	0	0	0	0	4	1	0	1	0	
	0	0	0	0	0	0	0	0	0	0	66.67
sag	0	0	0	0	0	0	15	0	0	0	
	0	0	0	0	0	0	0	0	1	0	93.75
sah	0	0	0	0	0	0	0	14	0	0	
	0	0	0	0	0	1	0	0	0	0	93.33
sai	0	0	0	0	0	0	0	0	2	0	
	0	0	0	0	0	0	0	0	0	0	100.00
saj	0	0	0	0	0	1	0	0	0	8	
	0	0	0	0	0	0	0	0	0	0	88.89
sak	0	0	0	0	0	0	0	0	1	0	
	12	1	0	0	0	0	0	0	0	0	85.71
sal	0	0	0	0	0	1	0	0	1	0	
	0	12	0	0	0	1	0	0	0	0	80.00
sam	0	1	1	1	1	0	0	0	0	0	
	0	0	11	0	0	0	0	0	0	0	73.33
san	0	0	0	0	0	0	0	0	0	0	
	1	0	0	8	0	0	0	0	0	0	88.89
sao	0	0	0	0	0	0	0	0	1	1	
	0	0	0	0	12	0	0	0	0	0	85.71
sap	0	0	0	0	0	0	0	0	0	0	
	1	0	0	1	0	13	0	0	0	0	86.67
saq	0	0	0	0	1	0	0	0	0	0	
	0	0	0	0	0	0	9	0	0	0	90.00
sar	1	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	8	0	0	88.89
sas	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	10	0	100.00
Total Recog./Misrecog./Overall Ave.						197,	26				88.34

Table 7.8.B3

CEL Problem : Testing Set

Classification Results - QDC  
(Feature Set Feat-S = 21)

Input Class	O u t p u t					C l a s s				Recog.	
	saa sak	sab sal	sac sam	sad san	sae sao	saf sap	sag saq	sah sar	sai sas	saj	%
saa	13	1	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	1	0		92.86
sab	0	13	1	0	1	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0		86.67
sac	0	0	14	0	1	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0		93.33
sad	0	0	2	12	1	0	0	0	0	0	
	0	0	0	0	0	0	0	0	1		80.00
sae	0	2	0	0	3	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0		60.00
saf	0	0	0	0	0	4	1	0	0	0	
	0	0	0	0	0	1	0	0	0		66.67
sag	0	0	0	0	0	0	16	0	0	0	
	0	0	0	0	0	0	0	0	0		100.00
sah	0	0	0	0	0	0	0	14	0	1	
	1	0	0	0	0	0	0	0	0		87.50
sai	0	0	0	0	0	0	0	0	2	0	
	0	0	0	0	0	0	0	0	0		100.00
saj	0	0	0	0	0	0	0	0	1	8	
	0	0	0	0	0	0	0	0	0		88.89
sak	0	0	0	0	0	0	0	0	2	1	
	12	0	0	0	0	0	0	0	0		80.00
sal	0	0	0	0	0	0	0	0	1	0	
	0	12	0	0	0	1	0	0	0		85.72
sam	0	2	1	0	1	0	0	0	0	0	
	0	0	9	0	0	0	2	0	0		60.00
san	0	0	0	0	0	0	0	0	0	0	
	1	0	0	8	0	0	0	0	0		88.89
sao	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	13	1	0	0	0		85.71
sap	0	0	0	0	0	0	0	0	0	0	
	1	0	0	0	0	14	0	0	0		93.33
saq	0	0	0	0	1	0	0	0	0	0	
	0	0	0	2	0	0	7	0	0		77.78
sar	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	9	0		100.00
sas	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	10		100.00
Total Recog./Misrecog./Overall Ave.						193,	30				86.55

### 7.8.3 Comments: Performance on CEL Data

In the medical literature this problem is dealt with in two different ways, 1) using signals the cells transmit, and 2) using their electromagnetic images. Several examples are present for both approaches [ANBA-87, COHE-86b, GEVI-87, SPEK-87]. In some of these examples simple image processing and pattern recognition methods such as traditional syntactic approaches, Bayesian classifier, discriminant analysis (see Chapter 6 for review and references on classification algorithms), and clustering were applied (see Chapter 3 for review and references on Clustering algorithms) and several reasonable results have been reported [CIAC-93, GEVI-87, UMBA-93, VEKL-93]. None of these examples presented any approach similar to ours. However, any comparison would not be justifiable unless we know the nature and nomenclature of the problem. Since we did not have any more knowledge than what is described in previous sections we were unable to compare our results with any of the published results. However, a recap on the results obtained is presented below.

The CEL problem was a 19 class problem having variable number of samples in various classes with 7 to 32 samples and only 461 samples altogether. Each sample was represented by 85 features. The data in the testing set had only 223 samples ranging from 2 to 16 samples in various classes. Using QDC classifier on all 19 classes simultaneously and their optimal feature sets Feat-F=21 and Feat-S=21 (see Table 7.8.3) a high recognition of 88.34% and 86.55% respectively was achieved. The application of such a large problem with so many variants is almost non-existent in the literature and as such we were unable to find even a conceptually similar example to compare our results with.

## 7.9 Discussion: Overall Recognition Performance

The objective of the present study was to, 1) synthesize and analyze the available waveform (or spectral) information and extract appropriate problem-solving knowledge; 2) select an optimal knowledge representation and its hierarchical organization scheme so that the pattern classes and the knowledge pertaining to their inherent characteristics can be naturally clustered; 3) extract an optimal set of features for a group of classes at each node; 4) store a number of parametric and non-parametric classifiers so that the inference mechanism, based on the parameters saved in the knowledge frame at each node of the tree, can select an appropriate classifier.

The tools and techniques we presented and demonstrated their performance in four different areas of applications fully meet the objectives stated in items 1 through 4. To convert these tools into an operational system a Human/Machine interface providing a communication and explanation facility would be desired which can be added to the system as a black box.

The recognition results on the EEG and PNA data were extremely good. Although the PNA problem was a 20 class problem, PAH classifier was able to give a perfect performance. Most of the algorithms produced excellent results on EEG-data and CEL-data. In the case of NDT data, for large defect and no-defect classes the results were generally between 90% and 100%. It was mainly the small defect classes which have lowered the overall performance on NDT data. Some of the poor results can be attributed to a low signal to noise ratio.

The approach that we have proposed is a comprehensive one in which all phases of a classification/interpretation system are addressed in order to develop an optimal classifier design. It either eliminates or minimizes the possibilities of human

biases at every step of decision making, thus realistically, making it an autonomous and more robust system. However, there is room for improvement in each major phase of the system. For example, a priori information about the experimental conditions and the source input can be used to adjust the preprocessing parameters so that the quality of data is further enhanced. In addition, for knowledge organization, a binary hierarchy is used, for even larger problems an n-ary hierarchy can be used to further minimize the overheads. Several classifiers are independently used; perhaps a hybrid of classifiers may provide better classification. Additional discussion on future extension is postponed until Chapter 8.

A comprehensive set of methods and algorithms were developed for various stages of classification with each giving different performance which suggests that there is no single best method to solve all the problems. Therefore one should not blindly choose one method or the other. Data analysis is critically important to achieve good results. In fact, the characteristics of the data should guide the feature selection process, knowledge organization scheme, and classification algorithm to be used. In choosing the features one should not involve individual preferences and personal biases; rather he/she should acquire as much information as conveniently possible and let the system identify the best parameters. Particularly, in situations where no information on source input is available, the data characteristics are the only means to guide the classification process. We developed rules and methods to automatically examine the data characteristics. In several instances we used existing methods. Several of these methods were used in different perspectives. For example, a discordance test to eliminate least varying features was developed using a simple t-test. Outliers in the data were identified using a z-test; correlated features were removed/merged using a collinearity test. We provided an order and

structure to the entire classification process. We used several classification algorithms and identified the behavior of different algorithms and established several rules to determine the situation where they can be used efficiently. We observed that non-parametric type of classifiers perform excellently well on homostat type of data. For data with significant variations (i.e., segregate type of data) QDC would be a better choice. If the amount of data is statistically sufficient and variations within classes exist BYC is an appropriate choice (see Table 6.3). PAH turned out to be the overall best classifier and is particularly useful for large data sets and class size problems [SIDD-94a,b].

We evaluated the performance of different classifiers using three metrics: sensitivity, predictive accuracy, and false positive rate. We defined these metrics as:

$$\text{Sensitivity (Sen)} = TP / (TP + FN)$$

$$\text{Predictive Accuracy (Pac)} = TP / (TP + FP)$$

$$\text{False Positive Rate (FPR)} = FP / TS$$

where

TP = Total number of True Positive, i.e., correctly classified events.

FN = False Negative, i.e., number of samples missed.

FP = False positive, i.e., number of samples misclassified

TS = Total number of input Samples.

Sensitivity measures the percentage of events that were correctly detected. Predictive accuracy measures the accuracy in correctly classifying the detected events and the False Positive Rate (FPR) measures the rate of misrecognition. Using these metrics the performance of classifiers on all four data

sets using various feature sets was measured; detailed results have already been reported in sections 7.5 through 7.8. Table 7.9.1 summarizes the performance of the classifier on 4 data sets.

All the classifiers except KNN were highly sensitive to the pattern classes from each problem domain with 91.5% predictive accuracy on NDT data when it is designed as a 4-class problem. QDC with Fisher ranked features gave satisfactory results with 84% correct predictive accuracy and 16% false positive rate. A predictive accuracy of above 95% was achieved on EEG data with only 4.3% FPR when we used the pseudo-similarity ranked features and PAH classifier. The same classifier on PNA data achieved perfect results. Using two different feature sets and MDC-E based PAH classifier obtained above 90% performance with 9.3% FPR. On CEL data only a few experiments were conducted achieving the best performance of 88.3% from QDC classifier giving 11.7% false positive rate.

A good performance of classification algorithms on data sets from different problem domains shows that the system developed is truly a generic and versatile classification system which automatically prunes the features and selects the best ones without caring for their identity and the nature of problem, developing a hierarchy of associated pattern classes to organize the knowledge and finally classifying the unknowns using a data-directed classification process choosing an appropriate algorithm at each stage of the process.



Table 7.9.1

## Sensitivity of Classifiers on Different Data Sets

Data Set	Feat. Set	Method	Total Sample (TS)	Positive		Neg. False (FN)	Sens. Rate % (Sen)	False PR % (FPR)	Pred. Acc. % (Pac)
				True (TP)	False (FP)				
NDT	Feat-A	MDC-M	200	165	35	0	100	17.5	82.5
	Feat-A	KNN-E	200	106	31	63	62.7	15.5	77.4
	Feat-F	QDC	200	168	32	0	100	16.0	84.0
	Feat-S	QDC	200	163	37	0	100	18.5	81.5
	Feat-F	BYC	200	156	44	0	100	22.0	78.0
NDT-4	Feat-S	PAH	200	183	17	0	100	8.5	91.5
EEG	Feat-A	MDC-E	300	243	57	0	100	19.0	81.0
	Feat-A	MDC-M	300	260	40	0	100	13.3	86.7
	Feat-F	MDC-E	300	254	46	0	100	15.3	84.7
	Feat-S	MDC-E	300	274	26	0	100	8.7	91.3
	Feat-A	KNN-M	300	273	23	4	98.6	1.3	92.2
	Feat-A	QDC	300	266	34	0	100	11.3	88.7
	Feat-F	QDC	300	266	34	0	100	11.3	88.7
	Feat-S	QDC	300	270	30	0	100	10.0	90.0
	Feat-S	BYC	300	272	28	0	100	9.3	90.7
	Feat-S	PAH	300	287	13	0	100	4.3	95.7
	Feat-FS	PAH	300	272	28	0	100	9.3	90.7
PNA	Feat-A	PAH	100	100	0	0	100	0.0	100.0
CEL	Feat-A	MDC-E	223	177	46	0	100	20.6	79.4
	Feat-A	MDC-M	223	183	40	0	100	17.9	82.1
	Feat-A	QDC	223	192	31	0	100	13.9	86.1
	Feat-F	QDC	223	197	26	0	100	11.7	88.3
	Feat-S	QDC	223	193	30	0	100	13.4	86.6

## **Chapter 8**

### **Performance Review, Directions for Further Research, and Conclusions**

#### **8.1 Introduction**

This study is a unique attempt to incorporate physical observations, analytical knowledge, empirical methods, and rules of interpretation along with signal processing and pattern recognition techniques. A number of new ideas and algorithms are introduced at different stages of the development (see Section 8.2). Along the way a number of existing algorithms have also been explored and used with appropriate adjustments to conform with our system design philosophy and performance objectives.

In this chapter we will critically review the performance of each of the components and the techniques used at various stages. Several areas are identified where a better approach could have been used. Directions for future extensions, improvements and suggestions for further research are presented in the following sections. These directions should be viewed in the light of an overall scheme and philosophy we have developed. If someone is interested in simulating these components, this person could learn from our experience and instead of duplicating the algorithms, perhaps make further development based on the concepts we have presented.

#### **8.2 Performance Review**

The feature selection, knowledge organization and classification algorithms were the major techniques and components of the recognition system that were outlined in Chapter 2. These components are developed and successfully tested on different problems of varied nature. The following sections review

closely the individual components and their performance on data collected from four real-life applications. Wherever discovered, the suggestions for improvements and possible extensions are included as well.

#### **8.2.1 System Concept Level**

The thesis has demonstrated the need for a flexible set of tools in a generic systems approach to statistical pattern classifier design and development. It promoted the concept of pattern association hierarchy (PAH) as the key structure for a knowledge based intelligent recognition system. Important tools developed include feature elimination and selection algorithms, PAH tree building methods, classification algorithms, and the structure and the ways the knowledge is to be organized as presented in this thesis.

#### **8.2.2 Feature Extraction and Selection Level**

Since the data on three of the four problems was provided in the form of a large set of features without any knowledge about the source of the signals, one of the major problem we were faced with the problem to decide: what features should be selected? Addressing this issue, we suggested that one should not bother with the enormity of the observations, rather he/she should extract as much information as conveniently possible and then remove the poorly performing features using a comprehensive feature elimination scheme, called the Successive Elimination Process. The objective behind this approach was to limit the human biases and let the system choose appropriate features based on the characteristics of the data. Furthermore the idea was to develop algorithms that are computationally inexpensive but efficient in performance and to use the inherent characteristics of the data to guide the feature elimination process. A new feature ranking and selection

algorithm called Pseudo-Similarity algorithm [SIDD-90a] based on linear (function) selection criterion is developed. This algorithm meets the objectives specified and it primarily maximizes the inter-class variations. The performance of this algorithm is compared with the Fisher's Discriminant ranking and turned out to be far less expensive computationally with comparatively similar results. Extensions to Pseudo-Similarity algorithm can be done by transforming it to a two-tier algorithm. One step would minimize intra-class variations while the other step checks (maximizes) its usefulness to differentiate between classes. This extension would, however, increase the cost.

### **8.2.3 Knowledge and Knowledge Representation**

We emphasized the use of physical observations and the information derived therefrom as the prime source for acquiring knowledge. In addition statistical observations and tools from statistical decision theory are used to emulate heuristics and human judgement. Although such complete reliance on statistics has solved most of the problems we encountered, we cannot deny the need for pure heuristics, i.e., human judgement. This avenue should be investigated thoroughly and perhaps a simple judgmental call may eliminate hundreds of statistical computations.

### **8.2.4 Knowledge Organization Level**

To further minimize the problem of information or knowledge explosion and redundancy, a new concept called the Pattern Association Hierarchy is introduced to organize the knowledge. This concept, without imposing any order of itself, uses the association that naturally exists among pattern classes to structure the knowledge. A new clustering algorithm called Generalized Variations Method is also developed to hierarchi-

cally organize the patterns and their associated knowledge. This method is developed for situations where we may have too large intra-class variations and too little information to separate between classes. Large variations between NDT pattern classes were instrumental in the creation of this algorithm. Although, the concept was very useful for knowledge organization and pattern classification, additional improvements to the PAH concept can be made. For example, the binary PAH-tree can be transformed to a  $k_i$ -ary PAH tree with  $i=2, \dots, u$ , branches at various internal nodes which would definitely bring more structure to the knowledge organization. In addition it would speed up the classification process as more classes would be eliminated at each node of the tree. But this approach will complicate the tree building method as well as the search strategy as  $k_i$  classes have to be compared with at each internal node.

The knowledge pertaining to pattern groups (or classes) at each non-terminal node is stored in frames, other schemes to structure the information such as scripts and semantic nets may be investigated and an optimal nodal knowledge organization scheme may be selected.

#### **8.2.5 Classification Level**

Numerous classification algorithms have been implemented along with several decision criteria and various weighing schemes. The objective was to choose an algorithm for the purpose it can serve best and not to impose any burden the algorithm cannot handle. A number of observation-dependent parameters are designed to automatically determine the population parameters which in turn determine the pattern classification algorithm to be used. We heavily relied on statistics in developing these parameters, there is definitely room for pure

heuristics and other non statistical methods. Perhaps a hybrid of these approaches may be an alternative as well.

A new classification algorithm based on ID3 algorithm [QUIN-87] is introduced. The algorithm is called Entropy based Discrimination Tree (EDT) which uses the same numeric features as were used for decision-theoretic methods. However, to use the feature in the sense of symbolic features, we developed a transformation scheme to covert them into pseudo-symbolic features. The entropy of each feature is used to construct the decision tree. The EDT's discrimination tree can be expressed in the form of a body of rules and because of this, EDT algorithm can be thought of as an inductive inferencing procedure for machine learning or for optimal rule acquisition and selection.

The concept of pattern association hierarchy along with feature selection methods, data-dependent parameters, a number of parametric and non parametric classifiers at each node of the hierarchy gave birth to a highly flexible PAH-classifier. This arrangement reduces the bias that may be introduced through human judgement. Although the scheme is inductive in nature, it is extremely flexible and intelligent as varied sets of features and decision criteria at each node of the tree can be used. We used empirical knowledge to determine the different transitional steps and for critical decision-making, perhaps the use of expert knowledge or the hybrid of the two may be investigated.

#### **8.2.6 Integration and Automation Level**

We demonstrated several elements of a comprehensive classification system incorporating a number of algorithms at various stages of pattern classification. Instead of human judgement, the system primarily relies on analytical tools to synthesize

the available information. This synthesis evolves a set of new parameters (empirical knowledge and meta knowledge) which are used to partially simulate human judgement. At each step attempts are made to automate the process so that the dependence on the expert can be minimized. For example, based on the characteristics of the data an appropriate algorithm for 1) tree building (PAH), 2) redundant feature elimination and selection, and, 3) a classification algorithm, can be selected. Some intuitive algorithm at each of the above steps may also be examined.

#### **8.2.7 Application Level**

The literature on knowledge-based approaches generally reports specialized and highly problem oriented approaches, this thesis develops algorithms and tools applicable in a generic signal (perhaps, any pattern) classification scheme and successfully applies the algorithms to four problems representing three different application areas, namely, non-destructive testing, spectroscopy (chemometrics) and medical diagnosis. On the last three problems, i.e., NDT, EEG, and CEL classification we were provided with a large set of features.

#### **8.2.8 Overall Efficiency and Effectiveness**

The approach primarily aims at solving large and complex real-life problems with high performance and robustness. This efficiency is introduced by hierarchically segmenting the problem into subproblems of smaller magnitude and in most instances solving a two class problem along with the most suitable feature set supported with appropriate weights and the best possible classification algorithm. We used feature selection and classification algorithms within their normal

constraints. We avoided making unrealistic assumptions just to simplify the magnitude of the problem.

Several levels of classification algorithms have increased the reliability of the system. Presently, only a handful of algorithms are investigated at each node of the tree, in fact any other classifier or hybrid of classifiers can be used, including a neural-net classifier or a genetic algorithm. In fact, additional efficiency can be achieved by introducing multi-tasking by allocating a processor at each node of the tree.

#### **8.2.9 Expansions and Growth**

As for any software, a system's life cycle is based on its flexibility to environment and adaptability to other applications. The components which were intended to be integrated into a general purpose classification and interpretation system are extremely generic and adaptive, the problem dependent knowledge is limited to the extraction of features only. Once the features are extracted every problem application has to run through rest of the stages. With such generic arrangement the system can be applied to new applications.

#### **8.3 Directions for Further Research**

The limit to the imagination of human mind is beyond skies, however, we will attempt not to make this section resemble a script from a novel on science fiction. We will suggest only what can be accomplished in the immediate foreseeable future. We reviewed all the components of the system in the previous section and presented suggestions for their improvement. Incorporating all those suggestions would certainly evolve a highly efficient and generic pattern classification system.



An immediate task would be to develop a fully functional Knowledge Acquisition system and Expert/User Interface and to integrate all the components into one unit system. One of the major steps, that is, the introduction of Reasoning and Explanation System will truly convert the present system to an intelligent expert system that requires the least amount of input from the expert and the same was initially proposed (see Section 1.8). Similar improvements can be incorporated into the inference engine. Additional improvements in each of the major components are described below.

#### **8.3.1 Future: Knowledge Acquisition**

We were supplied with the data in either raw digitized signal form or in the form of feature vectors. In Chapter 2 we had to make several assumptions because the information on experimental conditions and testing conditions was not available. A realistic knowledge acquisition system can be developed by incorporating all the parameters in their own right so that appropriate measures based on nature of data, test equipment, test parameters and the conditions under which the experiment was conducted can be considered and a high quality data is generated. In addition, should one require, data dependent preprocessing methods can be applied to enhance the quality.

#### **8.3.2 Future: Knowledge Formalization and Organization**

Analytical features and empirical knowledge derived from them were the main source of knowledge. There several problem-based parameters may be considered so that problem dependent data processing methods may also be applied. For example, in NDT problems, if the properties of the material under investigation are known appropriate adjustments to improve the quality of emitting signals can be made (see Section 2.3.1 for additional comments). To structure the knowledge the PAH

concept was used. In this concept mainly the hierarchical clustering algorithms were used so that a tree can be developed quickly. The tree so developed may not be an optimal one. Other clustering algorithms, though more expensive, such as partitioning and density matching may be investigated so that an optimally associated tree of pattern classes is developed. We introduced the generalized variations method for tree building (see Section 3.4.2), the major problem with this algorithm is that it is based on the variations among variables of different classes which may produce inconsistent results. Perhaps an algorithm that considers both within and between class variations may produce a better tree.

In addition, we have no way of knowing how individual features or parameters are performing in identifying individual classes. Here the histogram analysis using visual aids can be developed so that one can view the contribution of individual parameters should a subjective analysis of the problem is required.

### **8.3.3 Future: Modeling the Pattern Classes**

We tried to determine the best features to represent a pattern class. Instead one could try to model a pattern class from the available information. The model can be prepared by developing a fuzzy layer around a known kernel. The most representative samples of a class can be used to determine the kernel of a pattern class and the variations among pattern samples of a class could define the fuzzy layer surrounding the kernel. Addressing the problem in this fashion will open various research opportunities.

#### **8.3.4 Future: Inference Engine**

A generic inference engine consists of two sets of algorithms: discrimination, and information theoretic (EDT), both of which are competing with each other. Instead we can use them to complement each other and a hybrid of the two approaches may be developed. At all but the highest level intermediate nodes of the tree, the discrimination system may be used, and at the highest (non-terminal) nodes the EDT algorithm may be applied. At these nodes the discrimination system should provide k best choices it has found, which in turn should be used by the EDT algorithm to identify the single best choice. In all the classification algorithms a predetermined decision function is used. Another improvement would be using a discrimination function that could be automatically designed based on the feature values, individual objectives, and variations among classes at hand. This could be done using similar optimization techniques as were used for feature selection.

Learning from analogy is possible should we intend to apply the system on other unknown signal classification problems. This type of learning applies existing knowledge to a new problem instance on the basis of similarities between them. This involves modifications of the existing knowledge to fit the new case. Learning by analogy is common in human learning.

#### **8.3.5 Future: Expert/User Interface**

The design of the Expert/User interface was briefly discussed in Section 1.8. To complete the system this component has to be developed immediately. Upon implementation exhaustive testing is required to assure the integrity of the system. The suggested design is untested; during the development process several gray areas may emerge that may change the

design objectives. For example, if this phase is to be developed in a GUI environment various functional changes in the design may be inevitable.

#### **8.4. The Research Contributions**

The results reported in Chapter 7 demonstrated the potential of a general approach to an Intelligent Recognition System in several application areas. To successfully achieve the stated objectives and to maximize the functionality of the system, a number of new algorithms and concepts which constitute the contributions of this research were developed. Some of the notable contributions are summarized below.

##### **1. System Concept Level**

Back in 1986 when the work reported in the thesis was originally proposed, the knowledge based approaches to system development were scarce and they were essentially isolated. This project has presented an integrated system approach combining physical observations, empirical and simulated expert knowledge. The thesis has also demonstrated the feasibility of the approach in several applications areas (see Chapters 1 and 7).

##### **2. Knowledge and Knowledge Representation Level**

Instead of solely using the expert/heuristic knowledge or the procedural methods, the emphasis was laid on the physical observations, analytical features and inherent characteristics of the data. Empirical knowledge was derived using statistical methods and analytical knowledge. Data-dependent parameters were designed to automatically determine the population parameters which in turn determine the classification algorithm to be used (see Chapters 2, 3, 4, and 6).

### 3. Feature Extraction/Selection Level

To resolve the feature extraction and selection issues, we suggested to extract all useful features one thinks are essential and then use 'Successive Feature Elimination Process' to weed out the poor performers and later use one of the two feature ranking and selection algorithms, i.e., Pseudo-Similarity algorithm or Fisher's Discriminant ranking. Except the Fisher's Discriminant ranking all feature extraction, elimination, selection, and several weighing schemes described in Section 4.5 are developed in this thesis (see Chapter 4).

### 4. Knowledge Organization Level

To organize the problem solving knowledge, a new concept of pattern association hierarchy (PAH) is introduced wherein several existing algorithms and a new clustering algorithm, called "Generalized Variations" method were used to hierarchically organize the pattern classes and their associated knowledge. This arrangement has reduced the magnitude of information explosion and redundancy problems. Rules were also designed to select a suitable algorithm for knowledge organization (see Chapter 3).

### 5. Classification Level

The new concept of pattern association hierarchy along with feature elimination and selection methods, data-dependent parameters, and, a number of parametric and non parametric classifiers at each node of the hierarchy gave birth to the new flexible PAH-classifier. This arrangement reduces the bias that may be introduced through human judgement while selecting a classification algorithm. In addition a new entropy based classification algorithm called EDT algorithm is also developed (see Chapters 5 and 6).

## 6. Integration and Automation Level

Instead of human judgement, the system relies on analytical tools to synthesize the available information. This synthesis evolves a set of new parameters (empirical knowledge) which are used to partially simulate human judgement. Using empirical knowledge a set of rules are designed to automatically select an appropriate algorithm among several available at different phases of processing. Thus a high level of automation and integration with high recognition performance is achieved.

## 7. Application Level

The thesis develops a generic signal classification scheme by successfully applying the system to four different application areas, namely, non-destructive testing, spectroscopy, EEG classification, and genetic cell classification. In each of the problem area high recognition performance is achieved.

## 8. Size of Problem, Performance, and Robustness

The algorithms we developed are not restrained by the size or the nature of the problem. We solved four problems with 3 to 20 pattern classes, up to 112 features and 2 to 200 samples in a pattern class with consistently high individual class performance of 80% to 100% for various problems.

### 8.5 Conclusions

The goal of the present study was to: 1) synthesize and analyze the available waveform (signal or spectra) information and extracting appropriate problem solving knowledge without imposing human biases towards parameters; 2) integrate domain-dependent expert knowledge and utilize it to find the right

solution; 3) select a data driven scheme for knowledge representation and its hierarchical organization so that the patterns and the knowledge pertaining to the inherent characteristics of pattern classes or groups of classes can be clustered together, 4) extract an optimal set of features for a group of class(es) at each node using either the Fisher ranking or the pseudo-similarity algorithm, and, 5) store a number of parametric and non-parametric classifiers so that the inference mechanism, based on the parameters saved in the knowledge frame at each node of the tree, can select the most appropriate classifier for the application at hand.

The tools we presented and demonstrated their performance on four different problem areas fully meet the objectives defined earlier in Chapter 1. The results on the EEG and PNA data were very good. Although the PNA problem was a 20 class problem PAH classifier was able to give a perfect performance. Most of the algorithms produced excellent results on EEG-data and CEL-data. In case of NDT data, large defect and no-defect classes were generally recognizable at 90% to 100%. It was mainly the small defect classes which had lower performance. This can be attributed to a low signal to noise ratio. Since probes have different sensing ranges, weak signals can be missed even by a very sophisticated device. The potentials of the approach in three problem areas we studied are reviewed in next few sections. Other potential applications of the approach are described in the last section.

#### 8.5.1 NDT Problem

The non-destructive testing (NDT) methods were the prime target for the development of the recognition components. This was the only data on which we had available most of the needed information, including the raw data and hence references to this data were made throughout the thesis. NDT methods

are an integral part of the life-support system of the industrial world. Inspection reliability is of major interest and concern to all the industries. In traditional non-destructive evaluation (NDE) and inspection techniques, one is expected to know the boundary conditions of the problem which require the inspector to know material type, its geometry and type of flaws anticipated to exist. The automation in this area may be able to convert the expert's task to a technician kind of job. The methods we developed demonstrated the successful identification of medium to large size defects of various kinds at varied depth inside the material without having much knowledge about signal source and the equipment used for data acquisition. Use of digital signal processing supplemented with PR tools and knowledge engineering methods has minimized the operator dependence. The use of automated data acquisition and analysis are highly repeatable as compared with manual inspection system. Thus inspection reliability, integrity and robustness are improved significantly.

These automated tools should be incorporated into quantitative NDE programs to create technology based procedures for improving inspection and detection reliability. The problem of inadequate inspection reliability in many critical inspection situations encountered by armed forces, other department of defense agencies, and inspection community in general, is far too important to ignore. Safety in air and space travel can only be guaranteed by automated tools like ours. In the nuclear power industry reason and caution still prevail. A real-time NDT monitoring incorporating our algorithms can bring peace of mind and comfort.

#### **8.5.2 Medical Diagnosis Problem**

The diagnostic procedures in the medical profession are very unstructured and biased. Classification of multidimensional



data in medicine is an important topic since, if successful, it can lead to automated diagnosis or at least provide a tool to speed up and improve diagnosis. The recognition components we developed can be easily transformed into diagnostic and interpretation tools. They will be able to perform equally well if a situation is submitted as a waveform classification problem. A majority of medical phenomena require multitude of variables from a variety of physical conditions. There are applications where nominal or ordinal data may also be interesting. There are cases where important information may be included in patients' symptom records. To accommodate these variables reasonably, algorithms should allow a mixed type of variables without undermining the importance of any variable. Introduction of various data normalizing and feature weighing schemes is an attempt towards that direction.

Although the computing technology has performed many miracles, there is no substitute to human judgement. The prominent applications of this trade of human can be vastly observed in medical diagnosis. To reinforce human judgement we must provide appropriate analytical power and tools for synthesis so that the judgement can be supported by analytical arguments. Several classification methods and PAH structure provide such tools.

We successfully applied the algorithms to the EEG classification problem. Several EEG indications have already been identified in Fig. 2.2 (see Chapter 2). These indications can be easily incorporated as pattern classes while individual elements contaminating the EEG signals can be identified or removed. We applied similar concepts to ECG problem as well and were able to successfully differentiate normal heart beats from ventricular beats higher than 98% on hundreds of samples of data obtained from American Heart Association (AHA data) [SIDD-93b, SIDD-93c].

### **8.5.3 Exploration/Classification of Oils and Minerals**

Ultraviolet-visible (UV-vis) fluorescence has been used for the identification (spectral fingerprinting) and classification of petroleum oils since the early 1970's [SIDD-91a]. The methods established by the American Society for Testing and Materials (ASTM) have been based on this approach. Synchronous spectroscopy was found to produce greater spectral structure and hence increased information contents [EAST-83]. In this approach both excitation and emission monochromators were scanned at a wavelength offset typically between 3 and 25 nm. We presented a more powerful and versatile approach which utilizes more fully the range of similarity parameters and other pattern features available in the spectrum. We applied the system to a new data set of UV-vis synchronous fluorescence spectra of petroleum oils of various origins (both crude and fuel oils) and obtained a perfect performance in identifying 20 different classes of oils. We presented an original approach using pattern recognition and AI to model knowledge and spectral information and classifying unknown oils using advanced classifiers. Potential applications of the system in chemistry and geology particularly in environmental applications are innumerable.

### **8.5.4 General Remarks**

We considered data analysis as a critically important issue to achieve good results. We determined that it should be the characteristics of the data which should dictate the classification algorithm to be used. We developed rules and methods to automatically examine the characteristics of the data. In several instances we used simple statistical methods. For example, using t-test a discordance test was developed to eliminate least varying features. Outliers in the data were identified using a z-test; correlated features were removed/

merged using a colinearity test. We provided an order and structure to the entire classification process. We identified the behavior of different algorithms and established rules to determine the situation where one is suitable.

Since minimal expert knowledge is required, the proposed approach has significantly reduced the time required to acquire and organize the knowledge. Also the major source of knowledge is represented in the form of analytical and empirical knowledge. This arrangement has helped in minimizing the expert biases and providing consistent and robust decision making capability. The hierarchical classification algorithm was able to solve the signal classification problems of larger proportions without deteriorating the recognition performance.

In addition the approach will open further avenues in the research and application of expert systems and knowledge engineering. The use of pattern recognition methods to relieve the expert will not only provide another expert system building tool but also add new dimensions in the fields of both artificial intelligence and pattern recognition.

In fact, the recognition components we developed are generic in behavior and can be used for a variety of applications. The author conjectures that without any major restructure the system should perform equally well as long as a situation is submitted as a signal classification problem and a domain-dependent knowledge base is available. However, the input does not necessarily have to be in the form of signals, since we are using PR techniques minor data dependent modifications can be made in the Facts Gathering phase to include other forms of data. Possible classes of problems include: signal processing, quality assurance and evaluation of materials particularly in nuclear and aviation industry, grains and other food items; medical diagnostic systems; remote sensing,

and environmental applications and chemometric applications such as gas chromatography, mass spectroscopy and infrared spectroscopy.

## REFERENCES

- [ABEN-69] K. Abend, T.J. Hartley Jr., "Comments 'on the Mean Accuracy of Statistical Pattern Recognizers'," IEEE Trans. Info. Theory, Vol. IT-15, May, pp. 420-421 (1969).
- [ADLA-86] K.P. Adlassing, G. Kolarz, W. Scheithauer, G. Grabner, "Approach to a hospital-based application of the medical expert system CADIAG-2," Med. Info., Vol. 11, p. 205 (1986).
- [ADLA-89] K.P. Adlassing, W. Scheithauer, "Performance Evaluation of medical Expert Systems Using ROC Curves," Comp. & Biomed. Res., Vol. 22, pp. 297-313 (1989).
- [AHME-85] P. Ahmed, K.J. Siddiqui, C.Y. Suen, "A Dynamic Approach to Extract Shape Features and Knowledge Organization," 12th IASTED Int. Conf. on Applied Simulation and Modelling, June, Montreal, Canada, pp. 249-252 (1985).
- [AHME-86] P. Ahmed, K.J. Siddiqui, C.Y. Suen, "A Flexible Method to Define, Extract and Organize Shape Features in a Pattern Recognition System," 14th IASTED Int. Conf. on Applied Simulation & Modelling, June, Vancouver, Canada, pp. 557-560 (1986).
- [AIKI-83] J.S. Aikins, J.C. Kunz, E.H. Shortliffe, R.J. Fallat, "PUFF: An Expert System for Interpretation of Pulmonary Function Data," Computers & Biomed. Res., Vol. 16, pp. 199-208 (1983).
- [ALDR-92] L.R. Aldrich, "Advanced Technologies Applied to Work Management," Nuclear Plant J., May/June, pp. 82-86 (1992).
- [ANBA-87] M. Anbar (Ed.), "Computers in Medicine," Comp. Sci. Press, Rockville, MD (1987).
- [ALI-77a] F. Ali, T. Pavlidis, "Description and Recognition of Handwritten Numerals," Proc. Workshop Picture Data Proces. & Mgt., pp. 26-32 (1977).
- [ALI-77b] F. Ali, T. Pavlidis, "Syntactic Recognition of Handwritten Numerals," IEEE Trans. Sys. Man Cybern., Vol. SMC-7, pp. 537-541 (1977).
- [ANDE-73] M.R. Anderberg, "Cluster Analysis For Applications," Academic Press, New York, 1973.

- [ANDR-58] T.W. Anderson, "An Introduction to Multivariate Statistical Analysis," Wiley, New York, (1958).
- [AULD-83] B.A. Auld, A.G. Muennemann, M. Riazat, "Eddy Current Signal Calculations for Surface Breaking Cracks," Proc. of the Review of Progress in Quantitative NDE, Univ. California, Santa Cruz, CA, August (1983).
- [BENB-80] M. Ben-Bassat, R.W. Carlson, V.K. Puri, M.D. Devenport, J.A. Schriver, M. Latif, R. Smith, L.D. Portigal, E.H. Lipnick, M.H. Weil, "Pattern Based Interactive Diagnosis of Multiple Disorders: The MEDAS System," IEEE Trans. Pat. Anal. & Mach. Intel., Vol. PAMI-2, pp. 148-160 (1980).
- [BENN-78] J. Bennet, L. Creary, R.S. Engelmores, R. Melosh "SACON: A Knowledge Based Consultant for Structural Analysis," Report No. STAN-CS-78-699, Comp. Sci. Dept., Stanford Univ., Stanford (1978).
- [BETA-91] A. A.-Betanzos, V. M.-Bonillo, C. A.-Sande, "Fetos: An Expert System for Fetal Assessment," IEEE Trans. BioMed. Engr., Vol. BME-38, No. 2, pp. 199-211 (1991).
- [BIDA-87] H.B. Bidasaria, "Least Desirable Feature Elimination in a General Pattern Recognition Problem," Pat. Recog., Vol. 20, No. 3, pp. 365-370 (1987).
- [BLAC-74] F.W. Blackwell, "Combining Mathematical and Structural Pattern Recognition," Proc. 2nd. Int. Jt. Conf. Pattern Recognition, pp. 78-80 (1974).
- [BRAC-86] R.N. Bracewell, "The Fourier Transform and its Applications," 2nd Edn., Revised, McGraw-Hill, New York (1986).
- [BRAH-85] R.J. Brachman, J.G. Schmolze, "An overview of the KL-ONE Knowledge representation system," Cognitive Sci., Vol. 9, No. 2, Apr., pp. 171-216 (1985).
- [BRAT-78] I. Bratko, "Proving Correctness of Strategies for the AL1 Assertional Language," Info. Proces. Let., Vol. 7, 223-230 (1978).
- [BRAT-80] I. Bratko, D. Michie, "A Representation for Pattern knowledge in Chess End-games," in Advances in Computer Chess 2, M.R.B. Clarke, Ed., Edinburgh Univ. Press, Edinburgh, pp. 31-65 (1980).

- [BROW-88] S. Brown et al., "Chemometrics - Artificial Intelligence," Anal. Chem., Vol. 60, No. 12, p. 267R (1988).
- [BRWN-74] J.S. Brown, R.R. Burton, A.G. Bell, "SOPHIE: A Sophisticated Instructional Environment for Teaching Electronic Troubleshooting (an example of AI in CAI)," Rep. 2790, Bolt Beranek and Newman (1974).
- [BRWN-75] J.S. Brown, R.R. Burton, "Multiple Representation of Knowledge of Tutorial Reasoning," in Representation and Understanding, D.G. Bobrow, A. Collins Eds., Academic Press, New York, pp. 311-349 (1975).
- [BRWR-85] L. Brownston, R. Farrell, E. Kant, N. Martin, "Programming Expert Systems in OPS5," Addison-Wesley, Reading, MA (1985).
- [BUCH-78] B.G. Buchanan, E.A. Feigenbaum, "DENDRAL and Meta-DENDRAL: Their applications dimension," AI Vol. 11, pp. 5-24 (1978).
- [BUCH-79] B.G. Buchanan, "Issues of representation in conveying the scope and limitations of intelligent assistant problems," in Machine Intelligence 9, J.E. Hayes, D. Michie and L.I. Mikulich, Eds., Halsted Press (J. Wiley), New York (1979).
- [BUCH-84] B.G. Buchanan, E.H. Shortliffe, Eds., "Rule-based Expert Systems," Addison-Wesley, Reading, MA (1984).
- [BURT-82] R.R. Burton, J.S. Brown, "An investigation of computer coaching for informal learning activities, Intelligent Tutoring System, D. Sleeman, J.S. Brown, Eds., Academic Press, New York, pp. 79-98 (1982).
- [CAMP-82] A.N. Campbell, V.F. Hollister, R.O. Duda, P.E. Hart, "Recognition of a hidden mineral deposit by an artificial intelligence program," Science, Vol. 217, pp. 927-928 (1982).
- [CARB-70] J.G. Carbonell Sr., "AI in CAI: An AI approach to computer-aided instruction," IEEE Trans. Man-Machine Sys., Vol. 11, pp. 190-202 (1970).
- [CARB-83] J.G. Carbonell, R.S. Michalski, T.M. Mitchell, "An Overview of Machine Learning," in Machine Learning: An AI Approach, R.S. Michalski, J.G. Carbo-

nell, T.M. Mitchell, Eds., Vol II, Morgan Kaufman, Los Altos, CA (1986).

- [CASH-87] G.L. Cash, M. Hatamian, "Optical Character Recognition by the Method of Moments," *Comp. Vision, Grph., & Img. Proc.*, Vol. 39, pp. 291-310 (1987).
- [CATT-66] R.B. Cattell, M.A. Coulter, "Principles of Behavioural Taxonomy and the Mathematical Basis of the Taxonomic Computer Program," *Br. J. Math. Statist. Psychol.*, Vol. 19, pp. 237-269 (1966).
- [CHAN-80] W.Y. Chan, D.R. Hay, C.Y. Suen, O. Schewelb, "Application of Pattern Recognition Techniques in the Identification of Acoustic Emission Signals," *5th Int. Conf. on Pat. Recog.*, Florida, IEEE, pp. 108-111 (1980).
- [CHAN-82] W.Y. Chan, D.R. Hay, "Application of a General Purpose NDT Signal Classifier to Acoustic Emission," *Proc. Conf. Adv. NDE Tech.*, Montreal, pp. 30-36 (1982).
- [CHAN-85a] R.W.Y. Chan, D.R. Hay, V. Caron, M. Hone, R.D. Sharp, "Classification of Acoustic Emission Signals Generated During Welding," *J. Acoustic Emission*, Vol. 4, No. 4, pp. 115-123 (1985).
- [CHAN-85b] R.W.Y. Chan, R.D. Sharp, J.P. Monchalain, J. Busiers, D.R. Hay, "Ultrasonic Defect Sizing by Advanced Pattern Recognition Techniques," *Proc. Review of Progress in Quantitative NDE*, Vol. 4A, D. O. Thompson, D.E. Chimenti, Eds., Plenum Press, New York, pp. 213-223 (1985).
- [CHAN-89] R.W.Y. Chan, D.R. Hay, J.R. Hay, B.H. Patel, "Improving Acoustic Emission Crack/Leak Detection in Pressurized Piping by Pattern Recognition Techniques," *Non-Destructive Testing, Proc. 12th World Conf. on NDT*, J. Boogaard, G.M. van Dijk, Eds., Elsevier, Amsterdam, pp. 851-853 (1989).
- [CHAR-85] E. Charniak, D. McDermott, "Introduction to Artificial Intelligence," Addison-Wesley, Reading, MA (1985).
- [CHAS-88] B. Chandrasekaran, "Generic Tasks as Building Blocks for Knowledge-based Systems: The Diagnosis and Routine Design Examples," *Knowledge Engineering Review*, Vol 3, No. 3, pp. 183-210 (1988).



- [CHAY-73] C.Y. Chang, "Dynamic Programming as Applied to Feature Subset Selection in a Pattern Recognition System," IEEE Trans. Sys. Man Cybern., Vol.SMC-3, March, pp. 197-200 (1973).
- [CHEN-73] C.H. Chen, "Statistical Pattern of Recognition," Hayden, Rochelle Park, NJ (1973).
- [CHEN-82] C.H. Chen, "Statistical pattern Recognition," in Digital Waveform Processing and Recognition, C.H. Chen, Ed., CRC Press, Boca Raton, Florida, pp. 59-74 (1982).
- [CHIE-68] Y.T. Chien, K.S. Fu, "Selection and Ordering of Feature Observations in Pattern Recognition Systems," Inform. Contr., Vol. 12, May-June, pp. 394-414 (1968).
- [CIAC-93] E.J. Ciaccio, S.M. Dunn, M. Akay, "Biosignal Pattern Recognition and Interpretation Systems," IEEE Engineering in Med. & Bio., Vol. 12, No. 3, pp. 89-95 (1993).
- [CLAN-81] W.J. Clancy, R. Letsinger, "NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching," Proc. 7th IJCAI, Vancouver, pp. 829-836 (1981).
- [CLAN-87] W.J. Clancy, "Knowledge-Based Tutoring: The GUIDON Program," MIT Press, Cambridge, MA (1987).
- [CLAR-82] K.L. Clark, F.G. McCabe, "PROLOG: A language for implementing expert systems," In Machine Intelligence, J. Hayes, D. Michie, T. Pao, Eds., J. Wiley, New York, pp. 455-470 (1982).
- [COHE-86a] A. Cohen, "Biomedical Signal Processing: Time and Frequency Domains Analysis," Vol. 1, CRC Press, Boca Raton, Florida (1986).
- [COHE-86b] A. Cohen, "Biomedical Signal Processing: Compression and Automatic Recognition," Vol. 2, CRC Press, Boca Raton, Florida (1986).
- [COOP-84] G.F. Cooper, "NESTER: A medical decision support system that integrates casual, temporal, and probabilistic knowledge," Ph.D. dissertation, Comp. Sci. Dept. Stanford Univ., Stanford (1984).
- [COVE-67] T.M. Cover, P.E. Hart, "Nearest Neighbor Pattern Classification," IEEE Trans. on Info. Theory, Vol. IT-13, No. 1, pp. 21-27 (1967).

- [DATT-80] G.R. Dattareya, V.V.S. Sharma, "Decision Tree Design for Pattern Recognition including Pattern Measurement Cost," Proc. 5th Int. Conf. Pattern Recognition, pp. 1212-1214 (1980).
- [DAVI-81] R. Davis, H. Austin, I. Carlbom, B. Frawley, P. Pruchnik, R. Sneiderman, J.A. Gilreath, "The dip-meter advisor: interpretation of geological signals," Proc. 7th Int. Jt. Conf. on AI, IJCAI- 81, pp. 846-849 (1981).
- [DESR-86] H.L.M. Des Reis, D.M. McFarland, "On the Acousto-Ultrasonic Non-Destructive Evaluation of Wire Rope using the Stress Wave Factor Technique," British J. NDT, Vol 28, No. 3, May, pp. 155-156 (1986).
- [DEVI-82] P.A. Devijver, J. Kittler, "Pattern Recognition-A Statistical Approach," Prentice Hall International, Englewood Cliffs, NJ (1982).
- [DICA-93] L.A. DiCarlo, D. Lin, J.M. Jenkins, "Automated Interpretation of Cardiac Arrhythmias," J. Electrocardiology, Vol. 26, No. 1, pp. 53-63 (1993).
- [DUDA-66] R.O. Duda, H. Fossum, "Pattern Classification by Iteratively Determined Linear and Piecewise Linear Discriminant Functions," IEEE Trans. Electron. Comput., Vol. C-15, pp. 220-232 (1966).
- [DUDA-73] R.O. Duda, P.E. Hart, "Pattern Classification and Scene Analysis," J. Wiley, New York (1973).
- [DUDA-78a] R. Duda, J. Gaschnig, P.E. Hart, "Model Design in the PROSPECTOR Consultant System for Mineral Exploration," in Expert Systems in the Microelectronic Age, D. Michie, Ed., Edinburgh Univ. Press, Edinburgh, pp. 153-167 (1978).
- [DUDA-78b] R.O. Duda, P.E. Hart, P. Barette, J. Gashing, K. Konolige, R. Reboh, J. Slocum, "Development of the PROPECTOR consultant System for Mineral Exploration," Final Rep., SRI Proj. 5821 and 6415, AI Center, SRI International (1978).
- [DUDA-83] R.O. Duda, E.H. Shortliffe, "Expert Systems Research," Science 220, pp. 261-268 (1983).
- [DUER-80] B. Duerr et al., "A Combination of Statistical and Syntactical Pattern recognition Applied to Classification of Unconstrained Handwritten Numerals," Pattern Recognition Vol. 12, pp.

189-199 (1980).

- [EAST-83] D. Eastwood, "Use of or Luminescence Spectroscopy in Oil Identification," Modern Fluorescence Spectroscopy, D. Eastwood, Ed., American Society for Testing of Materials, STP-822, Philadelphia, PA, ASTM (1983).
- [EAST-91] D. Eastwood, R.L. Lidberg, K.J. Siddiqui, "The Role of Luminescence and Spectral Pattern Recognition in Environmental Programs," in Chemistry for Protection of the Environment, L. Pawlowski, W.J. Lacy, J. Dlugosz, Eds., Plenum Press, London, England, pp. 187-196 (1991).
- [ELSL-83] R.K. Elsley, L.J. Graham, "Identification of Acoustic Emission Sources by Pattern Recognition Techniques," in A Review of Progress in Quantitative NDE," D.O. Thompson and D.E. Chimenti, Eds., Plenum, New York, pp. 471-478 (1983).
- [ESTE-65] S.E. Estes, "Measurement selection for linear discriminants used in pattern classification," San Jose Res. Lab., IBM Res. Rep. RJ-331, April (1965).
- [FAGA-78] L.M. Fagan, "Ventilator Management: a program to provide on-line consultative advice in the intensive care unit", Memo HPP-78-16, Comp. Sci. Dept. Stanford Univ., Stanford (1978).
- [FAGA-80] L. Fagan, "VM: Representing time-dependent relations in clinical setting," Ph. D., Dissertation, Comp. Sci. Dept., Stanford Univ. Stanford (1980).
- [FARI-83] B. Faris, A. Behrouz, "Approximation of Multipath Planer Shapes in Pattern Recognition," Int. J. Comput. & Info. Sci., Vol. 12, No. 2, pp. 99-110 (1983).
- [FEIG-71] E.A. Feignbaum, G.G. Buchanan, J. Laderbert, "On Generality and Problem Solving; A Case Study using DENDRAL Program," in Machine Intelligence, B. Meltzer, D. Michie, Eds., 6, Edinburgh Univ. Press, Edinburgh, pp. 165-190 (1971).
- [FIKE-85] R. Fikes, T. Kehler, "The role of frame-based representation in reasoning," CACM Vol. 28, No. 9, Sept., pp. 904-920 (1985).
- [FINK-89] S.M. Finkelstein, P.L.M. Kerkhof, M. Okada, (Eds.) "Special Issue on Medical Applications of Arti-

ficial Intelligence and Information Systems,"  
IEEE Trans. BioMed. Engr., Vol. BME-36, No. 5  
(1989).

- [FIRS-85] M.B. First, L.J. Soffer, R.A. Miller, "QUICK (Quick Index to Caduceus Knowledge): Using the INTERNIST-1/CADUCEUS knowledge base as an electronic textbook of medicine," Comput. Biomed. Res., Vol 18, p. 137 (1985).
- [FOLE-72] D.H. Foley, "Considerations of Sample and Feature Size," IEEE Trans. Info. Theory, Vol. IT-18, No. 5, p. 618 (1972).
- [FORG-65] E.W. Forgey, "Cluster Analysis of Multivariate data: Efficiency versus Interpretability of Classification," Biometrics, Vol. 21, pp. 768-769 (1965).
- [FORM-77] C.L. Forgy, J. McDermott, "OPS, a domain independent production system language," Proc. 5th AJCAI, Cambridge, MA, Vol. 2, pp. 933-939 (1977).
- [FORO-87] I. Foroutan, J. Sklansky, "Feature Selection for Automatic Classification of Non-Gaussian Data," IEEE Trans. Syst. Man. Cybern., Vol. SMC-17, No. 2, March/April, pp. 187-198 (1987).
- [FUKS-67] K.S. Fu, Y.T. Chien, G.P. Cardillo, "A Dynamic Programming Approach to Sequential Pattern Recognition," IEEE Trans. Electron. Comput., Vol. EC-16, Dec., pp. 790-803 (1967).
- [FUKS-68] K.S. Fu, P.J. Min, "On Feature Selection in Multiclass Pattern Recognition," Purdue Univ., School of Elect. Eng., Lafayette, Tech. Rep. TR-EE 68-17, July (1968).
- [FUKS-82] K.S. Fu, "Syntactic Pattern Recognition and Applications," Prentice-Hall, Englewood Cliffs, NJ (1982).
- [FUKS-86] K.S. Fu, "Syntactic Pattern Recognition," in Handbook of Pattern Recognition and Image Processing, Academic Press, Orlando, FL, pp. 85-117 (1986).
- [FUKU-75] K. Fukunaga, P.M. Narendra, "A Branch and Bound Algorithm for Computing K-Nearest Neighbors," IEEE Trnas. Comput., Vol. C-24, p. 750 (1975).

- [FUKU-89] K. Fukunaga, "Effects of sample size in classifier design," IEEE Trans. Pat. Ana. & Mach Intel. Vol. PAMI-11, pp. 873-885 (1989).
- [FUKU-90] K. Fukunaga, "Introduction to Statistical Pattern Recognition," Academic Press, New York (1990).
- [GALL-86a] S.I. Gallant, "Brittleness and machine learning," Int. Meeting on Advances in Learning, CNRS, Paris, France, July/Aug. (1986).
- [GALL-86b] S.I. Gallant, "Optimal Linear discriminants," Proc. 8th Int. Conf. on PR, Paris, France, Oct., pp 849-852 (1986).
- [GEVI-80] A.S. Gevins, "Pattern Recognition of Human Brain Electrical Potentials," IEEE Trans. Pat. Anal. Mach. Intel., Vol PAMI-2, p. 383 (1980).
- [GEVI-86] A.S. Gevins, "Classifier-Directed Signal Processing in Brain Research," IEEE Trans. BioMed. Engr. Vol. BME-33, pp. 1054-1068 (1986).
- [GEVI-87] A.S. Gevins, A. Remond (Eds.) "Methods of Analysis of Brain Electrical and Magnetic Signals," in EEG Handbook, Revised Series, Vol. 1, Elsevier, Amsterdam (1987).
- [GOME-81] F. Gomez, B. Chandrasekaran, "Knowledge Organization and Distribution for Medical Diagnosis," IEEE Trans. on Sys., Man & Cybern., Vol. SMC-11 No. 1, Jan. (1981).
- [GORA-88] B. Goranzon, I. Josefson (Eds.) "Knowledge, Skill and Artificial Intelligence," Springer-Verlag, London (1988).
- [GRAJ-86] K.A. Grajski, L. Breiman, G.V. di Prisco, W.J. Freeman, "Classification of EEG Spatial Patterns with a Tree-Structured Methodology: CART," IEEE Trans. BioMed. Engr., Vol. BME-33, pp. 1054-1068 (1986).
- [HARR-80] T.P. Harrington, P.G. Doctor, "Acoustic emission analysis using pattern recognition," 5th Int. Conf. on Pat. Recog., Florida, pp. 1204-1207 (1980).
- [HAYD-84] D.R. Hay, R.W.Y. Chan, D. Sharp, K.J. Siddiqui, "Classification of Acoustic Emission Signals from Deformation Mechanism in Aluminum Alloys," J. Acous. Emis., Vol. 3, No. 3, pp. 118-129 (1984).

- [HAYD-87] D.R. Hay, "Private Communication and Discussion," during 1987 - 1992.
- [HAYD-88] D.R. Hay, R.W.Y. Chan, K.J. Siddiqui, J.R. Hay, "Pattern Classification Manual," Course Handbook for the Three Day Course on Engineered Pattern Recognition Systems for Non-Destructive Evaluation, Montreal, May 1988.
- [HAYE-83] F. Hayes-Roth, D.A. Waterman, D.B. Lenat, Eds., "Building Expert Systems," Addison-Wesley, Reading, MA (1983).
- [HEID-88] M.A. Heidari, K.-P. Adlassing, "Preliminary results on CADIAG-2/GALL: A diagnostic consultation system for gallbladder and biliary tract diseases," Proc. Med. Informatics Europe'88, pp.622-666, Springer-Verlag, Berlin (1988).
- [HERN-89] C. Hernandez-Sande, V. M.-Bonillo, A. A.-Betanzos, "ESTER: An Expert System for Management of Respiratory Weaning Therapy," IEEE Trans. BioMed. Engr., Vol. BME-36, No. 5, pp. 559-564 (1989).
- [HSIA-81] T.C. Hsia, "A note on invariant moments in image processing," IEEE Trans., Systems, Man, Cybernet. Vol. SMC-11, No. 12, pp. 831-834 (1981).
- [HUGH-68] G.F. Hughes, "On the Mean Accuracy of Statistical Pattern Recognizers," IEEE Trans. Info. Theory, Vol. 14, p. 55 (1968).
- [HUNT-62] E.B. Hunt, "Concept Learning: An Information Processing Problem," Wiley, New York (1962).
- [IRGO-90] A. Irgon, J. Zolnowski, K. J. Murray, G. Marvin, "Expert System Development: A Retrospective View of Five Systems," IEEE Expert, June, pp. 25-40 (1990).
- [JARD-71] N. Jardine, R. Sibson, "Mathematical Taxonomy," J. Wiley, New York (1971).
- [JACK-90] P. Jackson, "Introduction to Expert Systems," Addison-Wesley, Reading, MA (1990).
- [KANA-68] L.N. Kanal, B. Chandrasekran, "On Dimensionality and Sample Size in Statistical Pattern Recognition," Proc. Natl. Electron. Conf., Oak Brook, Ill. p 2 (1968).

- [KANA-72] L.N. Kanal, B. Chandrasekran, "On Linguistic, Statistical and Mixed Methods for Pattern Recognition," S. Watanabe (Ed.), Academic Press, New York (1972).
- [KANA-74] L. Kanal, "Patterns in Pattern Recognition: 1968-1974," IEEE Trans. on Info. Theory, Vol. IT-20, No. 6, Nov., pp. 697-722 (1974).
- [KATS-69] S. Katsurugi et al., "Recognition of Handwritten Numerals using Decision Graph," Proc. 1st Int. Jt. Conf. Artificial Intelligence, pp. 161-170 (1969).
- [KAUT-86] H.E. Kautz, "Acousto-Ultrasonic Verification of Strength of Filament Wound Composite Material," NASA Technical Memo 88827, July 1986.
- [KILL-81] T.J. Killeen, D. Eastwood, M.S. Hendrick, "Oil-Matching by using a Simple Vector Model for Fluorescence Spectra," Talents, Vol. 28, pp. 1-6 (1981).
- [KOON-75] W.L.G. Koontz, P.M. Narendra, K. Fukunaga, "A branch and bound clustering algorithm," IEEE Trans. Comput., Sept., pp. 908-915 (1975).
- [KOSK-92] B. Kosko, "Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence," Prentice Hall, Englewood Cliffs, NJ, (1992).
- [KRAM-73] H.P. Kramer et al., "The Recognition of Handwritten Symbols and its Relation to Formal Languages," Proc. 1st Int. Jt. Conf. Pattern Recognition, pp. 57-58 (1973).
- [KRAU-69] J. Krautkramer, H. Krautkramer, "Ultrasonic Testing of Materials," Springer-Verlag, New York (1969).
- [KUNZ-78] J. Kunz, "A physiological rule-based system for interpreting pulmonary function test results," Memo HPP 78-19, Comp. Sci. Dept., Stanford Univ., Stanford (1978).
- [LACA-88] V. Lacasse, J.R. Hay, D.R. Hay, "Pattern Recognition of Ultrasonic Signals for Detection of Wall Thinning," NATO ASI Series, Vol. F44, C.H. Chen, Ed., Springer-Verlag, Berlin, pp. 139-198 (1988).
- [LAMB-89] J. Lambert, G. Haralamb, U. Sauer, O.A. Barblian, "Operational Experience in Automatic Ultrasonic

Inspection of Steel Plate," NDT Proc. 12th World Conf., Amsterdam, The Netherlands, April, pp. 871-877 (1989).

- [LANC-67] G.N. Lance, W.T. Williams, "A General Theory of Classificatory Sorting Strategies: 1 Hierarchical Systems," Comput. J., Vol 9, pp. 373-380 (1967).
- [LENA-82] D.B. Lenat, "AM: AI Approach to Discovery in Mathematics as Heuristic Search," in Knowledge-based Systems in AI, R. Davis, D.B. Lenat, Eds., McGraw-Hill, New York (1982).
- [LENA-83] D.B. Lenat, "EURISKO: A program that learns new heuristics and domain concepts," The nature of Heuristics III: program design and results, AI Vol. 21, No. 1,2, pp. 51-99 (1983).
- [LIEB-86] S.A. Liebman, "Artificial Intelligence Applications in Chemistry," T.H. Pierce, B.A. Hohne, Eds. Vol. 306, Am. Chem. Soc., Washington, DC (1986).
- [LIN-80] Y.K. Lin, K.S. Fu, "Automatic Classification of Cervical Cells using a Binary Tree Classifier," Proc. 5th Int. Conf. Pat. Recog., pp. 570-574 (1980).
- [LINC-86] W.-C. Lin, K.S. Fu, "A Syntactic Approach to Three-Dimensional Object Recognition," IEEE Trans. Sys. Man, & Cybern., Vol. SMC-16, No. 3, MAY/June, pp. 405-421 (1986).
- [LIND-80] R.K. Lindsay, B.G. Buchanan, E.A. Feigenbaum, J. Lederberg, "Applications of AI for Organic Chemistry: The DENDRAL Project," McGraw-Hill, New York (1980).
- [LUGE-89] G.F. Luger, W.A. Stubblefield, "Artificial Intelligence and the Design of Expert Systems," Benjamin/Cummings, Redwood City, CA (1989).
- [MAHA-36] P.C. Mahalanobis, "On the Generalized Distance in Statistics, Proc. Nat. Inst. Sci., India, Vol. 122, pp. 49-55 (1936).
- [MAHL-85] S. Mahalingam, D.D. Sharma, "WELDEX - An Expert System for Non-Destructive Testing of Welds," 2nd Conf., AI Appln.: The Engineering of Knowledge Systems, Florida, pp. 572-576 (1985).
- [MARN-77] N. Martin, P. Friedland, J. King, M.J. Stefik, "Knowledge-based management for experiment plan-



- ning in molecular genetics," Proc. 5th Int. Jt. Conf. on AI, IJCAI-77, Pittsburgh (1977).
- [MART-71] W.A. Martin, R.J. Fateman, "The MACSYMA System," Proc. 2nd Symposium on Symbolic and Algebraic Manipulation, Los Angeles, pp. 59-75 (1971).
  - [MARZ-83] J.R. Martinez, A.J. Bahr, "Statistical Detection Model for Eddy-Current Systems," Proc. Review of Progress in Quantitative NDE, Santa Cruz, August 7-12, p. 22 (1983).
  - [MATT-88] J.R. Matthews, "Private Communication and Discussion," (1988).
  - [MATT-89] J.R. Matthews, D.R. Hay, R.W.Y. Chan, "Computer-aided Ultrasonic Inspection of Submarine Pressure," Proc. DARPA QNDE Meeting, Brunswick, Maine, July (1989).
  - [MCDE-81] J. McDermott, B. Steele, "Extending a Knowledge-based system to deal with adhoc constraints," Proc. 7th Int. Jt. Conf. on AI, IJCAI-81, Wm. Kaufmann Inc., pp. 824-828 (1981).
  - [MCDE-82] J. McDermott, "R1: A Rule-based Configuration of Computer Systems," AI 19, pp. 39-88 (1982).
  - [MCGO-61] W.J. McGonnagle, "Nondestructive Testing," McGraw-Hill, New York (1961).
  - [MEIS-68] W.S. Meisel, "Least Square Methods in Abstract Pattern Recognition," Info. Sci., Vol. 1, No. 1, pp. 43-54 (1968).
  - [MICH-82] D. Michie, "Expert Systems," in Machine Intelligence and Related Topics, D. Michie, Ed., Gordon & Breach, London, pp. 195-216 (1982).
  - [MILL-82] R. Miller, H. Pople, J. Myers, "INTERNIST-1: An experimental Computer-based Diagnostic Consultant for General Internal Medicine," New England J. of Medicine, Vol. 307, No. 8, pp. 468-476 (1982).
  - [MILL-84] R. A. Miller, "INTERNIST-1/CADUCES: Problems facing expert consultant programs," Methods Info. Med., Vol. 23, p. 9 (1984).
  - [MOLD-87] S. Moldoveanu, C. Rapson, "Spectral Interpretation for Organic Analysis using an Expert System," Ana. Chem., Vol. 59, No. 8, p. 1207 (1987).

- [MOWR-88] G. Mowrey, "Adaptive Signal Discrimination as applied to Coal Interface Detection," IEEE Conf. Industry Application Society Meeting, pp. 1277-1282 (1988).
- [MOYZ-82] J.A. Moyzis, Jr., D.M. Forney, Jr., "Increased Reliability - A Critical Research Goal," Review of Progress in Quantitative NDE, Vol. 1, D.O. Thompson, D.E. Chimenti, Eds., Plenum Press, New York (1982).
- [MUCC-71] A.N. Mucciardi, E.E. Gose, "A Comparison of Seven Techniques for Choosing Subsets of Pattern Recognition Properties," IEEE Trans. Comp., Vol. C-20, No. 9, Sept., pp. 1023-1031 (1971).
- [NADL-93] M. Nadler, E.P. Smith, "Pattern Recognition Engineering," J. Wiley, New York (1993).
- [NAGA-91] M. Nagamachi, "An Image Technology Expert System and its Application to Design Consultation," Int. J. of Human-Comp. Inter. Vol. 3, No. 3, pp. 267-279 (1991).
- [NAGY-68] G. Nagy, "State of the Art in Pattern Recognition," Proc. IEEE, Vol. 56, May, pp. 836-862 (1968).
- [NARA-69] R. Narasimhan, "On the Description, Generation, and Recognition of Classes of Pictures," in Automatic Interpretation and Classification of Images, A. Grasselli, Ed., Academic Press, New York, pp. 1-42 (1969).
- [NARE-76] P.M. Narendra, K. Fukunaga, "A Branch and Bound Algorithm for Feature Subset Selection," Proc. IEEE Int. Conf. on Cybern. & Society, IEEE, New York pp. 497-503 (1976).
- [NATH-84] M. Nathanson, "Physicians get HELP in cutting patient stay," Modern Healthcare, April, pp. 140-145 (1984).
- [NAUD-83] D.S. Nau, "Expert Computer Systems," IEEE Comp., Feb., pp. 63-85 (1983).
- [NEWE-63] A. Newell, H.A. Simon, "GPS: A Program that Simulates Human Thought," in Computers and Thought, E.A. Feigenbaum, J. Feldman, Eds., McGraw-Hill, New York (1963).

- [NEWE-72] A. Newell, H.A. Simon, "Human Problem Solving," Prentice-Hall, Englewood Cliffs, NJ (1972).
- [NEWE-73] A. Newell, "Production Systems: Models of Control Structures," in Visual Information Processing, W. G. Chase, Ed., Academic Press, New York (1973).
- [NEWE-80] A. Newell, "The Knowledge Level," AI Magazine, Vol. 2, No. 2, Presenditil address, AAAI, Stanford, Aug. (1980).
- [NILS-65] N.J. Nilsson, "Learning Machines: Foundations of Trainable Pattern Classifying Systems," McGraw-Hill, New York (1965).
- [NPJ-92] Nuclear Plant Journal, Vol. 10, No. 3, May/June, (1992).
- [NPJ-93] Nuclear Plant Journal, Vol. 11, No. 6, Nov./Dec., (1993).
- [ORR-79] E.C. Orr, J.R. Gouge, R. Shankar, M.F. Whalen, C.L. Brown, A.N. Mucciardi, "Application of Nonlinear Signal Processing to Pipe and Nozzle Inspection," EPRI Report NP-964, January (1979).
- [PAHL-87] O. Pahlm, L. Sornmo, "Data Processing of Exercise ECG's," IEEE Trans. of Biomed. Eng., Vol. BME-34, No. 2, Feb. pp. 158-164 (1987).
- [PAVE-86] R. Pavelle, "Artificial Intelligence," T.H. Pierce, B.A. Hohne, Eds., ACS Symposium Series 306, Am. Chem. Soc., Washington, DC, p. 106 (1986).
- [PAVL-71] T. Pavlidis, "Linguistic analysis of waveforms," Soft. Engr., J.T. Tou, Ed., Vol. 2, Academic Press, pp. 203-225 (1971).
- [PAVL-73] T. Pavlidis, "Waveform segmentation through functional approximation," IEEE Trans. Comp., Vol. C-22, No. 7, July, pp. 689-697 (1973).
- [PAVL-77] T. Pavlidis, "Structural Pattern Recognition," Springer-Verlag, New York (1977).
- [PAYN-90] E.C. Payne, R.C. McArthur, "Developing Expert Systems: A Knowledge Engineer's Handbook for Rules and Objects," J. Wiley, New York, (1990).
- [PICT-88] T.W. Picton, "Measurement of event-related potentials: signal extraction," in Handbook of Electroencephalography and clinical Neurophysiology-

Human Event-Related Potentials, T.W. Picton, Ed.,  
Revised Edn., Vol 3, Elsevier, pp. 7-44, (1988).

- [POPL-77] H.E. Pople, J.D. Myers, R.A. Miller, "DIALOG: a model of diagnostic logic for internal medicine," Proc. 5th Int. Jt. Conf. on AI, IJCAI-77, Pittsburgh (1977).
- [PRYO-83] T.A. Pryor, R.M. Gardner, P.D. Clayton, H.R. Warner, "The HELP system," J. Med. Syst., Vol. 7, No. 87 (1983).
- [QUIN-83] J.R. Quinlan, "Learning Efficient Classification Procedures and their Application to Chess Endgames," in Machine Learning, R.S. Michalski, J.G. Carbonell, T.M. Mitchell, Eds., Tioga, Palo Alto, CA (1983).
- [QUIN-86] J.R. Quinlan, "Induction of Decision Trees," Machine Learning, Vol. 1, No. 1 (1986).
- [QUIN-87] J.R. Quinlan, Ed., "Applications of Expert Systems," Addison-Wesley, Reading, MA (1987).
- [QUIN-88] J.R. Quinlan, "Simplifying Decision Trees," in Knowledge Acquisition for Knowledge Based Systems, B.R. Caine, J.H. Boose, Eds., Academic Press, New York (1988).
- [RAOP-67] P.S.R. Rao, "On selecting variables for Pattern Classification," Tech. Rep. available from IEEE Comput. Group Repository, 345 East 47 St., New York, N.Y. Dec. (1967).
- [RAUD-79] S.J. Raudys, "Determination of Optimal Dimensionality in Statistical Pattern Classification," Pat. Recog., Vol. 11, No. 4, p. 263 (1979).
- [REBO-81] R. Reboh, "Knowledge engineering techniques in the PROSPECTOR environment," Rep. 243, AI Center, SRI International, Menlo Park, CA (1981).
- [REDD-73] R. Reddy, "The HEARSAY speech understanding system," Third IJCAI, pp. 185-199 (1973).
- [REGG-83] J. Reggia, D. Nau, P. Wang, "Diagnostic Expert Systems Based on a Set Covering Model," Int. J. Man Mach. Stud., Vol 19., pp. 437-460 (1983).
- [RICH-91] E. Rich, K. Knight, "Artificial Intelligence," McGraw-Hill, New York (1991).

- [ROBE-92] B.W. Roberts, L. Smith, D.A. Steinke, G.P. Singh, "Computer Aided maintenance of Power Plant Steamlines Using an information Management System," Proc. Conf. on Adv. Digital Computers, Control and Automation Technologies for Power Plants, Palo Alto, CA, (EPRI) (1992).
- [ROSE-77] J.L. Rose, "A 23 flaw Sorting Study in Ultrasonics and Pattern Recognition," Mat. Eval., July, pp. 87-92 (1977).
- [ROSE-79a] J.L. Rose, G.P. Singh, "A pattern recognition reflector classification feasibility study in the ultrasonic inspection of stainless steel pipe welds," British J. of NDT. Vol. 21, No. 6, Nov., pp. 308-311 (1979).
- [ROSE-79b] J.L. Rose, G.P. Singh, "Stress Corrosion Cracking vs Geometric Reflector Classification Analysis for 304 Austenitic Stainless Steel pipe weld specimens," Proc. 9th World Conf. on NDT, Melbourne, Australia (1979).
- [ROSE-84] J.L. Rose, Y.H. Jeong, E. Alloway, C.T. Cooper, "A Methodology for Reflector Classification Analysis in Complex Geometric Welded Structures," Mat. Eval., Jan., pp. 98-106 (1984).
- [ROUN-80] E.M. Rounds, "A Combined Nonparametric Approach to Feature Selection and Binary Decision Tree Design," Pat. Recog., Vol. 12, pp. 313-317 (1980).
- [SCHA-92] R. Schalkoff, "Pattern Recognition: Statistical, Structural and Neural Approaches," J. Wiley, New York (1992).
- [SEBE-62] G.S. Sebestyne, "Decision-Making Process," Pattern Recognition, MacMillan, New York (1962).
- [SEDG-88] R. Sedgewick, "Algorithms," 2nd Edn., Addison-Wesley, Reading, MA (1988).
- [SETH-82] I.K. Sethi, G.P.R. Sarvarayudu, "Hierarchical Classifier Design using Mutual Information," IEEE Trans. on Pat. Recog. & Mach. Intel., PAMI-4, pp. 441-445 (1982).
- [SETT-87] F. Settle, "Application of Expert Systems in General Chemistry Laboratory," J. Chem. Ed., Vol. 64, No. 4, p. 340 (1987).

- [SHAN-78] R. Shankar, A.N. Mucciardi, W.E. Lawrie and R.N. Stein, "Development of Adaptive Learning Networks for Pipe Inspection," EPRI, NP-688, March (1978).
- [SHOR-74] E.H. Shortliffe, "Computer-Based Medical Consultations: MYCIN," Elsevier, N. Holland (1974).
- [SHOR-75] E.H. Shortliffe, R. Davis, B.G. Buchannan, S.G. Axline, C.C. Green, S.N. Cohen, "Computer-based Consultations in Clinical Therapeutics - Explanation and Rule Acquisition Capabilities of the MYCIN System," Comput. & Biomed. Res., Vol. 8, pp. 302-320 (1975).
- [SHOR-76] E.H. Shortliffe, "MYCIN: Computer-based consultations," American Elsevier, New York (1976).
- [SIDD-81] K.J. Siddiqui, "Machine Recognition of English Text using Contextual postprocessing," MCS thesis, Dept. of Comput. Sci., Concordia Univ., Montreal, Canada (1981).
- [SIDD-86a] K.J. Siddiqui, D.R. Hay, C.Y. Suen, "Evaluation of Materials using a Rule-based Non-destructive Monitoring System," ACM CSC Conf., Cincinnati, Feb., p. 459 (1986).
- [SIDD-86b] K.J. Siddiqui, "A Multifont Text Processing System Using N-gram Statistics," Project Report, Summer Fellowship 1986, Univ. of Neb. at Omaha, Dept. Math/CS, Dec. (1986).
- [SIDD-87a] K.J. Siddiqui, C.Y. Suen, D.R. Hay, "KNOMES: Knowledge Monitoring Expert System," Space Station Automation III, Proc. SPIE symposium on Advances in Intelligent Robotics Systems, Vol. 851, W.C. Chiou, Ed., Cambridge, MA, Nov., pp. 10-18 (1987).
- [SIDD-87b] K.J. Siddiqui, D. Doreen, "Clustering Techniques A Review," paper 87-030, Faculty of Commerce & Administration, Concordia University, Montreal, Nov. (1987).
- [SIDD-88] K.J. Siddiqui, "A Multidisciplinary Unified Approach to Signal Classification," Fourteenth Biennial Symposium on Communications, Kingston, Canada, June, pp. C3:1-C3:4 (1988).
- [SIDD-89a] K.J. Siddiqui, D. Eastwood, R.L. Lidberg, "Expert System for Characterization of Fluorescence Spectra for Environmental Applications, Proc. SPIE,

Fluorescence Detection III, Vol. 1054, pp. 77-90 (1989).

- [SIDD-89b] K.J. Siddiqui, C.Y. Suen, D.R. Hay, "A Knowledge-based Waveform Interpretation System for Intelligent Sensing of Defects in Materials," 4th IGS Conf. Trondheim, Norway, July (1989).
- [SIDD-89c] K.J. Siddiqui, Y.-H. Liu, D. Doreen, "An Integrated Decision Support System," in Managing Information Technology, 17th Annual N. American Conf. of IBSCUG, July, Hershey, PA, pp. 124-130 (1989).
- [SIDD-89d] K.J. Siddiqui, D.R. Hay, C.Y. Suen, "Knowledge-based Signal Interpretation System for Intelligent Sensing of Defects in Materials," Proc. SPIE, Intelligent Control and Adaptive Systems, Vol. 1196, G. Rodriguez, Ed., Philadelphia, PA, Nov., pp. 99-116 (1989).
- [SIDD-90a] K.J. Siddiqui, Y.-H. Liu, D.R. Hay, C.Y. Suen, "Optimal Waveform Feature Selection using a Pseudo-Similarity Method," Journal of Acoustic Emission, Vol. 1, No. 1, pp. 9-16 (1990).
- [SIDD-90b] K.J. Siddiqui, R. Arbeiter, "Knowledge Based Systems - Kinds and Characteristics," Faculty J., Creighton Univ., Vol. 8, April, pp. 9-21 (1990).
- [SIDD-90c] K.J. Siddiqui, L.E. Collins, D. Fitzpatrick, S. Hendricks, D.R. Hay, C.Y. Suen, "Intelligent Sensing of EEG Signals," Proc. SPIE, Applications of Digital Image Processing XI, Vol. 1349, R.E. Fischer, W.J. Smith, Eds., San Diego, CA, Jul., pp. 460-475, (1990).
- [SIDD-91a] K.J. Siddiqui, R.L. Lidberg, D. Eastwood, G. Gibson, "Expert Systems for Classification and Identification of Waterborne Petroleum Oils," in Monitoring Waters in the 1990's. Meeting New Challenges, ASTM STP 1102, J.R. Hall, G.D. Glysson, Eds., American Society for Testing Materials, Philadelphia, pp. 598-612 (1991).
- [SIDD-91b] K.J. Siddiqui, Y.-H. Liu, "A Proximity-Index Based Optimization Model for Feature Selection," 8th Int. Conf. on Math. & Comp. Modelling, College Park, MD, April (1991).
- [SIDD-91c] K.J. Siddiqui, J.S. Deogun, "Knowledge-Based Document Interpretation System for Psychiatric Diagnosis," Workshop on Future Directions in

Text Analysis and Understanding, Chicago, IL, Oct., pp. 127-131 (1991).

- [SIDD-91d] K.J. Siddiqui, P. Nair, D.D. Doreen, "A Relational Data-Knowledge Based System for Decision Support at Strategic Level of Management," 19th N. Am. Conf. IBSCUG, Columbia, SC, July (1991).
- [SIDD-93a] K.J. Siddiqui, E.C. Greco, N.N. Kadri, S. Mohiuddin, M.H. Sketch, "The prototype of a knowledge based system for monitoring and classification of ECG signals," Ann. Fall Meeting, Biomed. Engr. Soc., Oct. 21-24, Memphis, TN (1993).
- [SIDD-93b] K.J. Siddiqui, E.C. Greco, N.N. Kadri, S. Mohiuddin, M.H. Sketch, "Best Feature Selection using Successive Elimination of Poor Performers," IEEE 15th Ann. Int. Conf. of Engr. in Med. & Bio. Soc., Oct. 28-31, San Diego, CA (1993).
- [SIDD-93c] K.J. Siddiqui, S. Mohiuddin, M.H. Sketch, "Unbiased Feature Selection via Successive Elimination of Redundant Features," Submitted to IEEE BioMed. Engr. (1993).
- [SIDD-93d] K.J. Siddiqui, Y.-H. Liu, "A Proximity-Index Based Optimization Model for Feature Selection," J. Modelling in the Scientific Computing, Vol. 2, pp. 381-386 (1993). (Also presented at the 8th Int. Conf. on Math. & Comp. Modelling, College Park, MD, April 1991).
- [SIDD-94a] K.J. Siddiqui, Y.-H. Liu, D.R. Hay, C.Y. Suen, "Feature Selection using a Proximity - Index Optimization Model," Accepted for publication in Pattern Recognition Letters, Netherlands (1994).
- [SIDD-94b] K.J. Siddiqui, "Knowledge-Based Non-Destructive Classification of Defects in Materials," submitted to Knowledge Based Systems Journal (1994).
- [SILV-92] H.S. Silvus, "Failure Analysis of Electrical Components," Nuclear Plant J., May/June, pp. 58-67 (1992).
- [SING-81] G.P. Singh, "Flaw Classification in Austenitic Stainless Steel Pipes Using the Digital Signal-Processing and Pattern Recognition Approach," Final Rep., SW Res. Inst., Project No. 17-9318, Aug. (1981).



- [SING-83] G.P. Singh, R.C. Manning, "An Artificial Intelligence approach to Ultrasonic weld evaluation," in Rev. of Prog. in Quantitative NDE, D.O. Thompson, D.E. Chimenti, Eds., Plenum, New York, pp. 245-255 (1983).
- [SING-92] G.P. Singh, D.A. Steinke, "Piping System Inspection and Testing: managing the Massive Results, Records, and Reports," Nuclear Plant J., Nov./Dec. pp. 58-59 (1992).
- [SKOR-86] E. Skordalakis, "Syntactic ECG Processing: A Review," Pattern Recognition, Vol. 19, No. 4, pp. 305-313, (1986).
- [SLAG-89] J.R. Slagle, S.M. Finkelstein, L.A. Leung, W.A. Warwick, "Monitor: An Expert System that Validates and Interprets Time-Dependent Partial Data Based on a Cystic Fibrosis Home Monitoring Program," IEEE Trans. BioMed. Engr., Vol. BME-36, No. 5, pp. 552-558 (1989).
- [SMET-80] P. Smets, "New quantified approach for diagnostic classification," in Optimization of Computer ECG Processing, H.K. Wolf, P.W. McFarlane, Eds., N. Holland, p. 229 (1980).
- [SOGL-85] G. Sogliero, D. Eastwood, J. Gilbert, "A Concise Feature Set for the Pattern Recognition of Low-temperature Luminescence Spectra of Hazardous Chemicals," American Society for Testing of Materials, STP-863, L.J. Cline-Love, D. Eastwood, Eds., ASTM, Philadelphia, PA, pp. 95-115 (1985).
- [SPEC-67] D.F. Specht, "Generation of Polynomial Functions for Pattern Recognition," IEEE Trans. Electron. Comput., Vol. 16, No. 3, pp. 308-319 (1967).
- [SPEK-87] E.-J. Speckmann, C.E. Elger, "Introduction to the Neurophysiological Basis of the EEG and DC potentials," in EEG: Basic Principles, Clinical Applications and Related Fields, E. Niedermeyer, F.L. De Silva (Eds.), 2 Edn., Urban & Schwarzenberg (1987).
- [STAL-82] J.E. Staley, "An Integrated Equation Technique for Non-destructive Eddy Current Testing on Metals," MS Thesis, Dept. of Elect. Engr., Washington State Univ. (1982).
- [STEAL-88] S.D. Stearns, R.A. David, "Signal Processing Algorithms," Prentice-Hall, Englewood Cliffs, NJ

(1988) .

- [STEF-81] M. Stefik, "Planning and Meta-Planning (MOLGEN: Part 1)," AI Vol. 16, No. 2, pp. 111-139 (1981) .
- [SWAI-77] P.H. Swain, H. Huska, "The Decision Tree Classifier: Design and Potential," IEEE Trans. Geosci. Electron., Vol. GE-15, pp. 142-147 (1977) .
- [TALM-86] J.L. Talmon, "A multiclass nonparametric partitioning algorithm," Pat. Recog. in Practice II, E.S. Gelsema, L.N. Kanal, Eds., Elsevier, N. Holland, pp. 449-458 (1986) .
- [TATS-88] J. Tatsuno, H. Ashida, A. Takao, "Objective Evaluation of Differences in Patterns of EEG Topographical Maps by Mahalanobis Distance," Electroencephalography and Clinical neurophysiology, Elsevier, Ireland, pp. 287-290 (1988) .
- [THIM-90] H. Thimbleby, "User Interface Design," Addison-Wesley, Reading, MA (1990) .
- [TOU-63] J.T. Tou, R.P. Heydoren, "Some Approaches to Feature Extraction," Comput. and Info. Sci. II, Academic Press, New York, pp. 57-89 (1963) .
- [TOU-74] J.T. Tou, R.C. Gonzalez, "Pattern Recognition Principles," Addison-Wesley, Reading, MA (1974) .
- [TRAH-89] P. Trahanias, E. Skordalakis, "Bottom-up Approach to ECG Pattern Recognition Problem," J. Med. & Bio. Engr. & Comput. Vol. 27, pp. 221-229 (1989) .
- [TSAI-80] W. Tsai, K.S. Fu, "Attributed Grammar-A tool for Combining Syntactic and Statistical Approaches to Pattern Recognition," IEEE Trans. Sys. Man Cybern., Vol. SMC-10, No. 12, pp. 873-885 (1980) .
- [TUTT-83] M.S. Tuttle, D.D. Sherertz, M.S. Blois, N. Nelson, "Expertness: from structured text? RECONSIDER: A diagnostic promoting program," Proc. Conf. App. Natural Language Processing, Santa Monica, CA, pp. 124-131 (1983) .
- [ULLM-73] J.R. Ullman, "Pattern Recognition Techniques," Crane Russak, New York (1973) .
- [UMBA-93] S.E. Umbaugh, R.H. Moss, W.V. Stoecker, G.A. Hance, "Automatic Color Segmentation Algorithms," IEEE Engineering in Med. & Bio., Vol. 12, No. 3, pp. 75-82 (1993) .

- [VARY-79] A. Vary, R.F. Lark, "Correlation of Fiber Composite Tensile Strength with the Ultrasonic Stress Wave Factor," J. of Testing & Evaluation, Vol.7, No. 4, July, pp.185-191 (1979).
- [VARY-82] A. Vary, "Acousto-Ultrasonic Characterization of Fiber Reinforced Composites," Material Evaluation, Vol 40, No. 6, May, pp. 650-662 (1982).
- [VARY-87] A. Vary, " The Acousto-Ultrasonic Approach," NASA Technical Memo 89843, April (1987).
- [VEKL-93] E. Veklerov, M.S. Roos, "Management of Multidimensional Data Structures in MRI Imaging," IEEE Engineering in Med. & Bio., Vol. 12, No. 3, pp. 60-63 (1993).
- [WANG-84] Q.R. Wang, C.Y. Suen, "Analysis and Design of a Decision Tree based on Entropy Reduction and its Application to Large Character Set Recognition," IEEE Trans. Pat. Recog. Mach. Intel., Vol. PAMI-6, No. 4, pp. 406-417 (1984).
- [WARD-63] J.H. Ward, "Hierarchical Grouping to Optimize an Objective Function," J. Am. Statist. Ass. Vol. 58 pp. 235-244 (1963).
- [WATA-85] S. Watanabe, "Pattern Recognition: Human and Mechanical," J. Wiley, New York (1985).
- [WATT-71] A.H. Watt, R.L. Beurle, "Recognition of Hand-printed Numerals Reduced to Graph-representable Form," Proc. 2nd Int. Jt. Conf. Artificial Intelligence, pp. 322-332 (1971).
- [WEIS-84] S.M. Weiss, C.A. Kulikowski, "A Practical Guide to Designing Expert Systems," Rowman & Allenhed Publishers (1984).
- [WEXE-89] R.L. Wexelblat, "On Interface Requirements for Expert Systems," AI Magazine, Fall, pp. 66-78 (1989).
- [WILS-73] S.M. William, A.M. Demetrois, "A partitioning Algorithm with Application in Pattern Classification and Optimization of Decision Trees," IEEE Trans. Comput., Vol. C-22, No. 1, pp. 93-103(1973).
- [WIPK-74] W.T. Wipke, "Computer-assisted 3 dimensional synthetic analysis," in Computer Representation and Manipulation of Chemical Information, W.T. Wipke, S.R. Heller, J.R. Feldmann, E. Hyde, Eds. Wiley

Interscience, New York, pp. 147-174 (1974).

- [WURS-86] R. Wu, H. Stark, "Rotation-invariant Pattern Recognition using Optimum Feature Extraction," Pattern Recognition in Practice II, E.S. Gelsema, L.N. Kanal, Eds., Elsevier-Science, N. Holland, pp. 401-410 (1986).
- [YOUN-86] T.Y. Young, K.S. Fu (Eds.), "Handbook of Pattern Recognition and Image Processing," Academic Press, Orlando, FL (1986).
- [ZHAN-80] S. Zhang, "The Regular Expressions Inference for Syntactic Recognition of Handwritten Numerals," Proc. 5th Int. Conf. on Pattern Recognition, pp. 1004-1006 (1980).

## Appendix - A

### FEATURE EXTRACTION DETAILS

This Appendix provides additional details on the extraction of features. It is necessary to refer to Chapters 2, 3 and 4 to understand the contents of this appendix. The features extracted for two problems, namely: NDT and EEG are described in order. The features we used for PNA problem are already described in Section 7.7. We had no information on the CEL problem and as such the features provided by Tektrend were labeled in order of their storage, i.e., 1 through 85. A majority of features in all three problems were measured by extracting envelopes from the waveform in a specified domain. Since CEL problem was also a signal classification problem we suspect that the same methodology might have been used for this problem as well. The procedures for mapping and the algorithm for extracting an envelope are described in Chapter 2.

A computer program was developed to identify the significant peaks in a signal/spectra, in an information domain. The procedure returns 24 descriptive values listed in Table A.A.1. All descriptive values were related to seven reference points in each peak and are defined below:

$p_1$  = start of a peak

$p_2$  = maximum of a peak

$p_3$  = end of a peak

Ascending and descending slopes at points  $p_1$ ,  $p_2$ , and  $p_3$  were also determined which in turn determined the thresholds for the ascending and descending amplitudes, i.e.,

$$\text{Asc\_Amp} = \text{Slope } (p_2) - \text{Slope } (p_1)$$

$$\text{Dsc\_Amp} = \text{Slope } (p_2) - \text{Slope } (p_3)$$

$$p_4 = \text{sample where Slope} > \text{Slope } (p_1) + (0.1 * \text{Asc\_Amp})$$

$$p_5 = \text{sample where Slope} > \text{Slope } (p_1) + (0.9 * \text{Asc\_Amp})$$

$$p_6 = \text{sample where Slope} > \text{Slope } (p_3) + (0.9 * \text{Dsc\_Amp})$$

$$p_7 = \text{sample where Slope} > \text{Slope } (p_3) + (0.1 * \text{Dsc\_Amp})$$

Table A.A.1

Descriptive Values Derived from the Envelope

Value No.	Derivation
1	Ascending span = $p_5 - p_4$
2	Descending span = $p_7 - p_6$
3	Ascending slope (S) = $(S(p_5) - S(p_4)) / (p_5 - p_4)$
4	Descending slope = $(S(p_7) - S(p_6)) / (p_7 - p_6)$
5	Top = location of the peak's maximum
6	base width = $(p_7 - p_4)$
7	base width = $(p_6 - p_5)$
8	max (Asc_Amp, Dsc_Amp)
9	half ascent width = $\text{Top} - (p_4 + p_5) / 2$
10	half descent width = $(p_7 + p_6) / 2 - \text{Top}$
11	$p_1$
12	$p_3$
13	$p_4$
14	$p_5$
15	$p_6$
16	$p_7$
17	$S(p_4)$
18	$S(p_5)$
19	$S(p_6)$
20	$S(p_7)$
21	Total area under the peak. The baseline was Min ( $p_1$ , $p_3$ )
22	Area before Top
23	Area after Top
24	No. of peaks

## A. Features Measured for NDT Problem

To obtain a larger set of features the measurements were made in time domain as well as in four other domains. Features 1 through 108 were measured by the staff at Tektrend, whereas the features 109 through 114 were extracted by the author from the raw waveforms in the time domain. These features are listed in Table A.A-3.

### 1. Time Domain

Features 1 through 36, and 109 through 114 are derived from the time domain. The jump in feature numbers exists because the last six features were added after the rest of the list had been created.

Table A.A-2

Time Domain Features

Feature Id	Explanation / Procedure to Measure
1	Total number of significant peaks
2	Number of peaks whose amplitudes exceed 10% of the maximum amplitude of the signal
3	Number of peaks whose amplitudes exceed 25% of the maximum amplitude
4	Location of the largest peak
5	Amplitude of the largest peak
6	Location of the second largest peak
7	Amplitude of the second largest peak
8	Location of the third largest peak
9	Amplitude of third largest peak
10	Percentage of the total area under all peaks that the largest peak covers
11	Percentage of the total area under all peaks that the second largest peak covers
12	Percentage of the total area under all peaks that the third largest peak covers

Contd.

Table A.A-2 (Contd.)

## Time Domain Features

Feature Id	Explanation / Procedure to Measure
13	The distance (time) between the largest and the second largest peaks
14	The distance (time) between the largest and the third largest peaks
15	The distance (time) between the second largest and the third largest peaks
16	The time of significant ascent in the largest peak
17	The slope of significant ascent in the largest peak
18	The time of significant descent in the largest peak
19	Slope of significant descent in the largest peak
20	The base width of the largest peak
21	The width from half of ascent of the largest peak, plus the width to half of descent
22	The top width of the largest peak
23-29	Same as features 16 through 22, except they are for the second largest peak.
30-36	Same as features 16 through 22, except they are for the third largest peak
109	The sum of the products of the amplitude of each peak and the distance (time) to its predecessor. This is a slight modification of the Acousto Ultrasonic Parameter (AUP) suggested by [VARY-79]. It is an approximation to the area under the curve formed by joining the significant peaks of the time domain
110	This is a feature that can be considered another modification of the AUP. It is sum of the products of two numbers. The first number is the difference in amplitude between a peak and its predecessor. The second number is the distance between the same peak and its predecessor. The feature is zero if all significant peaks had the same amplitude, and it becomes larger as the amplitude of the significant peaks becomes more variable.

Contd.



Table A.A-2 (Contd.)

## Time Domain Features

Feature Id	Explanation / Procedure to Measure
111-114	These features are related to the Coefficient of Kurtosis. Coefficient of Kurtosis (CK) for one peak is defined as:

$$CK = m_4 / (m_2)^2,$$

where

$$m_4 = (1/N) * (A_i - A)^4,$$

$$m_2 = (1/N) * (A_i - A)^2,$$

N = number of samples in the peak,  
 $A_i$  = value of sample i,  
 and A = mean value of all samples in the peak.

The CK value is computed for the three largest peaks.

Feature 111 is CK of the largest peak, 112 is CK of the second largest peak, 113 is CK of the third largest peak, and 114 is the sum of 111 through 113.

## 2. Power Domain

The power domain provides features 37 through 54. Table A.A-3 lists these features.

Table A.A-3

## Power Domain Features

Feature Id	Explanation / Procedure to Measure
37	Number of peaks in the data set
38	Number of peaks above 10% of maximum power in the data set
39	Number of peaks above 25% of maximum power
40	Location of the largest peak
41	Amplitude of the largest peak

Contd.

Table A.A-3 (Contd.)

## Power Domain Features

Feature Id	Explanation / Procedure to Measure
42	Location of the second largest peak
43	Amplitude of the second largest peak
44	Percentage of the area under all peaks in the domain that is under the largest peak
45	Percentage of the area under all peaks that is under the second largest peak.
46	Distance (frequency span) between the location of the largest and second largest peaks
47-54	The original plan was that these features would be the partial power in each octant of frequency. Because the energy in the highest three quarters of frequency was filtered out, the remaining quarter of frequency that still has the data was divided into eight parts as these eight features.

## 3. Phase Domain

The phase domain contains features 55 through 72 and are listed in Table A.A-4:

Table A.A-4

## Phase Domain Features

Feature Id	Explanation / Procedure to Measure
55	Number of peaks in the data set
56	Number of peaks above 10% of maximum phase in the data set
57	Number of peaks above 25% of maximum phase
58	Location of the largest peak
59	Amplitude of the largest peak
60	Location of the second largest peak
61	Amplitude of the second largest peak
62	Percentage of the area under all peaks in the domain that is under the largest peak

Contd.

Table A.A-4 (Contd.)

## Phase Domain Features

Feature Id	Explanation / Procedure to Measure
63	Percentage of the area under all peaks that is under the second largest peak.
64	Distance (frequency span) between the location of the largest and second largest peaks
65-72	The original plan was that these features would be the partial power in each octant of frequency. Because the energy in the highest three quarters of frequency was filtered out, the remaining quarter of frequency that still has the data was divided into eight parts as these eight features.

## 4. Cepstral Domain

The cepstrum provides features 73 through 90 and are listed in Table A.A-5:

Table A.A-5

## Cepstral Domain Features

Index	Explanation / Procedure to Measure
73	Number of peaks in the data set
74	Number of peaks above 10% of maximum cepstrum value in the data set
75	Number of peaks above 25% of maximum cepstrum value
76	Location of the largest cepstral peak
77	Amplitude of the largest peak
78	Location of the second largest peak
79	Amplitude of the second largest peak
80	Percentage of the area under all peaks in the domain that is under the largest peak
81	Percentage of the area under all peaks that is under the second largest peak.

Contd.

Table A.A-5 (Contd.)

## Cepstral Domain Features

Index	Explanation / Procedure to Measure
82	Distance (frequency span) between the location of the largest and second largest peaks
83-90	These features are the partial area of the cepstrum curve in each octant of cepstral "frequency." It is not necessary to use only the lowest quarter of "frequency" because the process of taking the natural logarithm and doing another FFT spread energy to all "frequencies."

## 5. Autocorrelation Domain

The autocorrelation domain provides features 91 through 108 and are listed in Table A.A-6:

Table A.A-6

## Autocorrelation Domain Features

Feature Id	Explanation / Procedure to Measure
91	Number of peaks in the data set
92	Number of peaks above 10% of maximum autocorrelation value in the data set
93	Number of peaks above 25% of maximum autocorrelation value
94	Location of the largest cepstral peak
95	Amplitude of the largest peak
96	Location of the second largest peak
97	Amplitude of the second largest peak
98	Percentage of the area under all peaks in the domain that is under the largest peak
99	Percentage of the area under all peaks that is under the second largest peak.
100	Distance (frequency span) between the location of the largest and second largest peaks

Contd.

Table A.A-6 (Contd.)

## Autocorrelation Domain Features

Feature Id	Explanation / Procedure to Measure
101-108	These features are derived from the intermediate product, which is the transform of autocorrelation. The steps in this transform data were treated as frequency steps and Fourier power coefficients were derived (as described above) from these transform coefficients. These eight features were planned to be the partial power in each octant, but here again the preprocessing made all the frequency equal to zero. Therefore, the lowest quarter of frequency was divided into eight parts, and the partial powers in those parts were saved as these eight features.

## B. Features used for EEG Problem

## 1. Statistical Features

Using standard formulas for statistical measures of variations and measures of dispersions [ANDR-58] these features were evaluated and are listed in Table A.B.1

Table A.B-1

## Statistical Features

Index	Label	Explanation / Procedure to Measure
1.	SMV	Signal Mean Value
2.	SSD	Signal Standard Deviation
3.	SKF	Skewness Factor
4.	KUR	Kurtosis Excess
5.	CVR	Coefficient of Variation

## 2. Zero Crossing Features

These features are listed in Table A.B-2

Table A.B-2

### Statistical Features

Index	Label	Explanation / Procedure to Measure
6.	AVF	Average Frequency of Zero Crossing of Original Signal.
7.	AFD	Average Frequency of Zero Crossing in the same direction of Original Signal.
8.	AF1	Average frequency of Zero Crossing of 1st derivative.
9.	AF2	Average Frequency of Zero Crossing in the same direction of 1st Derivative.
10.	AF3	Average Frequency of Zero Crossing of 2nd Derivative.
11.	AF4	Average Frequency of Zero Crossing in the same direction of 2nd Derivative.

## 3. Hjorth Slope Descriptor

These features are listed in Table A.B-3.

Table A.B-3

### Statistical Features

Index	Label	Explanation / Procedure to Measure
12.	MOB	$MobilityM = \sqrt{(a_2/a_0)}$
13.	CPX	$ComplexityC = \sqrt{(a_4/a_2) - (a_2/a_0)}$
	where	a0 is the variance of the original signal a2 is the variance of the 1st derivative a4 is the variance of the 2nd derivative

## 4. Time Domain Pulse Shape Feature

These features are listed in Table A.B-4

Table A.B-4

## Statistical Features

Index	Label	Explanation / Procedure to Measure
14.	NPK	No. of Peaks above base line of the original signal.
15.	PK1	No. of Peaks above 10% maximum signal amplitude.
16.	PK2	No. of Peaks above 25% maximum signal amplitude.
17.	APR	Average Amplitude of Rising Peaks.
18.	APF	Average Amplitude of Falling Peaks.
19.	PRT	Greatest Peak Rise Time (Original Signal)
20.	PRS	Greatest Peak Rise Slope (Original Signal)
21.	PFT	Greatest Peak Fall Time (Original Signal)
22.	PFS	Greatest Peak Fall Slope (Original Signal)
23.	PPW	Greatest Peak Pulse Width (Original Signal)
24.	HPW	Greatest Peak Half Pulse Width (Original Signal)

## 5. Features from Derivatives of Original Signal

These features are listed in Table A.B-5.

Table A.B-5

## Statistical Features

Index	Label	Explanation / Procedure to Measure
25.	AIN	Average Interval between 2 consecutive zero crossings of same polarity of (1st derivative).
26.	PKD	No. of Peaks above signal base line (1st derivative).
27.	DR1	No. of Peaks above 10% maximum signal amplitude (1st derivative).
28.	DR2	No. of Peaks above 25% maximum signal amplitude (1st derivative).
29.	DR3	No. of Peaks above signal base line (2nd derivative).
30.	DR4	No. of Peaks above 10% maximum signal amplitude (2nd derivative).
31.	DR5	No. of Peaks above 25% maximum signal amplitude (2nd derivative).

## 6. Low Frequency Spectra Power Distribution Features

These features are listed in Table A.B-6.

Table A.B-6

### Statistical Features

Index	Label	Explanation / Procedure to Measure
32-81	P01 - P50	Features 32 to 81 represent the % of partial power in 1 Hz sections (from DC to 50Hz) expressed in units of percentage with respect to the sum of power distributed in the DC to 50Hz range.
82.	MXP	Maximum % of Partial Power in the DC to 50 Hz range.
83	MNP	Minimum % of Partial Power in the DC to 50 Hz range.

## 7. Higher Frequency Spectra Power Distribution Features

These features are listed in Table A.B-7.

Table A.B-7

### Statistical Features

Index	Label	Explanation / Procedure to Measure
84-92	PH1 - PH9	Features 84 to 92 represent the % of partial power in 50 Hz sections (from 50 Hz to 500 Hz) expressed in units of percentage with respect to the sum of power distributed in the DC to 500Hz range.
93.	XPN	Maximum % of Partial Power in the DC to 500 Hz range.
94.	NPH	Minimum % of Partial Power in the DC to 500 Hz range.



## 8. Auto-Correlation Pulse Shape Features

These features are listed in Table A.B-8.

Table A.B-8

### Statistical Features

Index	Label	Explanation / Procedure to Measure
95.	APK	No. of Peaks above signal base line (auto-correlogram).
96.	AP1	No. of Peaks above 10% maximum signal amplitude (auto-correlogram).
97.	AP2	No. of Peaks above 25% maximum signal amplitude Position
98.	GPK	Position of Greatest Peak
99	GPL	Greatest Peak Amplitude
100	2PP	2nd greatest Peak Position
101	2PA	Amplitude of 2nd the Greatest Peak
102	PKA	% Total Area under the Greatest Peak
103	TAR	% of Total Area under the 2nd greatest Peak
104	PDS	Distance between the two Greatest Peak.

## 9. Auto-Correlation Spectra Distribution features

These features are listed in Table A.B-9.

Table A.B-9

### Statistical Features

Index	Label	Explanation / Procedure to Measure
105	PP1	% of Partial power in 1st octant - auto correlation spectrum (ACR).
106	PP2	% of Partial power in 2nd octant - ACR
107	PP3	% of Partial power in 3rd octant - ACR
108	PP4	% of Partial power in 4th octant - ACR
109	PP5	% of Partial power in 5th octant - ACR
110	PP6	% of Partial power in 6th octant - ACR
111	PP7	% of Partial power in 7th octant - ACR
112	PP8	% of Partial power in 8th octant - ACR

## **Appendix - B**

### **Acquisition and Characteristics of Data Sets**

#### **B.1 NDT Data Set**

The NDT data was collected using ARIUS I [LACA-88] system which drove a 2.25 MHz ultrasonic signal into the steel bar. The transmitted signal was detected using a model A2385L 3252 Acoustic Emission Technology (AET) transducer and amplified with an AET model 140 preamplifier with 40 db gain and a passband from 100kHz to 2 MHz. The resulting acoustic emission signals were digitized at 16 MHz and the digitized signal was stored on the ARIUS hard disk. A total of 400 data files were created, 40 for each of the 9 flaw types and 40 for the unflawed bar as well. Each data file consisted of 2048 data points. Additional details on this data are given in Chapter 3.

#### **B.2 EEG Data Set**

Data on nine human subjects were recorded from 10 silver electrodes applied with Grass EC2 cream, and referenced to vertex. Electrodes placement was done using the standard international 10-20 placement system, i.e., assigning the electrodes to points: F3, F4, F7, F8, T3, T4, T5, T6, O1, O1, and O2 on the patient's skull. A Grass Model 8 (16 Channels) clinical polygraph with filters set at 0.5 Hz was used for all data acquisition. Data were recorded for later off-line analysis on a 16 channel Vetter - An instrumentation tape recorder having a minimum 3 db band width of 0-100 Hz. Epochs of one second duration were sampled at a rate of 256 samples and then stored on floppy diskettes.

Nine male volunteer subjects were instructed to generate eye and muscle artifacts. Eye artifacts were produced by blink-

ing, fluttering eyelids, and rolling eyes. Muscle artifacts included jaw clenching and raising eye brows. A five minute sample of eye-closed, resting EEG data was also collected. Original EEG paper tracings were inspected. A total of 5800 EEG epochs were selected. There were 2285 eye artifact epochs (977 obvious, 756 subtle, and 552 questionable), 2745 muscle artifact epochs (1639 obvious, 632 subtle, and 474 questionable), and 770 non-artifact epochs. Two subsets were created, one for use in classifier development and other for testing. Each of these subsets contained 600 epochs (200 eye artifact, 200 muscle artifact, and 200 non-artifact)

### **B.3 PNA Data Set**

The samples on 20 classes of petroleum oils (polynuclear aromatic or PNA compounds) were provided by Dr. Eastwood of Lockheed, ESC. The data was generated using the following process:

A Spex Fluorolog-2 spectrofluorometer was used to collect all fluorescence spectra. The system consisted of a double excitation monochromator and a double emission monochromator with gratings ruled at 1200 grooves per millimeter and blazed at 300 nanometer (nm) for excitation and 500 nm for emission. The excitation source consisted of a 450 Watts ozone generating Xenon lamp. Photomultiplier tubes used for the emission detector were a Hamatsu R928 and for the reference detector (rhodamine - B reference quantum counter) a Hamatsu R508. The spectrofluorometer was interfaced to a Spex DM3000 MS-DOS based personal computer. Slit widths used in all synchronous data collection were 1.25 mm (bandpass = 2.1 nm) for both excitation and emission spectra were collected using 1 ug/mL for the oils dissolved in spectroquality cyclohexane using a standard 10 mm fused silica cell. Synchronous spectra were collected by scanning both monochromators with a wave-length interval (off-

set) of 6 nm and collecting emission data from 260 to 610 nm and were digitized with a stepsize of 1 nm. All spectra were collected in the S/R mode, giving a ratio of the emission signal to the reference signal. The spectra were also collected using a radiometric correction. The spectrofluorometer was initially calibrated using a mercury pen lamp with daily calibration being done using an ovalene standard dissolved in polymethylmethacrylate, obtained from Starna Cells, Inc. Cyclohexane blanks were analyzed with each oil sample. These background spectra were then subtracted from the sample spectra. The synchronous spectra were then normalized to compensate for the difference in fluorescence yields among various oils. Cyclohexane was high purity grade from Brudick and Jackson. Reference oils were obtained from Oak Ridge National Laboratory and the Environmental Protection Agency (EMSL- Cincinnati).