

PATTERN RECOGNITION

STATISTICAL, STRUCTURAL
AND NEURAL APPROACHES

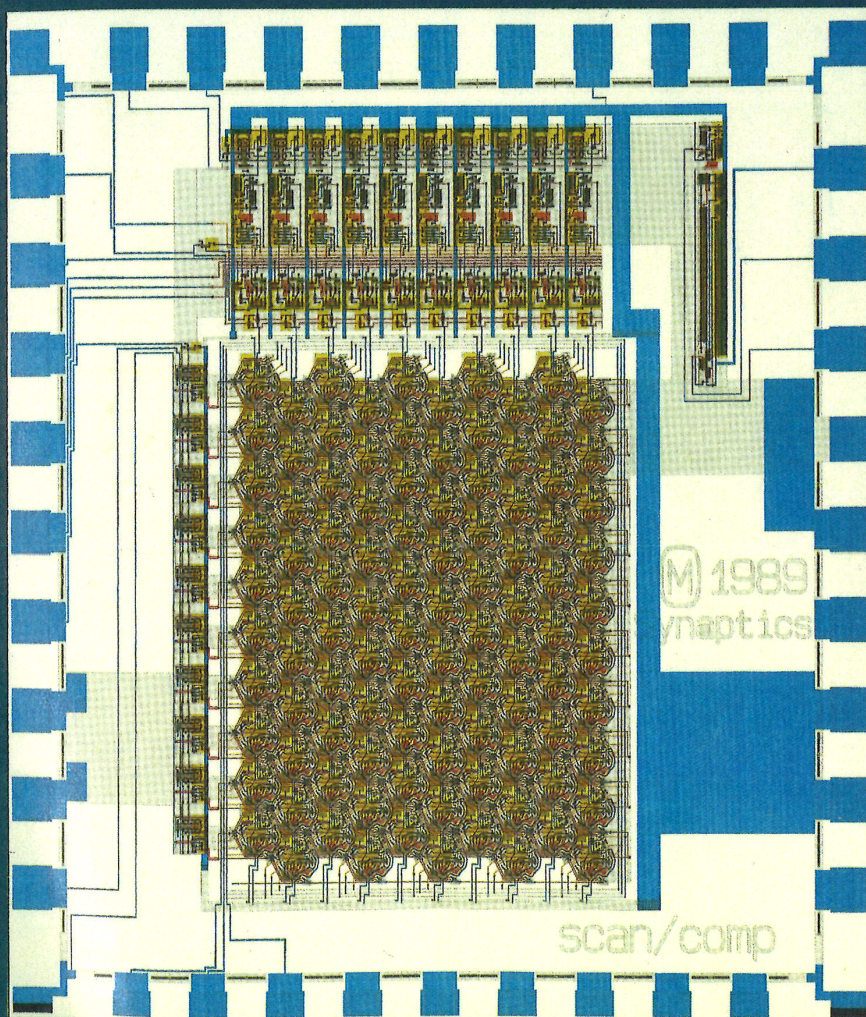
ROBERT SCHALKOFF

PATTERN RECOGNITION
STATISTICAL, STRUCTURAL
AND NEURAL APPROACHES

SCHALKOFF



WILEY



ABOUT THE TEXT

This self-contained text is structured in four main sections.

- **Part 1** (Chapter 1) introduces the overall subject by asking 'What is pattern recognition?' and provides an overview of pattern recognition concerns which are common to all approaches. The three approaches are then briefly introduced and compared.
- **Part 2** (Chapters 2-5) concerns the statistical approach, beginning with a simple example and ending with unsupervised learning through clustering.
- **Part 3** (Chapters 6-9) develops the syntactic or structural approach, beginning with a look at the representational capability of string grammars and corresponding recognition approaches (parsing), continuing with an exploration of higher-dimensional representations and graphical approaches, and ending with an introduction to supervised learning (grammatical inference).
- **Part 4** (Chapters 10-13) details the emerging neural approach, beginning with an overview of neural computing basics, an examination of pattern associators and feedforward nets and associated training algorithms, and concluding with unsupervised learning implementations.

In each case, following development of the theory, examples are shown to illustrate the concepts. Each chapter contains a concluding section which cites relevant literature for more in-depth study of specific topics—approximately 250 references are cited.

Diagram of a Neural Net (experimental). 1989.
Collection, The Museum of Modern Art, New York. Gift of Synaptics,
Incorporated.
Photograph © 1992 The Museum of Modern Art, New York.

Pattern Recognition: Statistical, Structural and Neural Approaches

Robert J. Schalkoff
Clemson University



John Wiley & Sons, Inc.
New York • Chichester • Brisbane • Toronto • Singapore

Acquisitions Editor Steven Elliot
Copy Editor Genevieve Scandone
Production Manager Linda Muriello
Senior Production Supervisor Savoula Amanatidis
Illustration Coordinator Sigmund Malinowski
Manufacturing Manager Lorraine Fumoso

Recognizing the importance of preserving what has been written, it is a policy of John Wiley & Sons, Inc. to have books of enduring value published in the United States printed on acid-free paper, and we exert our best efforts to that end.

Copyright ©1992, by John Wiley & Sons, Inc.

All rights reserved. Published simultaneously in Canada.

Reproduction or translation of any part of this work beyond that permitted by Sections 107 and 108 of the 1976 United States Copyright Act without the permission of the copyright owner is unlawful. Requests for permission or further information should be addressed to the Permissions Department, John Wiley & Sons.

Library of Congress Cataloging in Publication Data:

Schalkoff, Robert J.

Pattern recognition : statistical, structural and neural approaches / Robert J. Schalkoff.
p. cm.

Includes bibliographical references and index.

ISBN 0-471-52974-5 (cloth)

1. Pattern perception--Statistical methods. I. Title.

Q327.S27 1992

006.4--dc20

91-4751
CIP

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

About The Author

Robert J. Schalkoff is Professor of Electrical and Computer Engineering at Clemson University in Clemson, South Carolina. His primary research interests are in artificial intelligence and computer vision with a special emphasis on model-based image understanding, motion and stereo vision, and associated computer architectures. He is also the author of *Digital Image Processing and Computer Vision* (John Wiley & Sons, Inc., 1989) and *Artificial Intelligence: An Engineering Approach* (McGraw-Hill, 1990). He received the Ph.D. degree in Electrical Engineering from the University of Virginia in 1979.

Preface

The art of being wise is the art of knowing what to overlook.

The Principles of Psychology [1890], Chapter 22
William James 1842–1910

This book brings together an identifiable ‘core’ of pattern recognition (PR) ideas, techniques, and applications using statistical, syntactic and neural approaches. It is intended for beginning graduate/advanced undergraduate students as well as practicing engineers and scientists. It is the outgrowth of a graduate course taught in the Electrical and Computer Engineering Department at Clemson University. Like many similar medium-sized and ‘mainstream’ departments, we realized the need to introduce graduate students to the growing inventory of PR topics found in this book, including a structured introduction to neural networks for PR applications. The text is suitable for use in a one- or two-semester course and may be supplemented by individual student projects and readings from the current literature. Since there is more material than could comfortably be covered in a single semester, individual instructors may choose to emphasize or deemphasize specific topics.

As with most technical disciplines, the concept of ‘pattern recognition’ means different things to different people. PR researchers are typically fragmented into disjoint communities. It is my belief that PR is becoming a mature technology with roots in diverse areas such as estimation theory, formal languages, and optimization theory. In addition, the interfaces between pattern recognition and other technologies and application areas such as artificial intelligence and image processing, are quite fuzzy. All three PR approaches (statistical, syntactic, and neural) are currently receiving considerable attention and often share overlapping themes. The choice of one over another is a function of many variables, including constraints of the specific application and

the background of the PR system designer. Therefore, I have attempted an exposition that illustrates the similarities and differences among the three approaches. Although statistical and syntactic pattern recognition are different, they share common ideas. An example is the use of formal models for pattern generation, which leads to formal solution procedures for classification/description. In addition, neural network-based approaches share a number of characteristics with 'traditional' pattern recognition techniques. They include hierarchical structures, clustering, pattern association, and learning or training.

I have also tried to emphasize an 'engineering' approach to PR, since I believe we should not be teaching PR as esoteric mathematics or as the 'art' of developing a set of problem-solving heuristics. An understanding of the underlying models and techniques and their respective limitations is a fundamental aspect of PR system design. Like many engineering and scientific disciplines, PR system design forces the designer to consider trade-offs between exact solutions to approximate models and approximate solutions to exact models. One of the caveats I stress to students is **'In PR problem formulations, you will get what you ask for in the solution, whether you recognize it or not.'** This allows students to appreciate a number of mathematical results that also appeal to intuition. Examples are as follows:

- When a statistical pattern recognition-based algorithm changes decision regions to account for a change in a priori probabilities.
- When a grammatical approach to describing the structure inherent in a pattern yields a context-sensitive grammar.
- When a feedforward-structure neural net, after training with a well-known algorithm, develops an internal layer that allows the attachment of semantics to the action of the internal units.

After completing this book, students should have a good grounding in the fundamentals of PR, exposure to three diverse PR approaches, and some perspective on the applicability of each approach. They should also be able to begin reading the current literature.

Opportunities abound for the application of pattern recognition techniques to present-day problems. Current examples are automated character recognition, digital image processing and analysis, and recognition and understanding of speech. In fact, many pattern recognition applications arise in situations where they are not expected. Examples of these are weather forecasting and stock market analysis. Individual instructors may wish to emphasize PR application in their respective research domains. My own experience with computer vision problems provided a wealth of examples, some of which are used in this text.

An attempt was made to keep the presentation balanced. The text has four main sections. Part 1 (Chapter 1) introduces the overall subject by asking the question What is pattern recognition? It provides an overview of pattern recognition concerns that are common to all approaches. The three approaches are then briefly introduced and compared. Part 2 (Chapters 2-5) concerns the statistical approach, beginning with a simple example and ending with unsupervised learning through clustering. Part 3

(Chapters 6–9) develops the syntactic or structural approach, beginning with a look at the representational capability of string grammars and corresponding recognition approaches (parsing), continuing with an exploration of higher dimensional representations and graphical approaches, and ending with an introduction to supervised learning (grammatical inference). Finally, Part 4 (Chapters 10–13) details the emerging neural approach, beginning with an overview of neural computing basics, an examination of pattern associators and feedforward nets and associated training algorithms, and concluding with unsupervised learning implementations. In each case, following development of the theory, examples are shown to illustrate the concepts. Each chapter contains a concluding section that cites relevant literature for more in-depth study of specific topics. Approximately 250 references are cited.

Some background in probability, linear algebra, and discrete mathematics is necessary to fully appreciate the topics considered in this book. The appendices document these and other fundamental background concepts and enable the book to be self-contained. In addition, numerous exercises are presented that will challenge and motivate the reader to further explore relevant concepts. Many of these exercises provide the bases for projects and thesis work. Instructors should contact the publisher regarding the availability of a solution manual.

This book completes a trilogy of books I have written over the last three years. The other two are *Digital Image Processing and Computer Vision* (John Wiley & Sons, Inc.) and *Artificial Intelligence: An Engineering Approach* (McGraw-Hill). They cover a broad area of computing technology that I believe (or hope) is likely to receive widespread attention as it continues to mature during the '90s.

Acknowledgments

A textbook does not just 'happen,' but rather requires a significant commitment from its author as well as a supporting environment. I have been quite fortunate in this regard. My wife, Leslie, functioned as 'the ultimate pattern recognition system' in transcribing pages of scribbling into an attractive TeX document. Students who used earlier versions of the notes provided both valuable feedback and enhanced motivation, since they proved that the educational objective of this book could be achieved. The environment in the Department of Electrical and Computer Engineering and the College of Engineering at Clemson University was also conducive to this task. The efforts of the professional staff at John Wiley and Sons, especially Steven Elliott and Savoula Amanatidis, deserve special thanks. Finally, in watching my daughter Katie grow, I realize there are lots of things we'll never want to automate.

Robert J. Schalkoff
Clemson, South Carolina

Contents

PART 1 INTRODUCTION AND GENERAL PATTERN RECOGNITION CONCERNS	1
CHAPTER 1 PATTERN RECOGNITION (PR) OVERVIEW	2
What is This Book About?	2
Overview	2
‘Engineering’ Approaches to Pattern Recognition	3
Relationship of PR to Other Areas	3
Pattern Recognition Applications	3
Pattern Recognition, Classification, and Description	4
Abstract Representation of Pattern Mappings	4
Structure of a ‘Typical’ PR System	5
Patterns and Feature Extraction with Examples	6
Patterns and Features	6
Pattern Distortions—A Fundamental and Difficult Problem	8
Example: Feature Extraction Using Generalized Cylinders for 3-D Object Description and Classification	8
Example: Generating RST Invariant Features and Application to 2-D Figure Recognition	10
Feature Extraction	11
Numerical Results and Analysis	12
Analysis	12
What’s Really Different Among Different Patterns?	13
The Feature Vector and Feature Space	13
Definitions	14

Classifiers, Decision Regions and Boundaries, and Discriminant Functions	15
Training and Learning in PR Systems	16
Using A Priori Knowledge or 'Experience'	16
Learning Curves (Improved 'Performance')	17
Training Approaches	17
Pattern Recognition Approaches	18
The Statistical Pattern Recognition Approach (StatPR)	18
The Syntactic Pattern Recognition Approach (SyntPR)	19
The Neural Pattern Recognition Approach (NeurPR)	20
Comparing and Relating StatPR, SyntPR and NeurPR	20
Examples of Pattern Recognition Approaches	20
StatPR	21
SyntPR	26
Example: Combinations of StatPR and SyntPR	26
Procedures for PR System Engineering	26
Other Approaches to PR	27
'Black Box' Approaches	27
Reasoning-Driven Pattern Recognition	28
Overview of PR Literature and Resources	28
Books	29
Journals	29
Conferences	30
Exercises	30
 PART 2 STATISTICAL PATTERN RECOGNITION (StatPR)	 33
 CHAPTER 2 INTRODUCTION TO STATISTICAL PATTERN RECOGNITION	 34
Introduction to Statistical Pattern Recognition	34
Approaches to Developing StatPR Classifiers	34
Simple Examples of Statistical Models and Applications	35
The Gaussian Case and Class Dependence	39
Gaussian Models for $p(\underline{x} w_i)$	39
Discriminant Functions	39
Generalized Results (Gaussian Case)	40
Decision Surfaces for Specific Cases	40
A General Look at the StatPR Approach Structure	43
Additional Examples	43
Example: Uniform Densities, $c = 2$, $d = 1$	43
Example: $c = 3$, $d = 2$, Gaussian Case	44
Extensions	47
Training	47

Alternative Classification Procedures	47
Unsupervised Approaches	48
Classifier Performance, Risk, and Errors	49
Measures of Classification Performance	49
General Measures of Classification Risk	51
Bibliographical Remarks	54
Exercises	54
CHAPTER 3 SUPERVISED LEARNING (TRAINING)	
USING PARAMETRIC AND NONPARAMETRIC	
APPROACHES	58
Introduction	58
Parametric Estimation and Supervised Learning	58
Approaches to Parameter Estimation	59
Supervised versus Unsupervised Learning	59
Maximum Likelihood (ML) Estimation	59
Formulation	59
Use of the Training Set	59
The Likelihood Function	60
The Bayesian Parameter Estimation Approach	62
Supervised Learning Using Nonparametric Approaches	66
Nonparametric Approaches	66
General Aspects of Nonparametric Density Estimation	67
The Basic Approach	67
Parzen Windows	70
Unit Step Function $\phi(\underline{x})$	70
Extension to More General Interpolation Functions	73
k-nn Nonparametric Estimation	74
A Problem with V_0	74
Estimation of $P(w_i \underline{x})$ Directly	75
Examples of Nonparametric Learning	75
Direct Classification Using the Training Set [The	
Nearest Neighbor Rule (NNR)]	75
The NNR Approach	75
Bibliographical Remarks	83
Exercises	83
CHAPTER 4 LINEAR DISCRIMINANT FUNCTIONS	
AND THE DISCRETE AND BINARY	
FEATURE CASES	89
Introduction	89
Linear Discriminant Functions	89
Fisher's Linear Discriminant	90
Discrete and Binary Classification Problems	94

Classification Procedures for Discrete Feature Data	94
Formulation for the Binary Case	95
Classification Procedures for the Binary Case	96
Nonparametric Estimation of $P(\underline{x} w_i)$	98
Techniques to Directly Obtain Linear Classifiers	99
The Concept of Linear Separability	99
Design of Linear Classifiers	104
Bibliographical Remarks	105
Exercises	
CHAPTER 5 UNSUPERVISED LEARNING AND CLUSTERING	109
Formulation of Unsupervised Learning Problems	109
A Modified Training Set	109
Unsupervised Learning Approaches	110
Clustering for Unsupervised Learning and Classification	114
The Clustering Concept and the Search for 'Natural Clusters'	114
The c-means Algorithm	115
Learning Vector Quantization (LVQ)	116
Formal Characterization of General Clustering Procedures	117
Clustering Strategies	118
'Cluster Swapping' Approaches	119
Extended Example: A Hierarchical Clustering Procedure	120
Bibliographical Remarks	121
Exercises	121
PART 3 SYNTACTIC PATTERN RECOGNITION (SyntPR)	127
CHAPTER 6 OVERVIEW	128
Syntactic Pattern Recognition (SyntPR) Overview	128
Quantifying Structure in Pattern Description and Recognition	129
Hierarchical Approaches	129
Relational Models	130
Grammar-Based Approach and Applications	130
Using Formal Linguistics to Model and Describe Patterns	130
Structural Analysis and Description Applications	131
Elements of Formal Grammars	133
Definitions and Conventions	133
Grammar and Languages Using Strings	135
Grammar Application Modes	136
Grammar Types and Productions	137
Other Production Constraints in String Grammars	139
Additional Introductory Concepts	141

Example: Context-Sensitive, Context-Free, and Finite State (Regular) Grammar Productions	142
Examples of String Generation as Pattern Description	144
Example: 2-D Line Drawing Description Grammar [Shaw 1970]	144
Example: Character Description Using PDL	146
Example: Object Description Using Projected Cylinder Models	146
Extended Example: Blocks World Description	146
Remarks on the Heuristic Generation of Grammars	149
Bibliographical Remarks	150
Exercises	150
CHAPTER 7 SYNTACTIC RECOGNITION VIA PARSING AND OTHER GRAMMARS	155
Recognition of Syntactic Descriptions	155
Recognition by String Matching	155
Recognition by Parsing	156
Parsing	156
Fundamental Concepts	156
An Abstract View of the Parsing Problem	157
Parsing Approaches	159
The Cocke-Younger-Kasami (CYK) Parsing Algorithm	160
The Approach	160
The CYK Table	160
(Augmented) Transition Networks (ATNs) in Parsing	163
Transition Networks	163
Augmented Transition Nets (ATNs)	164
Higher Dimensional Grammars	165
Introduction	165
Tree Grammars	166
Traversing and Describing Trees	166
Productions and Derivations with Trees	169
Stochastic Grammars and Applications	170
Stochastic Grammars	171
Derivations in a Stochastic Language	172
Bibliographical Remarks	173
Exercises	174
CHAPTER 8 GRAPHICAL APPROACHES TO SyntPR	176
Graph-Based Structural Representations	176
From Digraphs to Semantic Nets to Relational Graphs	176
Application of Relational Graphs to PR	177
Graph Isomorphism	179

Uniqueness of Digraph-Based Structural Description and Isomorphism	179
Determining Isomorphism	180
Extensions to the Elementary Graph Matching Approach	181
A Formal Characterization of the Relational Graph Similarity Problem	182
Extensions of Relational Graphs (Attributed Graphs)	183
A Structured Strategy to Compare Attributed Graphs	185
Relaxing Requirements for Pattern Isomorphism and Subisomorphism	185
Match Graphs (MGs)	185
Recursive Procedure to Find Cliques [<i>cliques</i> (X, Y)]	186
Other Attributed Graph Distance or Similarity Measures	187
Design and Selection of Similarity Measures	187
Transforming G_i into G_j	188
Extended Example: Structural Unification Using Attributed Graphs	188
Bibliographical Remarks	191
Exercises	192
 CHAPTER 9 LEARNING VIA GRAMMATICAL INFERENCE	 194
Learning Grammars	194
Problem Overview	194
Difficulties in Structural Learning	194
Problem Formulation	195
Characterizing the Grammar Source, Ψ	195
Concerns Related to the Training Set	195
Grammatical Inference (GI) Approaches	197
Grammatical Inference Objectives	197
Intuitive Sample GI Procedure	197
Procedures to Generate Constrained Grammars	198
Using a Single String to Partially Infer G_{learn}	198
Generation of Canonical Definite Grammars	199
General Procedure to Generate CDFSG, (G_c)	199
Bibliographical Remarks	200
Exercises	201
 PART 4 NEURAL PATTERN RECOGNITION (NeurPR)	 203
 CHAPTER 10 INTRODUCTION TO NEURAL NETWORKS	 204
Neurons and Neural Nets	204
Introduction	204

History	205
Neural Networks as a Black Box Approach and 'Artificial' Neural Systems	205
Key Neural Network Concepts	206
Characteristics of Neural Computing Applications	206
'Connectionist' Models and Computing	207
Neural Network Structures for PR Applications	207
Neural Network Structure (How Should I Connect My Neurons?)	207
Learning in Neural Networks	208
The Neural PR Application Design Phase	210
Why Is Neural Computation So Important?	210
Physical Neural Networks	211
The Physical Neuron	211
Dynamics of a Biological Neural System	212
The Artificial Neural Network Model	213
Artificial Neuron Activation and Output Characteristics	213
Neural Unit Interconnection Strengths (Weights)	217
Other Neural Network Parameters	218
Network Implementations for Constraint Satisfaction and Matching Problems	219
Hardware Realizations of Neural Networks	220
Bibliographical Remarks	220
CHAPTER 11 INTRODUCTION TO NEURAL PATTERN ASSOCIATORS AND MATRIX APPROACHES	221
Neural Network-Based Pattern Associators	221
Introduction	221
Design Procedure	221
'Black Box' Structure	222
CAM and Other Neural Memory Structures	223
Desirable Pattern Associator (PA) Properties	224
Matrix Approaches (Linear Associative Mappings) and Examples	224
An Elementary Linear Network Structure and Mathematical Representation	224
Approach 1: A Linear CAM ('Hopfield') Network	226
Approach 2: Matched Filters/Adaptive Filters	227
Approach 3: Outer Product Formulations	228
Approach 4: Generalized Inverse Applications	229
Extended Example: Heteroassociative Memory Design	231
Hebbian or Correlation-Based Learning	232
Bibliographical Remarks	233
Exercises	233

CHAPTER 12 FEEDFORWARD NETWORKS AND TRAINING BY BACKPROPAGATION	236
Multilayer, Feedforward Network Structure	236
Introduction	236
Overall Feedforward Structure	236
The Role of Internal/Hidden Layers (What's 'Hidden?')	238
Training Considerations	239
Training the Feedforward Network: The Delta Rule (DR) and Generalized Delta Rule (GDR)	240
Overview	240
Gradient Descent Approaches for Training	241
Training by Sample or Training by Epoch	241
Derivation of the DR	242
A General Assessment of Error Sensitivity	242
Extension of the DR for Units in the Hidden Layers [The Generalized Delta Rule (GDR)]	246
Back Propagation—Summary of the Multistep Procedure	247
Adding <i>Momentum</i> to the Training Procedure	247
Training the Unit Bias Inputs	249
Example: 'Handworked' Training of a Single Unit Using the GDR	250
Extended Example: Pattern Associator for Character Classification	251
Chosen Features	251
Case 1: Single 4-Valued Output, Pseudoinverse Solution	252
Case 2: Single 4-Valued Output, 2-Layer Neural Network Solution	252
Case 3: 4 Output ('One-of') 2-Layer Neural Network Solution	253
Case 3: Summary of Results	254
Case 4: 4-Output, 3-Layer (Hidden) Neural Network Solution	254
Other Ramifications of Hidden Units	257
Bibliographical and Historical Remarks	258
Exercises	259
CHAPTER 13 CONTENT ADDRESSABLE MEMORY APPROACHES AND UNSUPERVISED LEARNING IN NeurPR	264
Introduction	264
The Hopfield Approach to Neural Computing	265
Network Parameters	265
Network Dynamics	265
Hamming Distance and Its Significance in CAM	267
Example: Hamming Distances Between Stored States	267
Example: Design of a Simple Hopfield Network—Storing and Accessing Stable States	268
Why the Hopfield Strategy Should Work	271

Additional Examples of CAM Applications in PR	272
Example: Character Recognition	272
Example: Relational Constraint Satisfaction (Coloring)	273
Example: Symbolic Constraint Satisfaction (Labeling)	275
Unsupervised Learning in NeurPR: Self-Organizing Networks	275
Introduction	275
Adaptive Resonance Architectures	276
Self-Organizing Feature Maps (Kohonen)	280
Summary	284
Bibliographical Remarks	285
Exercises	286
APPENDIX 1 Linear Algebra Review	289
APPENDIX 2 Probability and Random Variables/Vectors	298
APPENDIX 3 Discrete Mathematics Review	314
APPENDIX 4 Descent Procedures: Examples and Error Function	
Contours	323
APPENDIX 5 Similarity Measures, Matching Techniques, and	
Scale-Space Approaches	328
APPENDIX 6 Geometry for Classification and State-Space	
Visualization	337
REFERENCES	344
PERMISSION SOURCE NOTES	360
INDEX	361

Part 1

Introduction and General Pattern Recognition Concerns

Pattern Recognition (PR) Overview

1

Knowledge is of two kinds. We know a subject ourselves, or we know where we can find information upon it.

From James Boswell, *Life of Johnson* 1791
Samuel Johnson 1709–1784

WHAT IS THIS BOOK ABOUT

Overview

Machine intelligence will be a dominant technology in the 1990s. Pattern recognition (PR) techniques are often an important component of intelligent systems and are used for both data preprocessing and decision making. Broadly speaking, pattern recognition (PR) is the science that concerns the description or classification (recognition) of measurements. There is little doubt that PR is an important, useful, and rapidly developing technology with cross-disciplinary interest and participation. PR is not comprised of one approach, but rather is a broad body of often loosely related knowledge and techniques. Historically, the two major approaches to pattern recognition are the statistical (or decision theoretic) and the syntactic (or structural) approaches. Recently, the emerging technology of neural networks has provided a third approach, especially for 'black box' implementation of PR algorithms. Since no single technology is always the optimum solution for a given PR problem, all three are explored in this text.

Each of the three interrelated pattern recognition approaches:

- Statistical Pattern Recognition (StatPR)
- Syntactic Pattern Recognition (SyntPR)
- Neural Pattern Recognition (NeurPR)

is defined in the following sections. Where appropriate, computational issues related to practical or even 'real-time' implementation of the corresponding PR systems is considered. In order to achieve this breadth of coverage, the scope of some PR topics is restricted.

'Engineering' Approaches to Pattern Recognition

An engineering approach to problem solving involves incorporating all available and relevant problem information in a structured fashion to formulate a solution. This includes the development or modification of models that perhaps incorporate structural and a priori information, including 'training' data.¹ Therefore, throughout this book, our examination of PR approaches will be guided by these three questions:

1. Are pattern recognition techniques suitable, or even applicable to the problem at hand? (Can the problem be solved?)
2. Can we develop or modify useful models for the situation and, if necessary, determine the model parameters?
3. If so, are there formal tools and heuristics that may be applied, and does a computationally *practical* solution procedure exist?

Relationship of PR to Other Areas

Pattern recognition techniques overlap with other areas, such as:

- (Adaptive) signal processing and systems
- Artificial intelligence
- Neural modeling
- Optimization/estimation theory
- Automata theory
- Fuzzy sets
- Structural modeling
- Formal languages

Pattern Recognition Applications

PR applications include:

- Image preprocessing, segmentation, and analysis
- Computer vision
- Seismic analysis
- Radar signal classification/analysis

¹Pattern recognition (especially statistical) books are replete with the terms 'a priori' and 'a posteriori.' We will define 'a priori' as 'known before' (usually 'before' meaning 'before feature measurement') and 'a posteriori' to mean 'derived from subsequent measurements.'

- Face recognition
- Speech recognition/understanding
- Fingerprint identification
- Character (letter or number) recognition
- Handwriting analysis ('notepad' computers)
- Electrocardiographic signal analysis/understanding
- Medical diagnosis

PATTERN RECOGNITION, CLASSIFICATION, AND DESCRIPTION

Abstract Representation of Pattern Mappings

PR may be characterized as an *information reduction*, *information mapping*, or *information labeling* process. An abstract view of the PR classification/description problem is shown in Figure 1. We postulate a mapping between class-membership space, C , and pattern space, P . This mapping is done via a relation, G_i , for each class, and may be probabilistic. Each class, w_i , generates a subset of 'patterns' in pattern space, where the i th pattern is denoted p_i . Note, however, that these subspaces overlap, to allow patterns from different classes to share attributes. Another relation, M , maps patterns from subspaces of P into observations or measured patterns or features, denoted m_i . Using this concept, the characterization of many PR problems is simply that, given measurement m_i , we desire a method to *identify* and *invert* mappings M and G_i for all i . Unfortunately, in practice, these mappings are not functions. Even if they were, they are seldom 1:1, onto or invertible. For example, Figure 1 shows that identical measurements or observations may result from different p_i , which in turn correspond to different underlying classes. This suggests a potential problem with *ambiguity*. Nevertheless, it seems reasonable to attempt to model and understand these processes, in the hope that this leads to better classification/description techniques.

The abstract formulation of Figure 1 is well suited for modeling in both StatPR and SyntPR. Looking ahead, Figure 13 shows how more detailed formalizations of these approaches may be related to Figure 1. We may view the realized patterns in Figure 1 as basic 'world data,' which is then measured. Thus, another important aspect of Figure 1 concerns M . *This mapping reflects our choice of measurement system.* Measurement system design is an important aspect of PR system design, in the sense that good 'features' or 'primitives,' to be derived subsequently, probably require good, or at least adequate, measurements. It is unlikely that erroneous or incomplete measurements will facilitate good PR system performance. Finally, note that patterns that are generated by the same class (p_4 and p_1 , from w_i , for example) and 'close' in pattern space, P , do not necessarily yield measurements (m_1 and m_3 in this case) that are also 'close.' This is significant when 'clustering' of measurement (or feature or primitive) data is used for measuring pattern similarity.

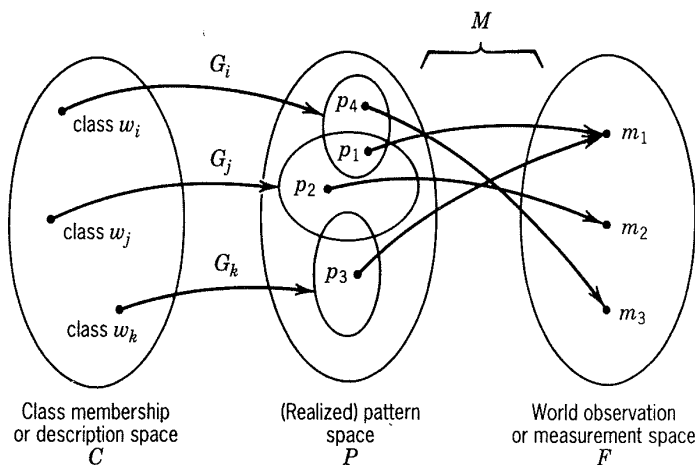


Figure 1: Mappings in an abstract representation of pattern generation/classification/interpretation systems.

Example Use of the Abstract Representation. Consider a two-class problem where w_1 corresponds to the class ‘apple’ and w_2 corresponds to the class ‘hand grenade.’ In realized pattern space, there will be some difference among the p_i resulting from apples, since not all apples are the same size, shape, weight, color, and so on. A similar remark holds true for realizations of class w_2 . Also, some realizations of ‘apple’ and ‘hand grenade’ share similar attributes (e.g., mass, volume, weight). Therefore, basing our PR system on a measurement consisting simply of realized pattern *weight* is likely to:

- (Mis)-classify some apples and hand grenades as the same.
- (Mis)-classify or distinguish between heavy apples and light apples (and similarly for w_2).

Ironically, at the same time we strive to understand and to ‘undo’ these mappings, alternative (so-called black box) approaches, principally implemented by using neural networks, are receiving increased attention. These black box approaches attempt to eliminate the need for detailed information on both the mappings and inverses and, instead, require a good set of training data (sample mappings between C and M) and a trainable system.

Structure of a ‘Typical’ PR System

The structure of a typical pattern recognition system is shown in Figure 2. Notice that it consists of a sensor (an image sensor or camera, for example), a feature extraction mechanism (algorithm), and a classification or description algorithm (depending on the approach). In addition, usually some data that have already been classified or described are assumed available in order to train the system (the so-called training set).

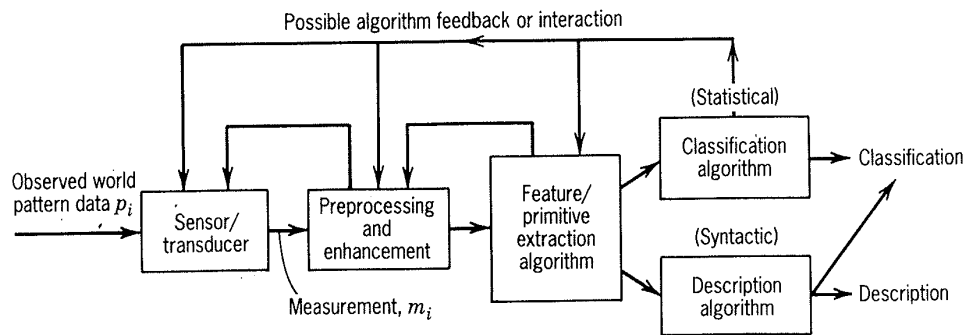


Figure 2: Typical pattern recognition system structures.

PATTERNS AND FEATURE EXTRACTION WITH EXAMPLES

Patterns and Features

Patterns. It comes as little surprise that much of the information that surrounds us manifests itself in the form of patterns. *The ease with which humans classify and describe patterns often leads to the incorrect assumption that this capability is easy to automate.* Sometimes the similarity in a set of patterns is immediately apparent, whereas in other instances it is not. Recognizing characters is an example of the former; economic forecasting illustrates the latter.

Pattern recognition, naturally, is based on *patterns*. A pattern can be as basic as a set of measurements or observations, perhaps represented in vector or matrix notation. Thus, the mapping M , from P to F in Figure 1, is an identity mapping. The use of measurements already presupposes some preprocessing and instrumentation system complexity. These measurements could be entities such as blood pressure, age, number of wheels, a 2-D line drawing, and the like. Furthermore, patterns may be converted from one representation to another.

The measurement, as shown in Figure 3, may be a two-dimensional image, a drawing, a waveform, a set of measurements, a temporal or spatial history (sequence) of events, the state of a system, the arrangement of a set of objects, and so forth.

Features. Broadly speaking, *features* are any extractable measurement used. Examples of low-level features are signal intensities. Features may be symbolic, numerical or both. An example of a symbolic feature is color; an example of a numerical feature is weight (measured in pounds). Features may also result from applying a *feature extraction algorithm or operator* to the input data. Additionally, features may be higher level entities: for example, geometric descriptors of either an image region or a 3-D object appearing in the image. For instance, aspect ratio and Euler number are higher level geometric features. Note that: (1) significant computational effort may be required in feature extraction; and (2) the extracted features may contain errors or

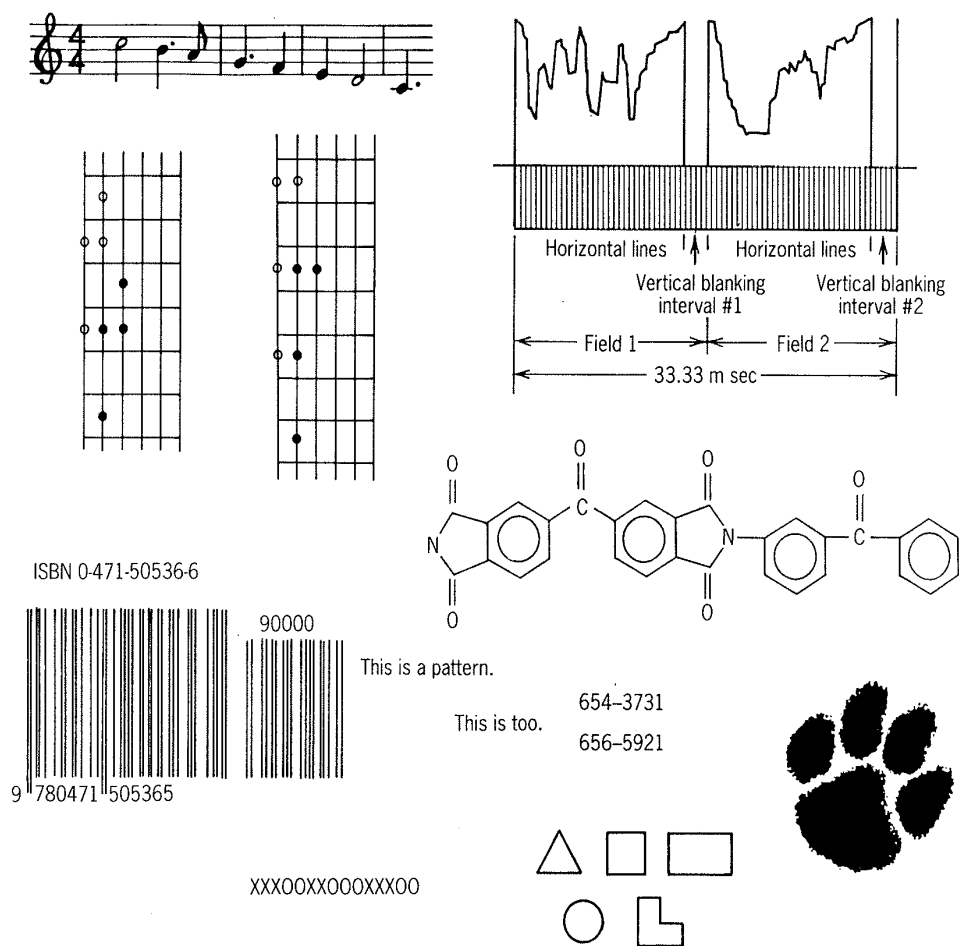


Figure 3: Examples of (visual) patterns/measurements.

'noise.' Features may be represented by continuous, discrete, or discrete-binary variables. *Binary features* may be used to represent the presence or absence of a particular feature.

The related problems of *feature selection* and *feature extraction* must be addressed at the outset of any PR system design. The key is to choose and to extract features that (1) are computationally feasible; (2) lead to 'good' PR system success²; and (3) reduce the problem data (e.g., raw measurements) into a manageable amount of information without discarding valuable (or vital) information.

Feature Selection. *Feature selection* is the process of choosing input to the PR system and involves judgment. It is important that the extracted features be relevant to the PR task at hand. In some cases there are mathematical tools that help in feature selection. In other cases, simulation may aid in the choice of appropriate features. Clearly,

²Perhaps in the form of small (mis)-classification errors.

restrictions on measurement systems may limit the set of possible features. Furthermore, features need not be low-level data. However, the level at which features are extracted determines the amount of necessary preprocessing and may also influence the amount of error that is introduced into the extracted features. For example, in medical applications the features may be *symptoms* and the classes might be states of health, including w_1 = 'healthy,' w_2 = 'has the flu,' and so forth.

Pattern Distortions—A Fundamental and Difficult Problem

Often, we seek classification, recognition, or description of a pattern that is invariant to some (known) changes or deviation in the pattern from the 'ideal' case. These deviations may be due to a variety of causes, including 'noise.' However, in many situations a set of patterns *from the same class* may exhibit wide variations from a single exemplar of the class. For example, humans are able to recognize (that is, classify) printed or handwritten characters with widely varying font sizes and orientations. Although the exact mechanism that facilitates this capability is unknown, it appears that the matching strongly involves structural analysis of each character. This is in contrast to a straightforward template matching or correlation approach (Appendix 5), which is usually unsuccessful in applications with significant pattern distortions.

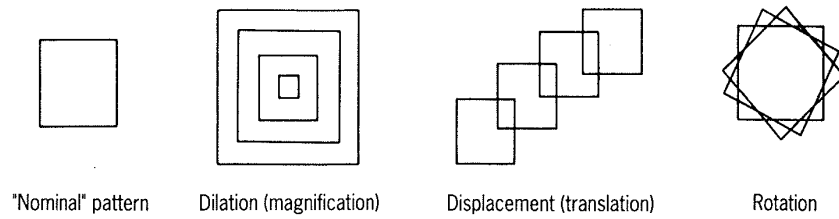
Examples of pattern perturbations are shown in Figure 4. In this case, a careful choice of *invariant features* and pattern structure could be used for recognition. For example, in image processing PR applications, we often seek recognition of objects when the objects may be in arbitrary positions (Translated), angular orientations (Rotated), and Scale. Thus, RST invariant features [Casasent 1976], [Schalkoff 1989] are desired.

EXAMPLE 1: Feature Extraction Using Generalized Cylinders for 3-D Object Description and Classification

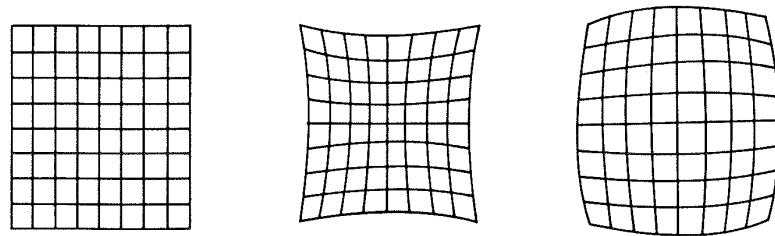
The classification of 3-D objects from visual (image) data is an important pattern recognition task. This task exemplifies many aspects of a typical PR problem, including feature selection, dimensionality reduction (3-D world information is mapped onto a 2-D image), and the use of qualitative and structural descriptors. Generalized cylinder (GC) [Agin/Binford 1976], [Biederman 1985] models of 3-D objects provide a framework for the following:

1. Models for class-specific generation of features;
2. Feature extraction; and
3. Structural approaches to building more complex 3-D objects.

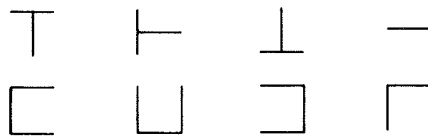
The basis of GC models is the concept of a 'swept volume' of 2-D area along a 3-D trajectory. Specifically, a generalized cylinder is a solid whose axis is a 3-D space curve. Usually the axis is perpendicular to the cross section. For example, the typical cylinder or 'can' may be described by sweeping a circle along a line. We characterize a generalized cylinder by three parameters:



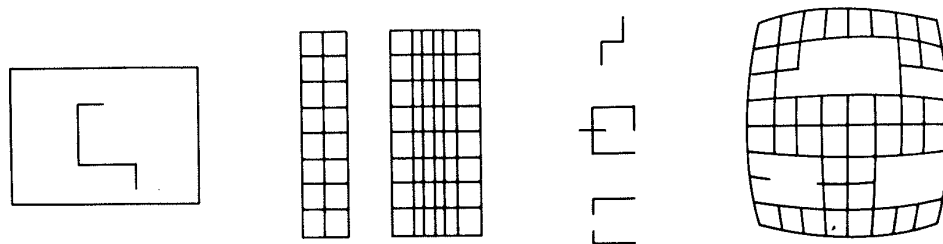
(a)



(b)



(c)



(d)

Figure 4: Examples of pattern distortions (visual patterns).
 (a) Geometric distortions of visual (image) patterns-square.
 (b) Geometric distortions of grid pattern.
 (c) Geometric distortions of character patterns.
 (d) More extreme pattern distortions (missing parts and extra parts) using the patterns of parts (a) to (c).

1. A planar cross section;
2. A three-dimensional curve or 'spine'; and
3. A sweeping rule.

As shown in Figure 5, different 3-D objects may be generated by defining the following features:

1. A qualitative descriptor of the cross section edge curvature. This may be either straight (S) or curved (C).
2. The cross section degree of symmetry. Symmetry is defined as invariant under reflection and rotation (Symm++), invariant under reflection only (Symm+), or asymmetric (Asymm).
3. The change of cross section as a function of sweep. This may be constant (Const), expanding (Exp), contracting (Contr), expanding then contracting (Exp-Contr), or contracting then expanding (Contr-Exp).
4. The degree of curvature of the spine. This may be straight (S) or curved (C).

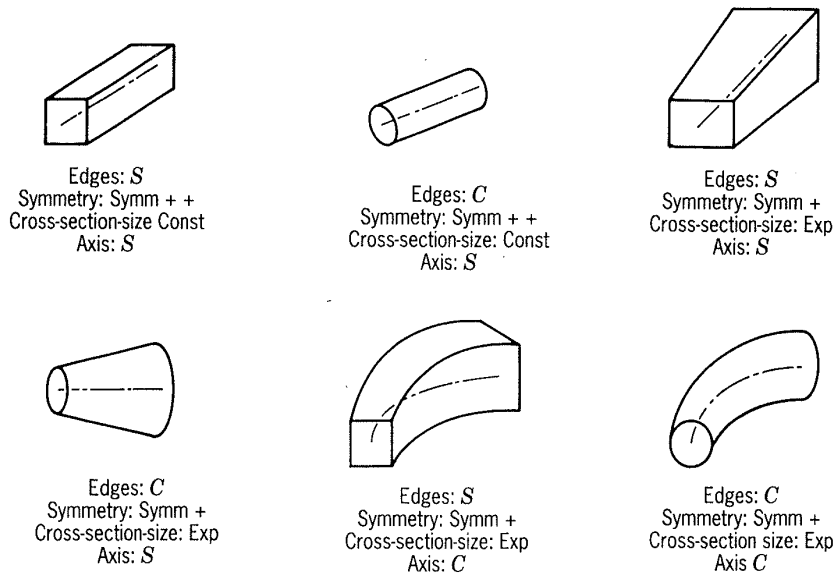


Figure 5: Example of feature extraction for 3-D figures.

Samples of generalized cylinder representations, together with corresponding qualitative features, are shown in Figure 5.

The generalized cylinder approach may be used in a hierarchical pattern representation scheme in which generalized cylinders are primitives and composition rules specify the relative orientations of the primitives [Schalkoff 1989]. ■

EXAMPLE 2: Generating RST Invariant Features and Application to 2-D Figure Recognition

Background. A common PR application involves the recognition of 2-D shapes from

visual (image) data. An example of several classes of shapes, hereafter referred to as regions, is shown in Figure 6. We show the approach and note the processing complexity required to achieve RST invariance.

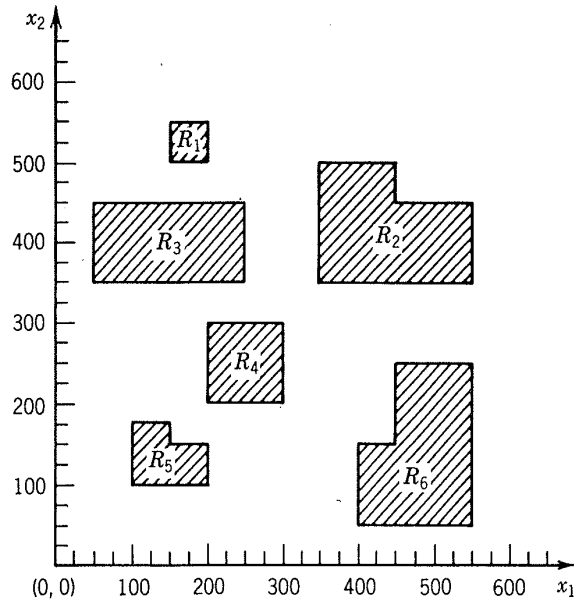


Figure 6: Example of 2-D regions for RST feature extraction.

Moments are extracted features that are derived from the raw measurements and that, in 2-D, may be used to achieve rotation (R), scale (S), and translation (T) (or position) invariant. Observe from Figure 6 that several regions are scaled, rotated, and translated versions of one another. ■

Feature Extraction

For simplicity, the strategy is shown for the continuous case. Given 2-D input data $f(x, y)$, a set of moment features [Hu 1961] is defined via

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad p, q = 0, 1, 2, \dots \quad (1-1)$$

So called *central moments*, μ_{pq} are defined via

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \hat{x})^p (y - \hat{y})^q f(x, y) dx dy \quad p, q = 0, 1, 2, \dots \quad (1-2)$$

where $\hat{x} = m_{10}/m_{00}$ and $\hat{y} = m_{01}/m_{00}$. As shown in [Schalkoff 1989], the central moment-based features convey descriptive information about a region. Normalizing the μ_{pq} features yields

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{[(p+q)/2]+1}} \quad p + q = 2, 3 \dots \quad (1-3)$$

From (1-3) a set of seven RST-invariant features, ϕ_1 – ϕ_9 , may be derived. They are shown in Table 1.1.

Table 1.1: Moment-Based RST Invariant Features

$\phi_1 = \eta_{20} + \eta_{02}$
$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$
$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$
$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$
$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]$ $+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$
$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$
$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]$ $-(\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$

Numerical Results and Analysis

Table 1.2 shows ϕ_1 – ϕ_9 corresponding to each of the six (binary) regions, R_1 – R_6 in Figure 6. Note from Figure 6:

- R_1 and R_4 , and R_2 and R_5 , are *scaled* versions of one another;
- R_6 is a *rotated* version of R_2 and a *scaled and rotated* version of R_5 .

Table 1.2: Moment-Based Features ϕ_i for $R_j = 1, 2, \dots, 7$; $j = 1, 2, \dots, 6$

Feature	R_1	R_2	R_3	R_4	R_5	R_6
ϕ_1	1.67E-01	1.94E-01	2.083E-01	1.67E-01	1.94E-01	1.94E-01
ϕ_2	0.00E+00	6.53E-03	1.56E-02	0.00E+00	6.53E-03	6.53E-03
ϕ_3	0.00E+00	1.02E-03	0.00E+00	0.00E+00	1.02E-03	1.02E-03
ϕ_4	0.00E+00	4.56E-05	0.00E+00	0.00E+00	4.56E-05	4.56E-05
ϕ_5	0.00E+00	4.25E-09	0.00E+00	0.00E+00	4.25E-09	4.25E-09
ϕ_6	0.00E+00	1.70E-06	0.00E+00	0.00E+00	1.70E-06	1.70E-06
ϕ_7	0.00E+00	-8.85E-09	0.00E+00	0.00E+00	-8.85E-09	-8.85E-09

Analysis

From Table 1.2 we observe the following:

- Comparing the ϕ_i for R_1 and R_4 and R_2 and R_5 we verify that the ϕ_i are invariant to S and T .
- Comparing the ϕ_i for R_2 , R_5 and R_6 we verify that the ϕ_i are invariant to R and S .
- Noting that R_3 is a shape class that is not an RS or T variant of any other R_i , we verify that the ϕ_i for R_3 provide numerical features that allow discrimination *between classes*.

Extensions of this feature extraction and classification technique are shown in [Schalkoff 1989], Chapter 6.

What's Really Different Among Different Patterns?

Since the objective of *pattern recognition or classification* is to distinguish between different types of patterns, it is logical to investigate the basis for this discrimination ability. One intuitive answer is that different patterns (different pattern classes) are composed of different features or features with different numerical values. Unfortunately, this is not generally true. For example, consider a newspaper picture or so-called *halftone*. The picture consists of an array of dots, each being either 'white' or 'black.' If we consider these dots to be the basic features or primitives, we observe that all pictures are made up of the same features. However, in attempting to distinguish between a picture of a car and a picture of a boat, we note that the *spatial arrangements* of these features are decidedly different.

The Concept of Similarity. Much of StatPR, SyntPR, and NeurPR are based on the concept of *pattern similarity*. For example, if a pattern, x , is very similar to other patterns known to belong to class w_1 , we would intuitively tend to classify x as belonging in w_1 . *Quantifying similarity* by developing suitable *similarity measures* is often quite difficult. Universally applicable similarity measures that enable good classification are both desirable and elusive. Appendix 5 explores this concept further.

The Feature Vector and Feature Space

It is often useful to develop a geometric viewpoint of features, especially in the StatPR case. Features are arranged in a d -dimensional³ *feature vector*, denoted $\underline{x}$, which yields a multidimensional measurement *space* or *feature space*. If each feature is an unconstrained real number, the feature space is R^d . In other instances, such as the artificial neural network units we consider, it is convenient to restrict feature space to a subspace of R^d . Specifically, if individual neuron outputs and network inputs are restricted to the range $[0, 1]$, for a d -dimensional feature vector, we have a pattern space that is a unit volume hypercube in R^d . Often classification is accomplished by partitioning feature space into regions for each class, as is shown below. Large feature vector dimensionality compounds feature extraction and application of direct pattern recognition approaches. For example, in image processing applications, it is impractical to use directly all the pixel intensities in an image as a feature vector, since a 512×512 pixel image yields a $262,144 \times 1$ feature vector!

Feature vectors are typically used in StatPR and NeurPR. Feature vectors are somewhat inadequate or, at least, cumbersome when it is necessary to represent relations between pattern components. Figure 7 shows how feature extraction is a common, yet different, problem for each PR application and approach.

³Note that often n is chosen as the dimension of the feature vector. To be consistent with other references, we use d and reserve n for other things, such as number of samples in the training set.

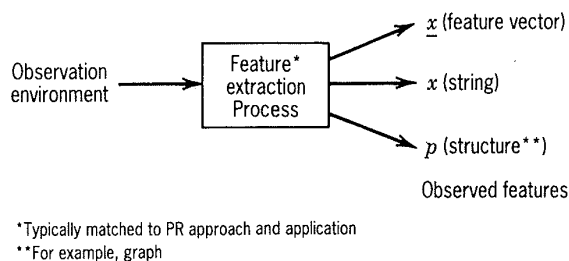


Figure 7: Input data preprocessing/feature extraction.

Scatter Plots. Scatter plots are plots of sample feature vectors, $\underline{x}$, in feature space. An example is shown in Figure 8. Where applicable, they are excellent visualization tools for determining feature vector distribution in R^d , where $d \leq 3$. Scatter plots often facilitate identifying natural or obvious clustering of class-specific feature data and the partitioning of R^d into 'decision regions' for classification.

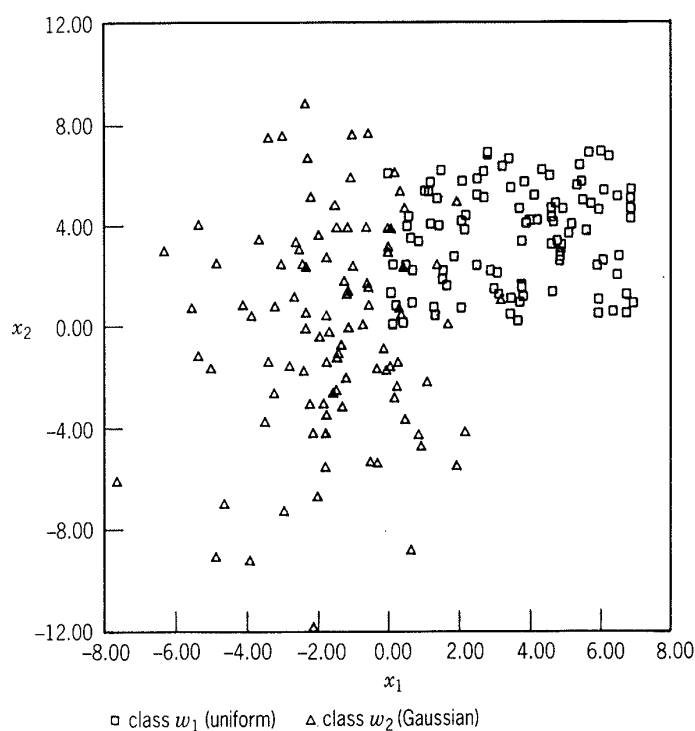


Figure 8: $c = 2$ class scatter plot example.

Definitions

Classification assigns input data into one or more of c prespecified classes based on extraction of significant features or attributes and the processing or analysis of these

attributes. It is common to resort to probabilistic or grammatical models in classification.

Recognition is the ability to classify. Often we formulate PR problems with a $c+1$ st class, corresponding to the ‘unclassifiable’ or ‘don’t know’ or ‘can’t decide’ class.

Description is an alternative to classification where a structural description of the input pattern is desired. It is common to resort to linguistic or structural models in description.

A *pattern class* is a set of patterns (hopefully sharing some common attributes) known to originate from the same source in C . The key in many PR applications is to identify suitable attributes (e.g., features) and to form a good measure of similarity and an associated matching process.

Preprocessing is the filtering or transforming of the raw input data to aid computational feasibility and feature extraction and minimize noise.

Noise is a concept originating in communications theory. In PR, the concept is generalized to represent a number of nonideal circumstances, including:

- Distortions or errors in the input signal/pattern (e.g., measurement errors)
- Errors in preprocessing
- Errors in feature extraction
- Errors in training data

Classifiers, Decision Regions and Boundaries, and Discriminant Functions

The concepts of decision regions are most applicable to StatPR and NeurPR. A *classifier* partitions feature space into class-labeled *decision regions*. In order to use decision regions for a possible and unique class assignment, these regions must cover R^d and be disjointed (nonoverlapping). An exception to the last constraint is the notion of fuzzy sets [Zadeh 1975]. The border of each decision region is a *decision boundary*. With this viewpoint, classification of feature vector $\underline{x}$ becomes quite simple: We determine the decision region (in R^d) into which $\underline{x}$ falls, and assign $\underline{x}$ to this class. Although the classification strategy is straightforward, the determination of decision regions is a challenge. It is sometimes convenient, yet not always necessary (or possible), to visualize decision regions and boundaries. Moreover, computational and geometric aspects of certain decision boundaries (e.g., linear classifiers that generate hyperplanar decision boundaries) are noteworthy.

In what follows, we also design a number of classifiers based on *discriminant functions*. In the c -class case, discriminant functions, denoted $g_i(\underline{x})$, $i = 1, 2, \dots, c$, are used to partition R^d as follows.

Decision Rule: Assign $\underline{x}$ to class w_m (region R_m), where $g_m(\underline{x}) > g_i(\underline{x}) \forall i = 1, 2, \dots, c$ and $i \neq m$.

Note that the case where $g_k(\underline{x}) = g_l(\underline{x})$ defines a decision boundary. Figures 9 and 10 show sample examples.

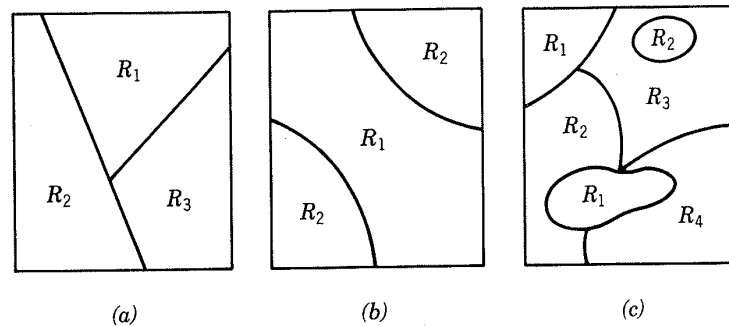
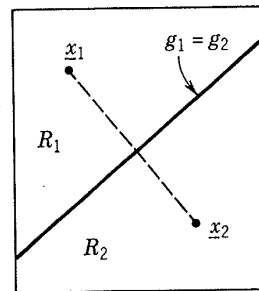


Figure 9: Sample decision regions (2-D example).
 (a) Linear (piecewise).
 (b) Quadratic (hyperbolic).
 (c) (Relatively) general.

$$g_1(\underline{x}) = \|\underline{x} - \underline{x}_1\|$$

$$g_2(\underline{x}) = \|\underline{x} - \underline{x}_2\|$$

(a)



(b)

Figure 10: Discriminant function and corresponding decision regions (minimum distance classifier).
 (a) Discriminant functions.
 (b) Corresponding partition of R^2 .

A particularly important discriminant function form is the *linear discriminant function*:

$$g_i(\underline{x}) = \underline{w}_i^T \underline{x} + w_{0i}$$

where $\underline{w}_i$ is a $d \times 1$ vector of weights used for class i . Appendix 6 shows that this function leads to decision boundaries that are hyperplanes. One important link between StatPR and NeurPR is the use of neural networks to implement discriminant functions.

TRAINING AND LEARNING IN PR SYSTEMS

Using A Priori Knowledge or 'Experience'

In each of the PR approaches we consider, it makes sense to employ the maximum amount of available information in designing the PR system. Intuitively, this would

seem to lead to the best achievable system performance. We assume that a certain amount of a priori information, such as sample patterns with known class origin, is available to the PR system designer. In fact, it might appear quite foolish to attempt the design of a PR system without this information.⁴

A set of 'typical' patterns, where typical attributes or the class or structure of each is known, forms a database denoted the *training set* denoted H . In a general sense, the training set provides significant information on how to associate input data with output decisions (i.e., classifications or structural descriptions). Training is often associated (or loosely equated) with learning. The training set is used to enable the system to 'learn' relevant information, such as statistical parameters, natural groupings, key features, or underlying structure. In SyntPR, training samples may be used to learn or infer grammars.

Learning Curves (Improved 'Performance')

In areas such as artificial intelligence, learning takes on a more general connotation, somewhat analogous to the self-adaptation processes used by humans [Michalski et al. 1986], [Schalkoff 1990, Chapter 18]. A learning system may adapt its internal structure to achieve a better response, perhaps on the basis of previous quantified performance. A performance measure could be the difference between desired and actual system output. This generic learning concept is related to the error-correction-based PR techniques we employ in developing linear discriminant functions in StatPR and the generalized delta rule in NeurPR. Both of these techniques are typical of gradient-descent techniques, where the system is modified following each experiment or iteration. This may lead to the typical 'learning curve' behavior in biological experiments, where $P(n)$ denotes the probability of the subject (animal or human) making the correct response in the n th trial of a learning experiment. A typical formula [Bolles 1979] to predict this behavior, which often matches experimental results, is

$$P(n) = 1 - [1 - P(1)](1 - \theta)^{n-1} \quad n \geq 1$$

where θ is a learning parameter. This is shown in Figure 11. Unfortunately, the monotonically increasing performance shown in this figure is often difficult to achieve in practical PR systems.

Training Approaches

Training uses representative (and usually labeled) samples of types of patterns to be encountered in the actual application.⁵ The training set is denoted H or H_i , where the subscript denotes a training set for a specific pattern class. In this context, *supervised*

⁴This is not always true. For example, see unsupervised learning (Chapter 5).

⁵An important extension of this definition, which is especially useful in grammatical inference and reasoning-based PR learning approaches, is that the training set for class w_i contains examples of patterns in w_i (positive exemplars) as well as examples of patterns not in w_i (negative exemplars).

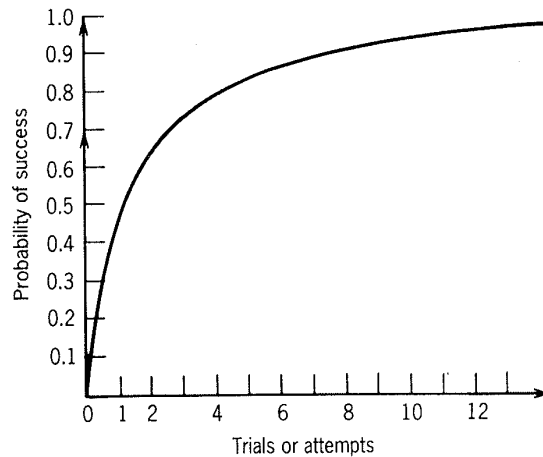


Figure 11: Typical learning curve.

learning assumes a labeled (with respect to pattern class) set, whereas in *unsupervised* learning, the elements of H do not have class labels and the system must determine 'natural' partitions of the sample data.

Linear Separability of the Training Set. Suppose we are given the labeled training set $H = \{\underline{x}_1, \underline{x}_2, \dots, \underline{x}_n\}$, where the class labeling partitions $H = \{H_1, H_2, \dots, H_c\}$. Sample $\underline{x}_i$ is assigned to partition H_j if the label on $\underline{x}_i$ is w_j . If a set of *linear discriminant functions* (also referred to as linear machines) can be used to correctly classify every element of H (or, equivalently, generate the decision regions that accomplish this), then a *linearly separable* problem results. A useful lemma is [Nilsson 1965] if H is linearly separable, then the partitions of H , that is, $H_1, H_2, \dots, H_c$, are *pairwise linearly separable*.

PATTERN RECOGNITION APPROACHES

Pattern recognition applications come in many forms. In some instances, there is an underlying and quantifiable *statistical basis* for the generation of patterns. In other instances, the underlying *structure* of the pattern provides the information fundamental for PR. In still others, neither of the above cases hold true, but we are able to develop and 'train' an architecture to correctly associate input patterns with desired responses. Thus, a given problem may allow one or more of these different solution approaches. When faced with solving a problem, one of the guiding principles an engineer or a scientist should apply is, loosely speaking, 'Use the right tool for the job!'

The Statistical Pattern Recognition Approach (StatPR)

Statistical (or 'decision-theoretic') PR assumes, as its name implies, a statistical basis for classification of algorithms. A set of characteristic measurements, denoted fea-

tures, are extracted from the input data and are used to assign each feature vector to one of c classes. Features are assumed generated by a state of nature, and therefore the underlying model is of a state of nature or class-conditioned set of probabilities and/or probability density functions.

The Syntactic Pattern Recognition Approach (SyntPR)

Many times the significant information in a pattern is not merely in the presence or absence, or the numerical values, of a set of features. Rather, the interrelationships or interconnections of features yield important *structural* information, which facilitates structural description or classification. This is the basis of syntactic (or structural) pattern recognition. However, in using SyntPR approaches, we must be able to quantify and extract structural information and to assess structural similarity of patterns. One syntactic approach is to relate the structure of patterns with the syntax of a formally defined language, in order to capitalize on the vast body of knowledge related to pattern (sentence) generation and analysis (parsing).

Typically, SyntPR approaches formulate hierarchical descriptions of complex patterns built up from simpler subpatterns. At the lowest level, *primitive elements* or 'building blocks' are extracted from the input data. One discrepancy or distinguishing characteristic of SyntPR involves the choice of primitives. *Primitives must be subpatterns or building blocks, whereas features are any measurements.*

Structural PR Example. Musical patterns are one of the best examples in which *structural information* is paramount. At the risk of being overly simplistic,⁶ we observe that all (Western) music basically consists of the same elemental features—an octave is subdivided into distinct tones, and we use about 6 octaves. Therefore, the feature set for all types of music is about 72 distinct tones, which are common to all musical classes. It is the temporal arrangement and structure of these notes that defines the music.

Is SyntPR Dead? It is probably worth asking this question, since there are PR observers who adamantly believe that SyntPR is a 'fading' field. It is true that the pattern representation capability of tractable formal grammars is somewhat limited, chiefly because of parsing difficulties. Advances in formal language research may alleviate this limitation. However, SyntPR encompasses a much broader spectrum of approaches, including representations such as trees and attributed graphs (Chapter 8). The SyntPR approach using graphical descriptions of structure appears to be gaining popularity, especially in areas like model-based vision and AI. Indeed, many applications exist where StatPR representational capabilities are limited and are surpassed by SyntPR capabilities. Examples might be speech recognition and handwriting analysis. Furthermore, recent studies indicate that powerful pattern representation capabilities may be achieved by combining StatPR and SyntPR in a syntactic-semantic approach [Fu 1986].

⁶Of course, timbre and other features also play a role.

The Neural Pattern Recognition Approach (NeurPR)

Modern digital computers do not emulate the computational paradigm of biological systems. The alternative of *neural computing* emerged from attempts to draw on knowledge of how biological neural systems store and manipulate information. This leads to a class of *artificial neural systems* termed *neural networks*. This study involves an amalgamation of research in many diverse fields such as psychology, neuroscience, cognitive science, and systems theory, and has recently received considerable renewed worldwide attention. Neural networks are a relatively new computational paradigm. It is probably safe to say that the advantages, disadvantages, applications, and relationships to traditional computing are not fully understood. Expectations (some might say 'hype') for this area are high. As shown in Chapters 11 to 13, neural networks are particularly well suited for pattern association applications. The notion that artificial neural networks can solve all problems in automated reasoning, or even all PR problems, is probably unrealistic. Neural networks are considered in this text as a means for implementing pattern recognition systems.

Comparing and Relating StatPR, SyntPR, and NeurPR

The boundaries between StatPR, SyntPR, and NeurPR are fuzzy and fading. They share common features and common goals. Often, given a specific PR problem, we choose one approach over another based on analysis of underlying *statistical* components (StatPR), underlying *grammatical* structure (SyntPR), as well as suitability of a *neural network* solution (and training ability), and perhaps lack of suitable statistical or structural models (e.g., 'black box').

In the StatPR or decision theoretic approach, the structure of the pattern is often deemed insignificant. However, the structure *could* be reflected by a suitable choice of features (e.g., a binary feature vector could indicate the presence or absence of observed relations). Similarly, the NeurPR approach is, in some cases, an implementation derived from StatPR and SyntPR approaches. When explicit structural information about the patterns is available, it makes sense to choose SyntPR. When this information is either unavailable or irrelevant, then StatPR may be used. Many practical PR applications fall between these two extremes. For example, an attributed grammar [Tsai/Fu 1980] provides a means to combine statistical and syntactic approaches. This is explored in Chapter 8.

Neural pattern classification is an emerging area with roots dating back approximately 30 years. It is still unclear whether we should consider NeurPR as a novel concept in pattern recognition (or computing) or merely a set of alternative techniques for the implementation of statistical or structural approaches. Table 1.3 and Figures 12 through 15 summarize the different PR approaches.

Examples of Pattern Recognition Approaches

The remainder of the text provides a detailed investigation of individual PR approaches. Three simple examples are shown here to give the reader a flavor of the approaches, as well as their differences.

Table 1.3: Comparing StatPR, SyntPR, and NeurPR Approaches

	StatPR	SyntPR	NeurPR
1. Pattern Generation (Storing) Basis	Probabilistic Models	Formal Grammars	Stable State or Weight Array
2. Pattern Classification (Recognition/Description) Basis	Estimation/ Decision Theory	Parsing	Based on (Predictable) Properties of NN
3. Feature Organization	Feature Vector	Primitives and Observed Relations	Neural Input or Stored States
4. Typical Learning (Training) Approaches			
<i>Supervised:</i>	Density/distribution estimation (usually parametric)	Forming grammars (heuristic or grammatical inference)	Determining NN system parameters (e.g., weights)
<i>Unsupervised:</i>	Clustering	Clustering	Clustering
5. Limitations	Difficulty in expressing structural information	Difficulty in learning structural rules	Often little semantic information from network

StatPR

EXAMPLE 3: Elementary Application of Statistical Pattern Recognition to Image Data (Remote Sensing)

Landsat Image Features and Land Type Determination. Satellite-based sensors in orbit around the earth are one of the largest sources of quantitative image data. The acquired earth images, which may be spectrally distributed, contain information that has significant military, economic, and humanitarian application. For example, the identification of earth areas containing likely mineral deposits or the classification of areas on the basis of vegetation (i.e., specific crops) is possible. A comprehensive bibliography of remote sensing research is presented in [Land 1981].

A series of satellites used for earth remote sensing constitute the Landsat program, initiated in 1972 and somewhat defunct in 1990. Landsat satellites are equipped with multispectral scanner systems (MSS), which generate a feature vector for every pixel in the Landsat image, with the i th component of the feature vector representing the intensity response in a particular spectrum at this pixel location. For example, Table 1.4 shows nominal spectral parameters for the early generation Landsat sensors.

Useful applications of Landsat data are in land use classification and cartography. Development of a map that indicates type of land (e.g., sand, trees, mountains, and

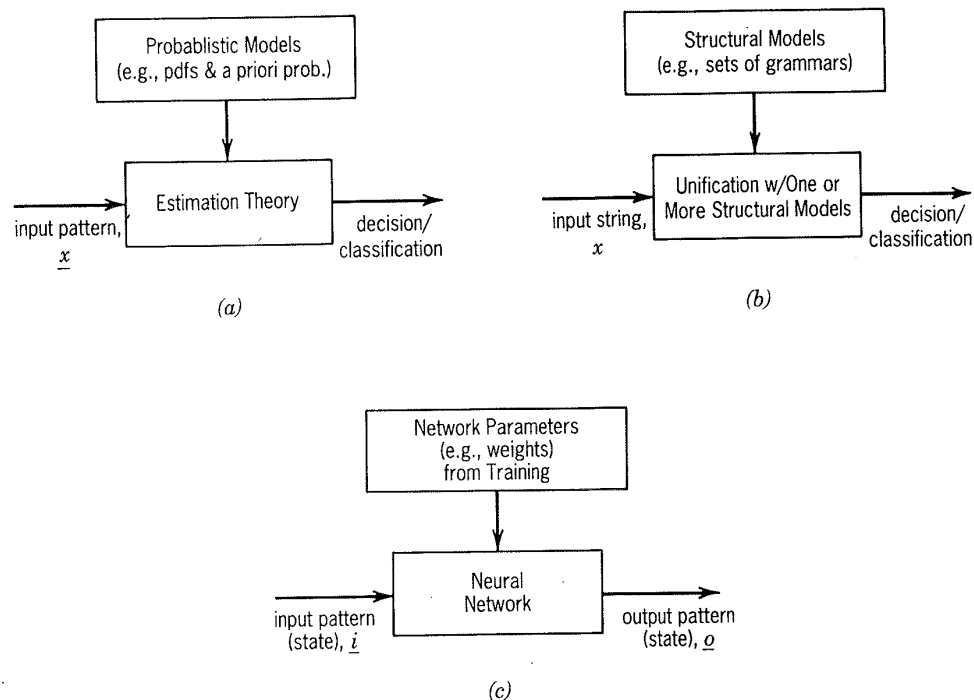


Figure 12: PR approaches.

- (a) Statistical PR (StatPR) approach.
- (b) Syntactic PR (SyntPR) approach.
- (c) Neural network (NeurPR) approach.

water) imaged is desirable. Because of the different spectral responses of these types of land (e.g., the response of sand in the spectra shown in Table 1.4 is different from that of water), it is possible to postulate a class-specific model, train the model, and then allow the system to autonomously classify pixels into one of these regions. As shown in [Land 1981], experiments with Landsat image data, using these four features

Table 1.4: Nominal Landsat Spectral Parameters for Image Point-Specific Features

Feature	Wavelength	Band
x_1	0.50–0.60 μm	visible
x_2	0.60–0.70 μm	visible
x_3	0.70–0.80 μm	near IR
x_4	0.80–1.10 μm	near IR

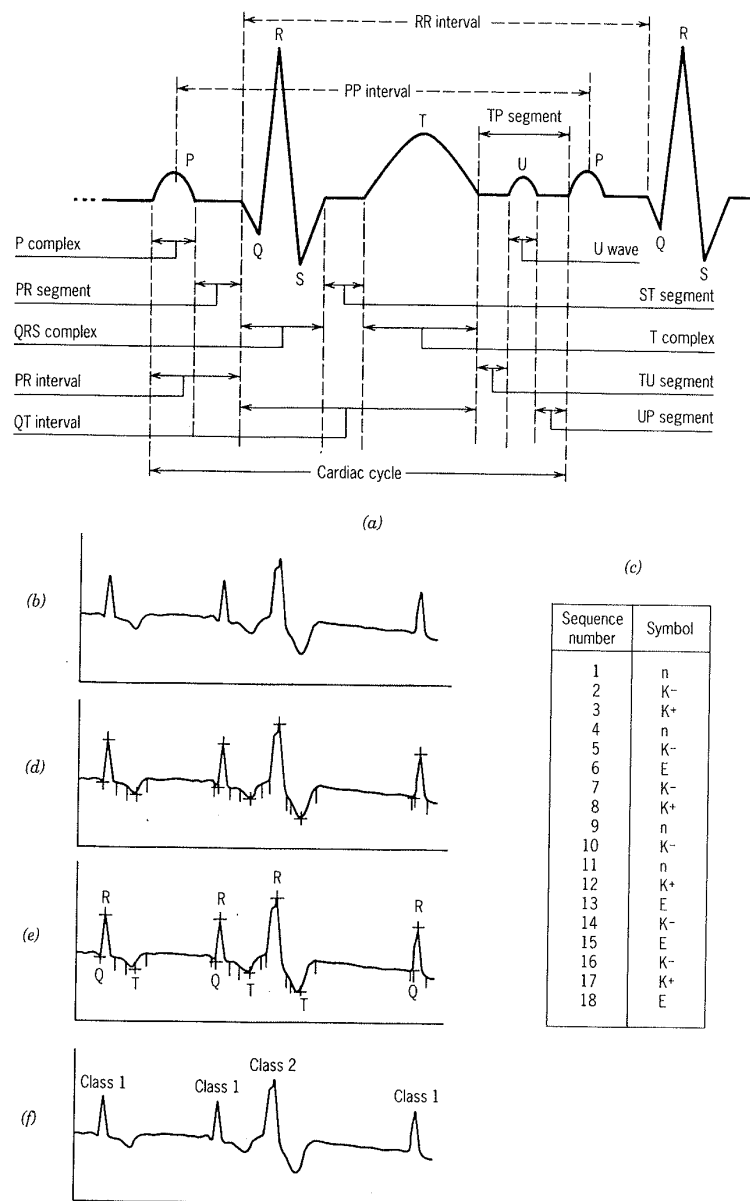


Figure 13: Use of SyntPR for ECG waveform description and classification (from [Trahanias/Skordalakis 1990]).

- (a) A cardiac cycle and its constituent patterns.
- (b) Input waveform.
- (c) Extracted primitives from waveform of (b).
- (d) Graphical display of extracted primitives from (c).
- (e) Recognized ECG cycles and constituent patterns from (d).
- (f) Classification of QRS complexes from ECG cycles of (e).

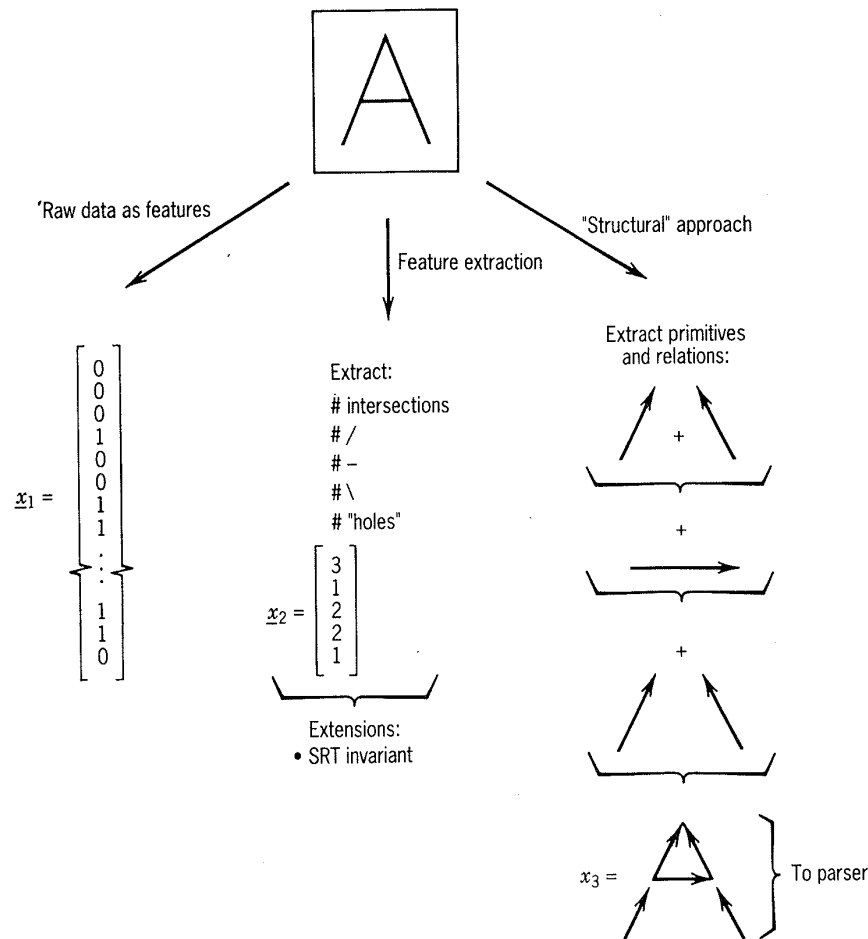


Figure 14: Example of PR approaches.
 (a) Input pattern.
 (b) Possible feature extraction approaches.

alone (following a training phase), yield a classification error of approximately 17%. Since we have considered only *spectral* features, it is natural to attempt even smaller error rates by extraction of other features, for example, *spatial* and *temporal* features.⁷ Spatial features typically used in Landsat data classification, and more generally image segmentation, include *texture*, *context*, *shape*, and *structural relationships*. Note that generally an increase in classification accuracy requires a corresponding increase in the computational cost of the classification procedure. ■

⁷This leads to *contextual classification*.

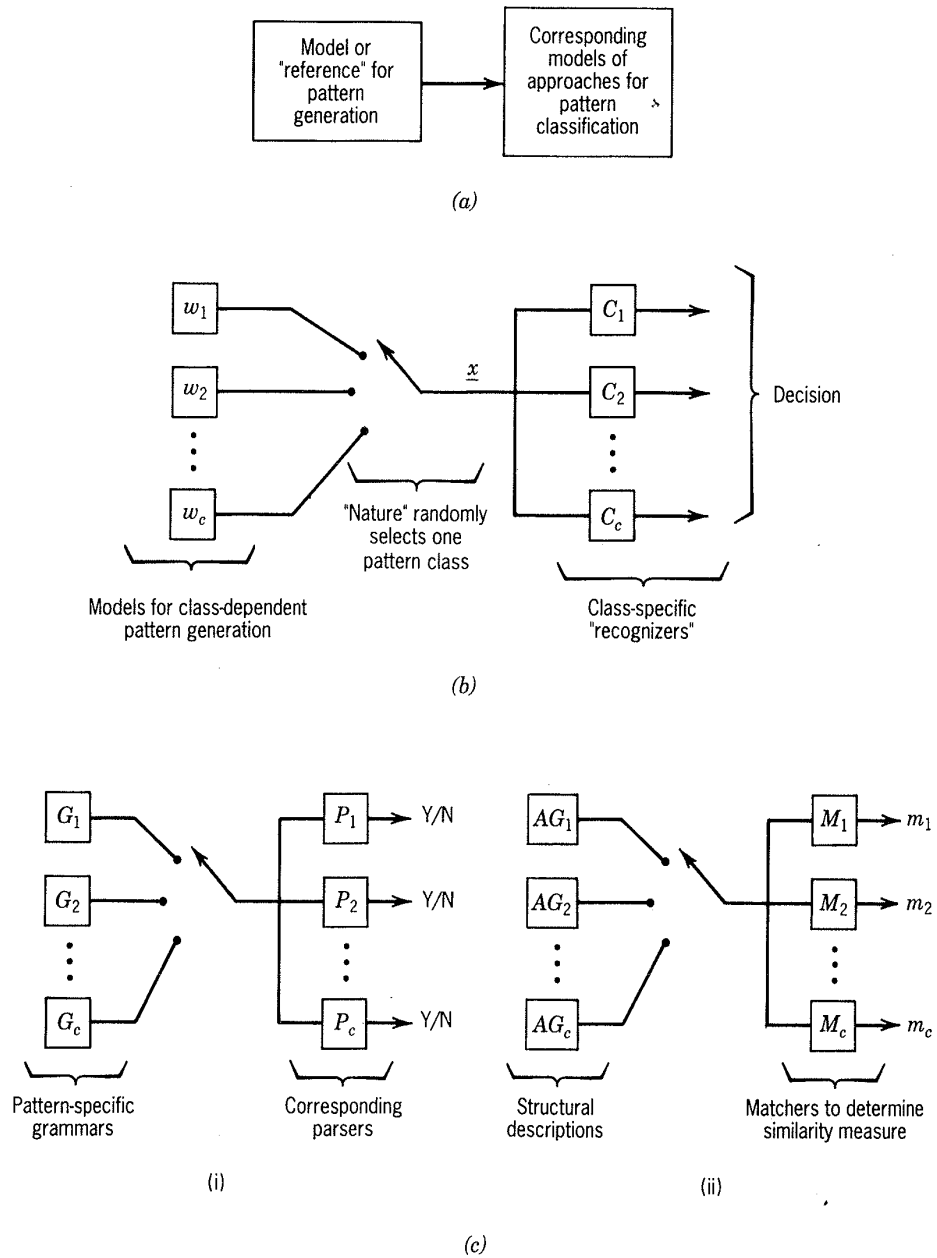


Figure 15: Relating StatPR and SyntPR in the context of model-based PR.

(a) Overall structure.

(b) StatPR example.

(c) SyntPR examples.

(i) Grammatical approach.

(ii) Attributed-graph approach.

SyntPR

EXAMPLES 4 AND 5:

Suppose we are given an input pattern consisting of a string of characters, each of which is either 'a', 'b', or 'c'. These characters may be assignments resulting from the preprocessing of raw input patterns, such as EKG waveforms [Trahanis/Skordalakis 1990] or speech signals. Two states of nature generate two classes of patterns, each characterized by its *structure*. Samples are shown below.

Table 1.5: Typical Patterns (Strings)

Class 1	Class 2
<i>ac</i>	<i>bc</i>
<i>abc</i>	<i>abc</i>
<i>abbc</i>	<i>aabc</i>
<i>abbbc</i>	<i>aaabc</i>

Observe that the class of patterns generated by Class 1 is of the form ab^nc $n \geq 0$ whereas the form or *structure* of Class 2 is a^nbc $n \geq 0$. In addition, note that several patterns belonging to both classes may be generated. In a classification application of SyntPR, given a pattern such as 'aaabbc', the goal would be to determine which, if any, class could have produced this pattern.

Figure 13 (from [Trahanias/Skordalakis 1990]) illustrates the application of a syntactic approach to the analysis of electrocardiograms. To facilitate waveform analysis, chosen primitives are waveform positive and negative peaks, straight line segments, and parabolic segments. An alphabet of symbols $V_T = \{K^+, K^-, E, \Pi\}$ is used to denote these primitives for a given wave. Primitive extraction thus yields a linguistic representation consisting of a sequence of these symbols, in a form similar to that shown in Table 1.5. Subsequences that comprise patterns are identified and classified, as shown in the figure. ■

EXAMPLE 6: Combinations of StatPR and SyntPR

Figure 14 shows a simple character recognition problem wherein a spectrum of PR approaches is possible. As shown, these approaches range from little feature extraction to extraction of complex primitives and accompanying spatial relations.

Figure 15 further relates the StatPR and SyntPR approaches from the viewpoint of model-based pattern generation and 'inversion' of this process in classification. The reader should compare the approaches shown in Figure 11 with the abstract PR formulation shown in Figure 1. ■

Procedures for PR System Engineering

In realistic applications, the design of an autonomous PR system is a complex, usually iterative and interactive task. Although it is impossible to provide an all-inclusive algo-

rhythmic procedure, the highly interrelated, skeletal steps outlined here reflect typical efforts and concerns.

- Step 1:** Study the classes of patterns under consideration to develop possible characterizations. This includes assessments of (quantifiable) pattern structure and probabilistic characterizations, as well as exploration of possible within-class and interclass similarity/dissimilarity measures. In addition, possible pattern deformations or invariant properties and characterization of 'noise' sources should be considered at this point.
- Step 2:** Determine the availability of feature/measurement data.
- Step 3:** Consider constraints on desired system performance and computational resources (e.g., parts/minute, classification accuracy).
- Step 4:** Consider the availability of training data.
- Step 5:** Consider the availability of suitable and known PR techniques (e.g., StatPR, SyntPR, and clustering). an overall PR system structure.
- Step 6:** Develop a PR system simulation. This may involve choosing models, grammars, or network structures.
- Step 7:** Train the system.
- Step 8:** Simulate system performance.
- Step 9:** Iterate among the above steps until desired performance is achieved.

OTHER APPROACHES TO PR

'Black Box' Approaches

Both StatPR and SyntPR consider, to some extent, formal mechanisms to convey how patterns might be produced. In StatPR, statistical models are often used, whereas in SyntPR, structural or grammatical models are used.

Often, the quantitative aspects of a complex PR problem are largely unknown, defy quantification, and/or are so complex that detailed modeling is hopeless. An (engineering) alternative to handling problems of this type is to avoid modeling of the pattern-generating mechanism and, instead, to treat the problem from an input/output or 'black box' viewpoint. A black box system is specified by two things: (1) an internal computation; and (2) a stimulus/response (S-R) based training set (e.g., a set of input/output pattern pairs). For example, the human brain is, for purposes of present artificial intelligence implementations, a black box. We are able to observe and emulate intelligent behavior (including pattern recognition and classification) without a detailed set of algorithms that quantify the input-output characteristics of the brain. NeurPR often provides good examples of black box systems in cases where we do not care about the 'internal' semantics of the network but rather simply use training data to achieve good pattern mappings.

Note that under certain assumptions a StatPR approach may be viewed from the black box perspective. It is important to note that the black box implements a *computation*, but this computation may be structurally unrelated to the particular application. For example, the particular application may be reflected only in a choice of parameters used in the computation.

Reasoning-Driven Pattern Recognition

We could argue that all autonomous systems (in particular those that emulate human PR capability) make decisions based on *evidence*. This evidence may exist at varying levels; for example, it may consist of raw feature data (measured voltages, currents, temperatures, and the like), or at a higher level, perhaps as a result of preprocessing. Furthermore, the evidence need not simply come from external inputs; the use of stored information or knowledge may aid in the generation of evidence. The manipulation of evidence forms the basis of many approaches inherent in pattern recognition, learning machines, and expert systems.

Pattern Recognition and Artificial Intelligence. Certain PR problems have a strong relationship to techniques of artificial intelligence (AI). Numerous research efforts to develop 'expert systems' for PR applications are underway ([Schalkoff 1990], Chapter 13). Consider, for example, the 'incomplete' pattern in Figure 16, which shows a small portion of an object. The human observer, through a somewhat difficult-to-quantify inference process, both recognizes and completes the pattern. We explore a somewhat reduced-order version of this process in the graphical (model-driven) approaches to SyntPR. We note in closing our brief discussion of reasoning-driven PR that emulation of human or biological processes is neither always possible nor desirable. For example, humans have mastered the concept of flight without emulating the mechanisms used by birds. One especially important area of PR and AI overlap is that of classifier design from labeled training data (PR) and the derivation of rules from large data sets [Bundy et. al. 1985], [Goodman/Smyth 1990]. In this case, a set of n labeled samples, each consisting of a d -tuple of features or attributes (which may be symbolic instead of numerical values, such as color) $a_1, a_2 \dots a_c$ and a corresponding classification label, w_c , are assumed given. In the PR domain, the objective is to determine a *classifier*, whereas the AI objective is to use the data to infer a set of general *rules* that produce the desired class.⁸

OVERVIEW OF PR LITERATURE AND RESOURCES

Each of the following chapters contains a Bibliographical Remarks section that provides further references that are related to the specific chapter topics. What follows is a summary of general PR resources.

⁸This classifier may take any number of forms.

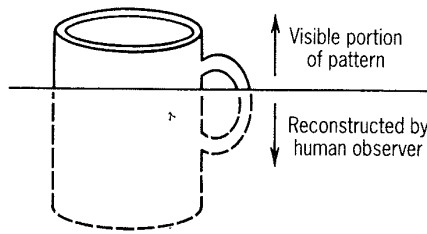


Figure 16: Example of reasoning-driven PR.

Books

The field of PR is thoroughly covered in books devoted to this subject. However, the coverage is usually partitioned along StatPR, SyntPR, or NeurPR lines.

Statistical pattern recognition is explored in depth in books by [Duda/Hart 1973], [Dejver/Kittler 1982], [Fukunaga 1972], [Bow 1984], [Watanabe 1985], [Chien 1978], [Chen 1973], [Patrick 1972], [Young/Calvert 1974], [Sklansky 1973], [Agarwala 1977], [Jain/Dubes 1988], and [Therrien 1989].

Coverage of syntactic pattern recognition approaches are presented in [Fu 1 1982], [Tou/Gonzalez 1974], [Gonzalez/Thomason 1978], [Miclet 1986], and [Pavlidis 1977]. A unified view of StatPR and SyntPR is shown in [Fu 1980]. An interesting and heavily mathematically oriented approach, which explores StatPR and some aspects of reasoning-driven PR and PR applications in Artificial Intelligence, is presented in [Patrick/Fattu 1986].

Fundamental neural network architecture and application book references are [Anderson/Rosenfeld 1988], [Kohonen 1984], [Paö 1989], [Khanna 1990], and [Rummelhart/McClelland 1, 2 1986].

Journals

Work on various aspects of PR continues to cross-pollenate journals. Useful sources include the following:

- *Pattern Recognition* (letters)
- *Pattern Recognition* (journal)
- *IEEE Transactions on Systems, Man and Cybernetics*
- *IEEE Transactions on Pattern Analysis and Machine Intelligence*
- *IEEE Transactions on Acoustics, Speech and Signal Processing*
- *IEEE Transactions on Information Theory*
- *IEEE Transactions on Geoscience and Remote Sensing*
- *IEEE Transactions on Industrial Electronics*
- *IEEE Transactions on Neural Networks*
- *Optical Engineering* (SPIE)

- *Image and Vision Computing*
- *Neural Networks*
- *Applied Optics*

Conferences

Major related and periodic conferences include:

- International Conference on Pattern Recognition (sponsored by International Association for Pattern Recognition)
- IEEE Computer Vision and Image Processing Conference (formerly Pattern Recognition and Image Processing)
- International Joint Conferences on Neural Networks (IJCNN, cosponsored by IEEE).

EXERCISES

- 1.1 Distinguish between a pattern and a signal.
- 1.2 Suppose you were charged with the design of a PR system for character recognition. Suggest some RST invariant features.
- 1.3 Does a knowledge of *similarity* necessarily imply a knowledge of *dissimilarity*?
- 1.4 Distinguish between a feature and a measurement.
- 1.5 Show examples of features that are not primitives and vice versa.
- 1.6 Cite examples of applications where StatPR is more appropriate than SyntPR and vice versa.
- 1.7 Try to relate SyntPR with StatPR by using the SyntPR Example 2. What *features* of the pattern are useful for classification? Do you expect difficulty in statistically characterizing these features?
- 1.8 Repeat Example 2 (moment features for RST invariance) by using just region boundary data or the outline from Figure 6 as the raw data for moment calculations. Are the ϕ_i still RST invariant?
- 1.9 Again referring to Example 2 for the data of Figure 6, consider the use of a single, simpler feature for each R_i , namely,

$$D_i = \frac{A_{R_i}}{(P_{R_i})^2}$$

where A_{R_i} is the *area* of region R_i and P_{R_i} is the corresponding *perimeter*.

- 1.10 For the following set of characters

A B C D E

develop a set of features that allow unique identification of each character. (Hint: Consider possibilities such as total number of segments, number of straight segments, number of curved segments, number of enclosed regions, etc.) Discuss the feasibility of extracting these features.

- 1.11** Consider the recognition of the character 'E' in the presence of a reasonable set of pattern perturbations, including orientation, scale, script (handwritten) versus printed, and the like. What types of features would you extract to achieve recognition invariant to these distortions?