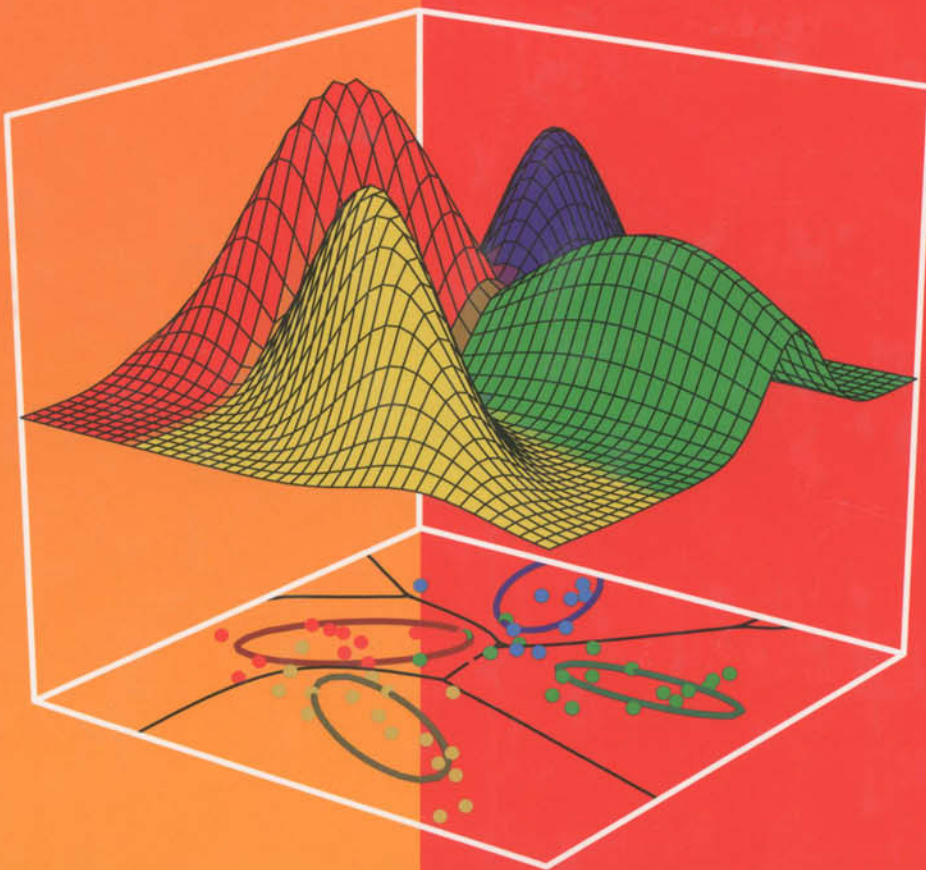


Richard O. Duda  
Peter E. Hart  
David G. Stork

# Pattern Classification



Second Edition



---

# PATTERN CLASSIFICATION

---



---

# PATTERN CLASSIFICATION

---

Second Edition

Richard O. Duda

Peter E. Hart

David G. Stork



A Wiley-Interscience Publication  
**JOHN WILEY & SONS, INC.**

**New York ■ Chichester ■ Weinheim ■ Brisbane ■ Singapore ■ Toronto**

---

This book is printed on acid-free paper.

Copyright © 2001 by John Wiley & Sons, Inc. All rights reserved.

Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4744. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 605 Third Avenue, New York, NY 10158-0012, (212) 850-6011, fax (212) 850-6008. E-Mail: PERMREQ@WILEY.COM.

For ordering and customer service, call 1-800-CALL-WILEY.

***Library of Congress Cataloging-in-Publication Data:***

Duda, Richard O.

Pattern classification / Richard O. Duda, Peter

E. Hart [and] David G. Stork. — 2nd ed.

p. cm.

“A Wiley-Interscience Publication.”

Includes bibliographical references and index.

Partial Contents: Part 1. Pattern classification.

ISBN 0-471-05669-3 (alk. paper)

1. Pattern recognition systems. 2. Statistical decision.

I. Hart, Peter E. II. Stork, David G. III. Title.

Q327.D83 2000

006.4—dc21

99-29981

CIP

Printed in the United States of America

20 19 18 17 16 15 14 13

*To*  
*C. A. Rosen*  
*and*  
*C. W. Stork*





# CONTENTS

---

## PREFACE

xvii

## 1

## INTRODUCTION

---

1

- 1.1 Machine Perception, 1
  - 1.2 An Example, 1
    - 1.2.1 Related Fields, 8
  - 1.3 Pattern Recognition Systems, 9
    - 1.3.1 Sensing, 9
    - 1.3.2 Segmentation and Grouping, 9
    - 1.3.3 Feature Extraction, 11
    - 1.3.4 Classification, 12
    - 1.3.5 Post Processing, 13
  - 1.4 The Design Cycle, 14
    - 1.4.1 Data Collection, 14
    - 1.4.2 Feature Choice, 14
    - 1.4.3 Model Choice, 15
    - 1.4.4 Training, 15
    - 1.4.5 Evaluation, 15
    - 1.4.6 Computational Complexity, 16
  - 1.5 Learning and Adaptation, 16
    - 1.5.1 Supervised Learning, 16
    - 1.5.2 Unsupervised Learning, 17
    - 1.5.3 Reinforcement Learning, 17
  - 1.6 Conclusion, 17
- Summary by Chapters, 17
- Bibliographical and Historical Remarks, 18
- Bibliography, 19

## 2

## BAYESIAN DECISION THEORY

---

20

- 2.1 Introduction, 20
- 2.2 Bayesian Decision Theory—Continuous Features, 24
  - 2.2.1 Two-Category Classification, 25
- 2.3 Minimum-Error-Rate Classification, 26
  - \*2.3.1 Minimax Criterion, 27

vii

- \*2.3.2 Neyman-Pearson Criterion, 28
- 2.4 Classifiers, Discriminant Functions, and Decision Surfaces, 29
  - 2.4.1 The Multicategory Case, 29
  - 2.4.2 The Two-Category Case, 30
- 2.5 The Normal Density, 31
  - 2.5.1 Univariate Density, 32
  - 2.5.2 Multivariate Density, 33
- 2.6 Discriminant Functions for the Normal Density, 36
  - 2.6.1 Case 1:  $\Sigma_i = \sigma^2 \mathbf{I}$ , 36
  - 2.6.2 Case 2:  $\Sigma_i = \Sigma$ , 39
  - 2.6.3 Case 3:  $\Sigma_i = \text{arbitrary}$ , 41
    - Example 1** Decision Regions for Two-Dimensional Gaussian Data, 41
- \*2.7 Error Probabilities and Integrals, 45
- \*2.8 Error Bounds for Normal Densities, 46
  - 2.8.1 Chernoff Bound, 46
  - 2.8.2 Bhattacharyya Bound, 47
    - Example 2** Error Bounds for Gaussian Distributions, 48
  - 2.8.3 Signal Detection Theory and Operating Characteristics, 48
- 2.9 Bayes Decision Theory—Discrete Features, 51
  - 2.9.1 Independent Binary Features, 52
    - Example 3** Bayesian Decisions for Three-Dimensional Binary Data, 53
- \*2.10 Missing and Noisy Features, 54
  - 2.10.1 Missing Features, 54
  - 2.10.2 Noisy Features, 55
- \*2.11 Bayesian Belief Networks, 56
  - Example 4** Belief Network for Fish, 59
- \*2.12 Compound Bayesian Decision Theory and Context, 62
- Summary, 63
- Bibliographical and Historical Remarks, 64
- Problems, 65
- Computer exercises, 80
- Bibliography, 82

- 3.1 Introduction, 84
- 3.2 Maximum-Likelihood Estimation, 85
  - 3.2.1 The General Principle, 85
  - 3.2.2 The Gaussian Case: Unknown  $\mu$ , 88
  - 3.2.3 The Gaussian Case: Unknown  $\mu$  and  $\Sigma$ , 88
  - 3.2.4 Bias, 89
- 3.3 Bayesian Estimation, 90
  - 3.3.1 The Class-Conditional Densities, 91
  - 3.3.2 The Parameter Distribution, 91
- 3.4 Bayesian Parameter Estimation: Gaussian Case, 92
  - 3.4.1 The Univariate Case:  $p(\mu|\mathcal{D})$ , 92
  - 3.4.2 The Univariate Case:  $p(x|\mathcal{D})$ , 95
  - 3.4.3 The Multivariate Case, 95

- 3.5 Bayesian Parameter Estimation: General Theory, 97
  - Example 1 Recursive Bayes Learning, 98
  - 3.5.1 When Do Maximum-Likelihood and Bayes Methods Differ?, 100
  - 3.5.2 Noninformative Priors and Invariance, 101
  - 3.5.3 Gibbs Algorithm, 102
- \*3.6 Sufficient Statistics, 102
  - 3.6.1 Sufficient Statistics and the Exponential Family, 106
- 3.7 Problems of Dimensionality, 107
  - 3.7.1 Accuracy, Dimension, and Training Sample Size, 107
  - 3.7.2 Computational Complexity, 111
  - 3.7.3 Overfitting, 113
- \*3.8 Component Analysis and Discriminants, 114
  - 3.8.1 Principal Component Analysis (PCA), 115
  - 3.8.2 Fisher Linear Discriminant, 117
  - 3.8.3 Multiple Discriminant Analysis, 121
- \*3.9 Expectation-Maximization (EM), 124
  - Example 2 Expectation-Maximization for a 2D Normal Model, 126
- 3.10 Hidden Markov Models, 128
  - 3.10.1 First-Order Markov Models, 128
  - 3.10.2 First-Order Hidden Markov Models, 129
  - 3.10.3 Hidden Markov Model Computation, 129
  - 3.10.4 Evaluation, 131
    - Example 3 Hidden Markov Model, 133
  - 3.10.5 Decoding, 135
    - Example 4 HMM Decoding, 136
  - 3.10.6 Learning, 137
- Summary, 139
- Bibliographical and Historical Remarks, 139
- Problems, 140
- Computer exercises, 155
- Bibliography, 159

## 4

## NONPARAMETRIC TECHNIQUES

161

- 4.1 Introduction, 161
- 4.2 Density Estimation, 161
- 4.3 Parzen Windows, 164
  - 4.3.1 Convergence of the Mean, 167
  - 4.3.2 Convergence of the Variance, 167
  - 4.3.3 Illustrations, 168
  - 4.3.4 Classification Example, 168
  - 4.3.5 Probabilistic Neural Networks (PNNs), 172
  - 4.3.6 Choosing the Window Function, 174
- 4.4  $k_n$ -Nearest-Neighbor Estimation, 174
  - 4.4.1  $k_n$ -Nearest-Neighbor and Parzen-Window Estimation, 176
  - 4.4.2 Estimation of *A Posteriori* Probabilities, 177
- 4.5 The Nearest-Neighbor Rule, 177
  - 4.5.1 Convergence of the Nearest Neighbor, 179
  - 4.5.2 Error Rate for the Nearest-Neighbor Rule, 180
  - 4.5.3 Error Bounds, 180
  - 4.5.4 The  $k$ -Nearest-Neighbor Rule, 182

- 4.5.5 Computational Complexity of the  $k$ -Nearest-Neighbor Rule, 184
- 4.6 Metrics and Nearest-Neighbor Classification, 187
  - 4.6.1 Properties of Metrics, 187
  - 4.6.2 Tangent Distance, 188
- \*4.7 Fuzzy Classification, 192
- \*4.8 Reduced Coulomb Energy Networks, 195
- 4.9 Approximations by Series Expansions, 197
- Summary, 199
- Bibliographical and Historical Remarks, 200
- Problems, 201
- Computer exercises, 209
- Bibliography, 213

## 5

## LINEAR DISCRIMINANT FUNCTIONS

215

- 5.1 Introduction, 215
- 5.2 Linear Discriminant Functions and Decision Surfaces, 216
  - 5.2.1 The Two-Category Case, 216
  - 5.2.2 The Multicategory Case, 218
- 5.3 Generalized Linear Discriminant Functions, 219
- 5.4 The Two-Category Linearly Separable Case, 223
  - 5.4.1 Geometry and Terminology, 224
  - 5.4.2 Gradient Descent Procedures, 224
- 5.5 Minimizing the Perceptron Criterion Function, 227
  - 5.5.1 The Perceptron Criterion Function, 227
  - 5.5.2 Convergence Proof for Single-Sample Correction, 229
  - 5.5.3 Some Direct Generalizations, 232
- 5.6 Relaxation Procedures, 235
  - 5.6.1 The Descent Algorithm, 235
  - 5.6.2 Convergence Proof, 237
- 5.7 Nonseparable Behavior, 238
- 5.8 Minimum Squared-Error Procedures, 239
  - 5.8.1 Minimum Squared-Error and the Pseudoinverse, 240
    - Example 1** Constructing a Linear Classifier by Matrix Pseudoinverse, 241
  - 5.8.2 Relation to Fisher's Linear Discriminant, 242
  - 5.8.3 Asymptotic Approximation to an Optimal Discriminant, 243
  - 5.8.4 The Widrow-Hoff or LMS Procedure, 245
  - 5.8.5 Stochastic Approximation Methods, 246
- 5.9 The Ho-Kashyap Procedures, 249
  - 5.9.1 The Descent Procedure, 250
  - 5.9.2 Convergence Proof, 251
  - 5.9.3 Nonseparable Behavior, 253
  - 5.9.4 Some Related Procedures, 253
- \*5.10 Linear Programming Algorithms, 256
  - 5.10.1 Linear Programming, 256
  - 5.10.2 The Linearly Separable Case, 257
  - 5.10.3 Minimizing the Perceptron Criterion Function, 258
- \*5.11 Support Vector Machines, 259
  - 5.11.1 SVM Training, 263
    - Example 2** SVM for the XOR Problem, 264

- 5.12 Multicategory Generalizations, 265
  - 5.12.1 Kesler's Construction, 266
  - 5.12.2 Convergence of the Fixed-Increment Rule, 266
  - 5.12.3 Generalizations for MSE Procedures, 268
- Summary, 269
- Bibliographical and Historical Remarks, 270
- Problems, 271
- Computer exercises, 278
- Bibliography, 281

## 6

## MULTILAYER NEURAL NETWORKS

282

- 6.1 Introduction, 282
- 6.2 Feedforward Operation and Classification, 284
  - 6.2.1 General Feedforward Operation, 286
  - 6.2.2 Expressive Power of Multilayer Networks, 287
- 6.3 Backpropagation Algorithm, 288
  - 6.3.1 Network Learning, 289
  - 6.3.2 Training Protocols, 293
  - 6.3.3 Learning Curves, 295
- 6.4 Error Surfaces, 296
  - 6.4.1 Some Small Networks, 296
  - 6.4.2 The Exclusive-OR (XOR), 298
  - 6.4.3 Larger Networks, 298
  - 6.4.4 How Important Are Multiple Minima?, 299
- 6.5 Backpropagation as Feature Mapping, 299
  - 6.5.1 Representations at the Hidden Layer—Weights, 302
- 6.6 Backpropagation, Bayes Theory and Probability, 303
  - 6.6.1 Bayes Discriminants and Neural Networks, 303
  - 6.6.2 Outputs as Probabilities, 304
- \*6.7 Related Statistical Techniques, 305
- 6.8 Practical Techniques for Improving Backpropagation, 306
  - 6.8.1 Activation Function, 307
  - 6.8.2 Parameters for the Sigmoid, 308
  - 6.8.3 Scaling Input, 308
  - 6.8.4 Target Values, 309
  - 6.8.5 Training with Noise, 310
  - 6.8.6 Manufacturing Data, 310
  - 6.8.7 Number of Hidden Units, 310
  - 6.8.8 Initializing Weights, 311
  - 6.8.9 Learning Rates, 312
  - 6.8.10 Momentum, 313
  - 6.8.11 Weight Decay, 314
  - 6.8.12 Hints, 315
  - 6.8.13 On-Line, Stochastic or Batch Training?, 316
  - 6.8.14 Stopped Training, 316
  - 6.8.15 Number of Hidden Layers, 317
  - 6.8.16 Criterion Function, 318
- \*6.9 Second-Order Methods, 318
  - 6.9.1 Hessian Matrix, 318
  - 6.9.2 Newton's Method, 319

- 6.9.3 Quickprop, 320
- 6.9.4 Conjugate Gradient Descent, 321
  - Example 1 Conjugate Gradient Descent, 322
- \*6.10 Additional Networks and Training Methods, 324
  - 6.10.1 Radial Basis Function Networks (RBFs), 324
  - 6.10.2 Special Bases, 325
  - 6.10.3 Matched Filters, 325
  - 6.10.4 Convolutional Networks, 326
  - 6.10.5 Recurrent Networks, 328
  - 6.10.6 Cascade-Correlation, 329
- 6.11 Regularization, Complexity Adjustment and Pruning, 330
- Summary, 333
- Bibliographical and Historical Remarks, 333
- Problems, 335
- Computer exercises, 343
- Bibliography, 347

**7****STOCHASTIC METHODS****350**

- 7.1 Introduction, 350
- 7.2 Stochastic Search, 351
  - 7.2.1 Simulated Annealing, 351
  - 7.2.2 The Boltzmann Factor, 352
  - 7.2.3 Deterministic Simulated Annealing, 357
- 7.3 Boltzmann Learning, 360
  - 7.3.1 Stochastic Boltzmann Learning of Visible States, 360
  - 7.3.2 Missing Features and Category Constraints, 365
  - 7.3.3 Deterministic Boltzmann Learning, 366
  - 7.3.4 Initialization and Setting Parameters, 367
- \*7.4 Boltzmann Networks and Graphical Models, 370
  - 7.4.1 Other Graphical Models, 372
- \*7.5 Evolutionary Methods, 373
  - 7.5.1 Genetic Algorithms, 373
  - 7.5.2 Further Heuristics, 377
  - 7.5.3 Why Do They Work?, 378
- \*7.6 Genetic Programming, 378
- Summary, 381
- Bibliographical and Historical Remarks, 381
- Problems, 383
- Computer exercises, 388
- Bibliography, 391

**8****NONMETRIC METHODS****394**

- 8.1 Introduction, 394
- 8.2 Decision Trees, 395
- 8.3 CART, 396
  - 8.3.1 Number of Splits, 397
  - 8.3.2 Query Selection and Node Impurity, 398
  - 8.3.3 When to Stop Splitting, 402
  - 8.3.4 Pruning, 403

- 8.3.5 Assignment of Leaf Node Labels, 404
  - Example 1** A Simple Tree, 404
- 8.3.6 Computational Complexity, 406
- 8.3.7 Feature Choice, 407
- 8.3.8 Multivariate Decision Trees, 408
- 8.3.9 Priors and Costs, 409
- 8.3.10 Missing Attributes, 409
  - Example 2** Surrogate Splits and Missing Attributes, 410
- 8.4 Other Tree Methods, 411
  - 8.4.1 ID3, 411
  - 8.4.2 C4.5, 411
  - 8.4.3 Which Tree Classifier Is Best?, 412
- \*8.5 Recognition with Strings, 413
  - 8.5.1 String Matching, 415
  - 8.5.2 Edit Distance, 418
  - 8.5.3 Computational Complexity, 420
  - 8.5.4 String Matching with Errors, 420
  - 8.5.5 String Matching with the “Don’t-Care” Symbol, 421
- 8.6 Grammatical Methods, 421
  - 8.6.1 Grammars, 422
  - 8.6.2 Types of String Grammars, 424
    - Example 3** A Grammar for Pronouncing Numbers, 425
  - 8.6.3 Recognition Using Grammars, 426
- 8.7 Grammatical Inference, 429
  - Example 4** Grammatical Inference, 431
- \*8.8 Rule-Based Methods, 431
  - 8.8.1 Learning Rules, 433
- Summary, 434
- Bibliographical and Historical Remarks, 435
- Problems, 437
- Computer exercises, 446
- Bibliography, 450

**9****ALGORITHM-INDEPENDENT MACHINE LEARNING****453**

- 
- 9.1 Introduction, 453
  - 9.2 Lack of Inherent Superiority of Any Classifier, 454
    - 9.2.1 No Free Lunch Theorem, 454
      - Example 1** No Free Lunch for Binary Data, 457
    - \*9.2.2 Ugly Duckling Theorem, 458
    - 9.2.3 Minimum Description Length (MDL), 461
    - 9.2.4 Minimum Description Length Principle, 463
    - 9.2.5 Overfitting Avoidance and Occam’s Razor, 464
  - 9.3 Bias and Variance, 465
    - 9.3.1 Bias and Variance for Regression, 466
    - 9.3.2 Bias and Variance for Classification, 468
  - 9.4 Resampling for Estimating Statistics, 471
    - 9.4.1 Jackknife, 472
      - Example 2** Jackknife Estimate of Bias and Variance of the Mode, 473
    - 9.4.2 Bootstrap, 474
  - 9.5 Resampling for Classifier Design, 475

- 9.5.1 Bagging, 475
- 9.5.2 Boosting, 476
- 9.5.3 Learning with Queries, 480
- 9.5.4 Arcing, Learning with Queries, Bias and Variance, 482
- 9.6 Estimating and Comparing Classifiers, 482
  - 9.6.1 Parametric Models, 483
  - 9.6.2 Cross-Validation, 483
  - 9.6.3 Jackknife and Bootstrap Estimation of Classification Accuracy, 485
  - 9.6.4 Maximum-Likelihood Model Comparison, 486
  - 9.6.5 Bayesian Model Comparison, 487
  - 9.6.6 The Problem-Average Error Rate, 489
  - 9.6.7 Predicting Final Performance from Learning Curves, 492
  - 9.6.8 The Capacity of a Separating Plane, 494
- 9.7 Combining Classifiers, 495
  - 9.7.1 Component Classifiers with Discriminant Functions, 496
  - 9.7.2 Component Classifiers without Discriminant Functions, 498
- Summary, 499
- Bibliographical and Historical Remarks, 500
- Problems, 502
- Computer exercises, 508
- Bibliography, 513

## 10

## UNSUPERVISED LEARNING AND CLUSTERING

517

- 10.1 Introduction, 517
- 10.2 Mixture Densities and Identifiability, 518
- 10.3 Maximum-Likelihood Estimates, 519
- 10.4 Application to Normal Mixtures, 521
  - 10.4.1 Case 1: Unknown Mean Vectors, 522
  - 10.4.2 Case 2: All Parameters Unknown, 524
  - 10.4.3  $k$ -Means Clustering, 526
  - \*10.4.4 Fuzzy  $k$ -Means Clustering, 528
- 10.5 Unsupervised Bayesian Learning, 530
  - 10.5.1 The Bayes Classifier, 530
  - 10.5.2 Learning the Parameter Vector, 531
    - Example 1 Unsupervised Learning of Gaussian Data, 534
  - 10.5.3 Decision-Directed Approximation, 536
- 10.6 Data Description and Clustering, 537
  - 10.6.1 Similarity Measures, 538
- 10.7 Criterion Functions for Clustering, 542
  - 10.7.1 The Sum-of-Squared-Error Criterion, 542
  - 10.7.2 Related Minimum Variance Criteria, 543
  - 10.7.3 Scatter Criteria, 544
    - Example 2 Clustering Criteria, 546
- \*10.8 Iterative Optimization, 548
- 10.9 Hierarchical Clustering, 550
  - 10.9.1 Definitions, 551
  - 10.9.2 Agglomerative Hierarchical Clustering, 552
  - 10.9.3 Stepwise-Optimal Hierarchical Clustering, 555
  - 10.9.4 Hierarchical Clustering and Induced Metrics, 556
- \*10.10 The Problem of Validity, 557



- \*10.11 On-line clustering, 559
  - 10.11.1 Unknown Number of Clusters, 561
  - 10.11.2 Adaptive Resonance, 563
  - 10.11.3 Learning with a Critic, 565
- \*10.12 Graph-Theoretic Methods, 566
- 10.13 Component Analysis, 568
  - 10.13.1 Principal Component Analysis (PCA), 568
  - 10.13.2 Nonlinear Component Analysis (NLCA), 569
  - \*10.13.3 Independent Component Analysis (ICA), 570
- 10.14 Low-Dimensional Representations and Multidimensional Scaling (MDS), 573
  - 10.14.1 Self-Organizing Feature Maps, 576
  - 10.14.2 Clustering and Dimensionality Reduction, 580
- Summary, 581
- Bibliographical and Historical Remarks, 582
- Problems, 583
- Computer exercises, 593
- Bibliography, 598

## A

**MATHEMATICAL FOUNDATIONS****601**

- A.1 Notation, 601
- A.2 Linear Algebra, 604
  - A.2.1 Notation and Preliminaries, 604
  - A.2.2 Inner Product, 605
  - A.2.3 Outer Product, 606
  - A.2.4 Derivatives of Matrices, 606
  - A.2.5 Determinant and Trace, 608
  - A.2.6 Matrix Inversion, 609
  - A.2.7 Eigenvectors and Eigenvalues, 609
- A.3 Lagrange Optimization, 610
- A.4 Probability Theory, 611
  - A.4.1 Discrete Random Variables, 611
  - A.4.2 Expected Values, 611
  - A.4.3 Pairs of Discrete Random Variables, 612
  - A.4.4 Statistical Independence, 613
  - A.4.5 Expected Values of Functions of Two Variables, 613
  - A.4.6 Conditional Probability, 614
  - A.4.7 The Law of Total Probability and Bayes Rule, 615
  - A.4.8 Vector Random Variables, 616
  - A.4.9 Expectations, Mean Vectors and Covariance Matrices, 617
  - A.4.10 Continuous Random Variables, 618
  - A.4.11 Distributions of Sums of Independent Random Variables, 620
  - A.4.12 Normal Distributions, 621
- A.5 Gaussian Derivatives and Integrals, 623
  - A.5.1 Multivariate Normal Densities, 624
  - A.5.2 Bivariate Normal Densities, 626
- A.6 Hypothesis Testing, 628
  - A.6.1 Chi-Squared Test, 629
- A.7 Information Theory, 630
  - A.7.1 Entropy and Information, 630

A.7.2 Relative Entropy, 632  
A.7.3 Mutual Information, 632  
A.8 Computational Complexity, 633  
Bibliography, 635

**INDEX**

# PREFACE

---

Our purpose in writing this second edition—more than a quarter century after the original—remains the same: to give a systematic account of the major topics in pattern recognition, based whenever possible on fundamental principles. We believe that this provides the required foundation for solving problems in more specialized application areas such as speech recognition, optical character recognition, or signal classification. Readers of the first edition often asked why we combined in one book a Part I on pattern classification with a Part II on scene analysis. At the time, we could reply that classification theory was the most important domain-independent theory of pattern recognition, and that scene analysis was the only important application domain. Moreover, in 1973 it was still possible to provide an exposition of the major topics in pattern classification and scene analysis without being superficial. In the intervening years, the explosion of activity in both the theory and practice of pattern recognition has made this view untenable. Knowing that we had to make a choice, we decided to focus our attention on classification theory, leaving the treatment of applications to the books that specialize on particular application domains. Since 1973, there has been an immense wealth of effort, and in many cases progress, on the topics we addressed in the first edition. The pace of progress in algorithms for learning and pattern recognition has been exceeded only by the improvements in computer hardware. Some of the outstanding problems acknowledged in the first edition have been solved, whereas others remain as frustrating as ever. Taken with the manifest usefulness of pattern recognition, this makes the field extremely vigorous and exciting.

While we wrote then that pattern recognition might appear to be a rather specialized topic, it is now abundantly clear that pattern recognition is an immensely broad subject, with applications in fields as diverse as handwriting and gesture recognition, lipreading, geological analysis, document searching, and the recognition of bubble chamber tracks of subatomic particles; it is central to a host of human-machine interface problems, such as pen-based computing. The size of the current volume is a testament to the body of established theory. Whereas we expect that most of our readers will be interested in developing pattern recognition systems, perhaps a few will be active in understanding existing pattern recognition systems, most notably human and animal nervous systems. To address the biological roots of pattern recognition would of course be beyond the scope of this book. Nevertheless, because neurobiologists and psychologists interested in pattern recognition in the natural world continue to rely on more advanced mathematics and theory, they too may profit from the material presented here.

Despite the existence of a number of excellent books that focus on a small set of specific *techniques*, we feel that there is still a strong need for a book such as ours, which takes a somewhat different approach. Rather than focus on a specific technique such as neural networks, we address a specific *class of problems*—pattern recognition problems—and consider the wealth of different techniques that can be applied to it. Students and practitioners typically have a particular problem and need to know which technique is best suited for their needs and goals. In contrast, books that focus on neural networks may not explain decision trees, or nearest-neighbor methods, or many other classifiers to the depth required by the pattern recognition practitioner who must decide among the various alternatives. To avoid this problem, we often discuss the relative strengths and weaknesses of various classification techniques.

These developments demanded a unified presentation in an updated edition of Part I of the original book. We have tried not only to expand but also to improve the text in a number of ways:

**New Material.** The text has been brought up to date with chapters on pattern recognition topics that have, over the past decade or so, proven to be of value: neural networks, stochastic methods, and some topics in the theory of learning, to name a few. While the book continues to stress methods that are statistical at root, for completeness we have included material on syntactic methods as well. “Classical” material has been included, such as Hidden Markov models, model selection, combining classifiers, and so forth.

**Examples.** Throughout the text we have included worked examples, usually containing data and methods simple enough that no tedious calculations are required, yet complex enough to illustrate important points. These are meant to impart intuition, clarify the ideas in the text, and to help students solve the homework problems.

**Algorithms.** Some pattern recognition or learning techniques are best explained with the help of algorithms, and thus we have included several throughout the book. These are meant for clarification, of course; they provide only the skeleton of structure needed for a full computer program. We assume that every reader is familiar with such pseudocode, or can understand it from context here.

**Starred Sections.** The starred sections (\*) are a bit more specialized, and they are typically expansions upon other material. Starred sections are generally not needed to understand subsequent unstarred sections, and thus they can be skipped on first reading.

**Computer Exercises.** These are not specific to any language or system, and thus can be done in the language or style the student finds most comfortable.

**Problems.** New homework problems have been added, organized by the earliest section where the material is covered. In addition, in response to popular demand, a Solutions Manual has been prepared to help instructors who adopt this book for courses.

**Chapter Summaries.** Chapter summaries are included to highlight the most important concepts covered in the rest of the text.

**Graphics.** We have gone to great lengths to produce a large number of high-quality figures and graphics to illustrate our points. Some of these required extensive

calculations, selection, and reselection of parameters to best illustrate the concepts at hand. Study the figures carefully! The book's illustrations are available in Adobe Acrobat format that can be used by faculty adopting this book for courses to create presentations for lectures. The files can be accessed through a standard web browser or an ftp client program at the Wiley STM ftp area at:

[ftp://ftp.wiley.com/public/sci\\_tech\\_med/pattern/](ftp://ftp.wiley.com/public/sci_tech_med/pattern/)

The files can also be accessed from a link on the Wiley Electrical Engineering software supplements page at:

[http://www.wiley.com/products/subject/engineering/electrical/software\\_supplem\\_elec\\_eng.html](http://www.wiley.com/products/subject/engineering/electrical/software_supplem_elec_eng.html)

**Mathematical Appendixes.** It comes as no surprise that students do not have the same mathematical background, and for this reason we have included mathematical appendixes on the foundations needed for the book. We have striven to use clear notation throughout—rich enough to cover the key properties, yet simple enough for easy readability. The list of symbols in the Appendix should help those readers who dip into an isolated section that uses notation developed much earlier.

This book surely contains enough material to fill a two-semester upper-division or graduate course; alternatively, with careful selection of topics, faculty can fashion a one-semester course. A one-semester course could be based on Chapters 1–6, 9 and 10 (most of the material from the first edition, augmented by neural networks and machine learning), with or without the material from the starred sections.

Because of the explosion in research developments, our historical remarks at the end of most chapters are necessarily cursory and somewhat idiosyncratic. Our goal has been to stress important references that help the reader rather than to document the complete historical record and acknowledge, praise, and cite the established researcher. The Bibliography sections contain some valuable references that are not explicitly cited in the body of the text. Readers should also scan through the titles in the Bibliography sections for references of interest.

This book could never have been written without the support and assistance of several institutions. First and foremost is of course Ricoh Innovations (DGS and PEH). Its support of such a long-range and broadly educational project as this book—amidst the rough and tumble world of industry and its never-ending need for products and innovation—is proof positive of a wonderful environment and a rare and enlightened leadership. The enthusiastic support of Morio Onoe, who was Director of Research, Ricoh Company Ltd. when we began our writing efforts, is gratefully acknowledged. Likewise, San Jose State University (ROD), Stanford University (Departments of Electrical Engineering, Statistics and Psychology), The University of California, Berkeley Extension, The International Institute of Advanced Scientific Studies, the Niels Bohr Institute, and the Santa Fe Institute (DGS) all provided a temporary home during the writing of this book. Our sincere gratitude goes to all.

Deep thanks go to Stanford graduate students Regis Van Steenkiste, Chuck Lam and Chris Overton who helped immensely on figure preparation and to Sudeshna Adak, who helped in solving homework problems. Colleagues at Ricoh aided in numerous ways; Kathrin Berkner, Michael Gormish, Maya Gupta, Jonathan Hull

and Greg Wolff deserve special thanks, as does research librarian Rowan Fairgrove, who efficiently found obscure references, including the first names of a few authors. The book has been used in manuscript form in several courses at Stanford University and San Jose State University, and the feedback from students has been invaluable. Numerous faculty and scientific colleagues have sent us many suggestions and caught many errors. The following such commentators warrant special mention: Leo Breiman, David Cooper, Lawrence Fogel, Gary Ford, Isabelle Guyon, Robert Jacobs, Dennis Kibler, Scott Kirkpatrick, Daphne Koller, Benny Lautrup, Nick Littlestone, Amir Najmi, Art Owen, Rosalind Picard, J. Ross Quinlan, Cullen Schaffer, and David Wolpert. Specialist reviewers—Alex Pentland (1), Giovanni Parmigiani (2), Peter Cheeseman (3), Godfried Toussaint (4), Padhraic Smyth (5), Yann Le Cun (6), Emile Aarts (7), Horst Bunke (8), Tom Dietterich (9), Anil Jain (10), and Rao Vemuri (Appendix)—focused on single chapters (as indicated by the numbers in parentheses); their perceptive comments were often enlightening and improved the text in numerous ways. (Nevertheless, we are responsible for any errors that remain.) George Telecki, our editor, gave the needed encouragement and support, and he refrained from complaining as one manuscript deadline after another passed. He, and indeed all the folk at Wiley, were extremely helpful and professional. Finally, deep thanks go to Nancy, Alex, and Olivia Stork for understanding and patience.

DAVID G. STORK  
RICHARD O. DUDA  
PETER E. HART

*Menlo Park, California*  
*August, 2000*

---

# PATTERN CLASSIFICATION

---





---

# INTRODUCTION

The ease with which we recognize a face, understand spoken words, read handwritten characters, identify our car keys in our pocket by feel, and decide whether an apple is ripe by its smell belies the astoundingly complex processes that underlie these acts of pattern recognition. Pattern recognition—the act of taking in raw data and making an action based on the “category” of the pattern—has been crucial for our survival, and over the past tens of millions of years we have evolved highly sophisticated neural and cognitive systems for such tasks.

## 1.1 MACHINE PERCEPTION

---

It is natural that we should seek to design and build machines that can recognize patterns. From automated speech recognition, fingerprint identification, optical character recognition, DNA sequence identification, and much more, it is clear that reliable, accurate pattern recognition by machine would be immensely useful. Moreover, in solving the myriad problems required to build such systems, we gain deeper understanding and appreciation for pattern recognition systems in the natural world—most particularly in humans. For some problems, such as speech and visual recognition, our design efforts may in fact be influenced by knowledge of how these are solved in nature, both in the algorithms we employ and in the design of special-purpose hardware.

## 1.2 AN EXAMPLE

---

To illustrate the complexity of some of the types of problems involved, let us consider the following imaginary and somewhat fanciful example. Suppose that a fish-packing plant wants to automate the process of sorting incoming fish on a conveyor belt according to species. As a pilot project it is decided to try to separate sea bass from salmon using optical sensing. We set up a camera, take some sample images, and begin to note some physical differences between the two types of fish—length, lightness, width, number and shape of fins, position of the mouth, and so on—and these suggest *features* to explore for use in our classifier. We also notice noise or

variations in the images—variations in lighting, position of the fish on the conveyor, even “static” due to the electronics of the camera itself.

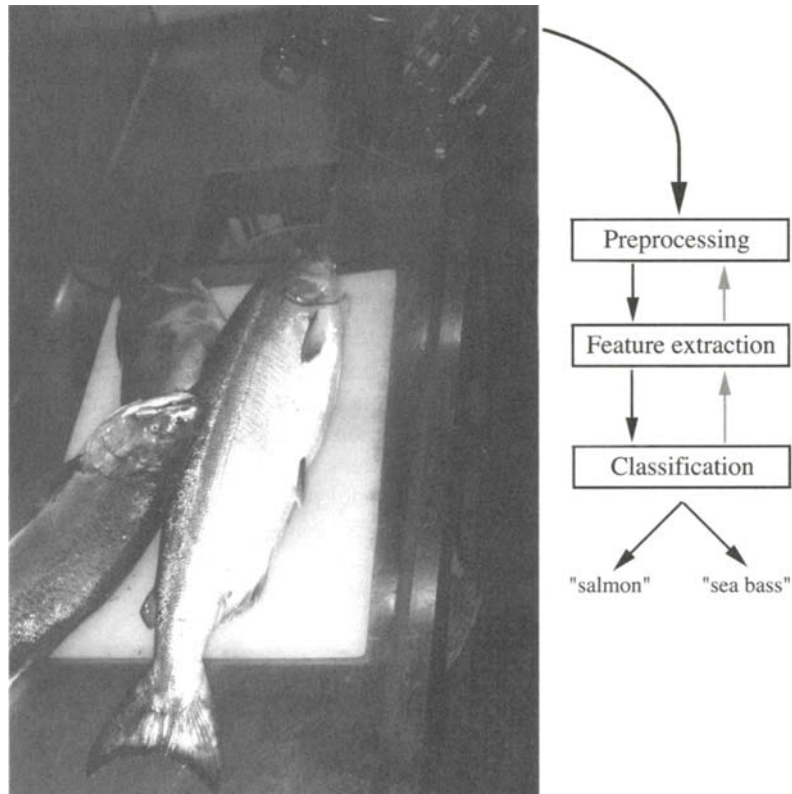
### MODEL

Given that there truly are differences between the population of sea bass and that of salmon, we view them as having different *models*—different descriptions, which are typically mathematical in form. The overarching goal and approach in pattern classification is to hypothesize the class of these models, process the sensed data to eliminate noise (not due to the models), and for any sensed pattern choose the model that corresponds best. Any techniques that further this aim should be in the conceptual toolbox of the designer of pattern recognition systems.

### PREPROCESSING SEGMENTATION

Our prototype system to perform this very specific task might well have the form shown in Fig. 1.1. First the camera captures an image of the fish. Next, the camera’s signals are *preprocessed* to simplify subsequent operations without losing relevant information. In particular, we might use a *segmentation* operation in which the images of different fish are somehow isolated from one another and from the background. The information from a single fish is then sent to a *feature extractor*, whose purpose is to reduce the data by measuring certain “features” or “properties.”

### FEATURE EXTRACTION



**FIGURE 1.1.** The objects to be classified are first sensed by a transducer (camera), whose signals are preprocessed. Next the features are extracted and finally the classification is emitted, here either “salmon” or “sea bass.” Although the information flow is often chosen to be from the source to the classifier, some systems employ information flow in which earlier levels of processing can be altered based on the tentative or preliminary response in later levels (gray arrows). Yet others combine two or more stages into a unified step, such as simultaneous segmentation and feature extraction.

These features (or, more precisely, the values of these features) are then passed to a *classifier* that evaluates the evidence presented and makes a final decision as to the species.

The preprocessor might automatically adjust for average light level, or threshold the image to remove the background of the conveyor belt, and so forth. For the moment let us pass over how the images of the fish might be segmented and consider how the feature extractor and classifier might be designed. Suppose somebody at the fish plant tells us that a sea bass is generally longer than a salmon. These, then, give us our tentative *models* for the fish: Sea bass have some typical length, and this is greater than that for salmon. Then length becomes an obvious feature, and we might attempt to classify the fish merely by seeing whether or not the length  $l$  of a fish exceeds some critical value  $l^*$ . To choose  $l^*$  we could obtain some *design* or *training samples* of the different types of fish, make length measurements, and inspect the results.

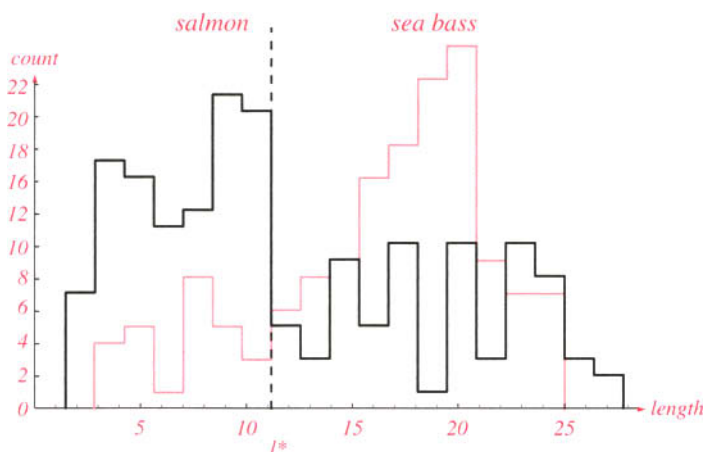
### TRAINING SAMPLES

Suppose that we do this and obtain the histograms shown in Fig. 1.2. These disappointing histograms bear out the statement that sea bass are somewhat longer than salmon, on average, but it is clear that this single criterion is quite poor; no matter how we choose  $l^*$ , we cannot reliably separate sea bass from salmon by length alone.

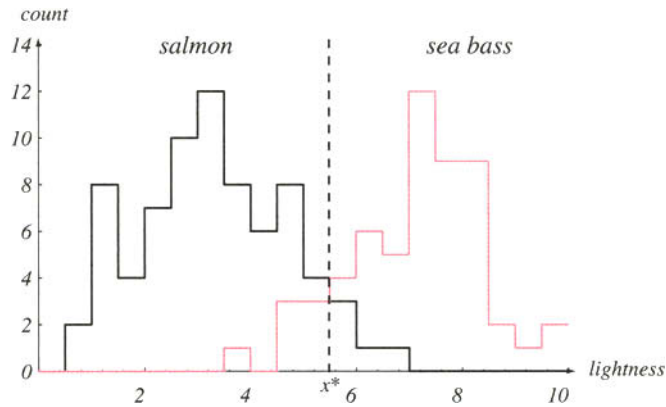
Discouraged, but undeterred by these unpromising results, we try another feature, namely the average lightness of the fish scales. Now we are very careful to eliminate variations in illumination, because they can only obscure the models and corrupt our new classifier. The resulting histograms and critical value  $x^*$ , shown in Fig. 1.3, are much more satisfactory: The classes are much better separated.

So far we have tacitly assumed that the consequences of our actions are equally costly: Deciding the fish was a sea bass when in fact it was a salmon was just as undesirable as the converse. Such a symmetry in the *cost* is often, but not invariably, the case. For instance, as a fish-packing company we may know that our customers easily accept occasional pieces of tasty salmon in their cans labeled “sea bass,” but they object vigorously if a piece of sea bass appears in their cans labeled “salmon.” If we want to stay in business, we should adjust our decisions to avoid antagonizing our customers, even if it means that more salmon makes its way into the cans of

### COST



**FIGURE 1.2.** Histograms for the length feature for the two categories. No single threshold value of the length will serve to unambiguously discriminate between the two categories; using length alone, we will have some errors. The value marked  $l^*$  will lead to the smallest number of errors, on average.



**FIGURE 1.3.** Histograms for the lightness feature for the two categories. No single threshold value  $x^*$  (decision boundary) will serve to unambiguously discriminate between the two categories; using lightness alone, we will have some errors. The value  $x^*$  marked will lead to the smallest number of errors, on average.

sea bass. In this case, then, we should move our decision boundary to smaller values of lightness, thereby reducing the number of sea bass that are classified as salmon (Fig. 1.3). The more our customers object to getting sea bass with their salmon (i.e., the more costly this type of error) the lower we should set the decision threshold  $x^*$  in Fig. 1.3.

## DECISION THEORY

Such considerations suggest that there is an overall single cost associated with our decision, and our true task is to make a decision rule (i.e., set a decision boundary) so as to minimize such a cost. This is the central task of *decision theory* of which pattern classification is perhaps the most important subfield.

Even if we know the costs associated with our decisions and choose the optimal critical value  $x^*$ , we may be dissatisfied with the resulting performance. Our first impulse might be to seek yet a different feature on which to separate the fish. Let us assume, however, that no other single visual feature yields better performance than that based on lightness. To improve recognition, then, we must resort to the use of *more* than one feature at a time.

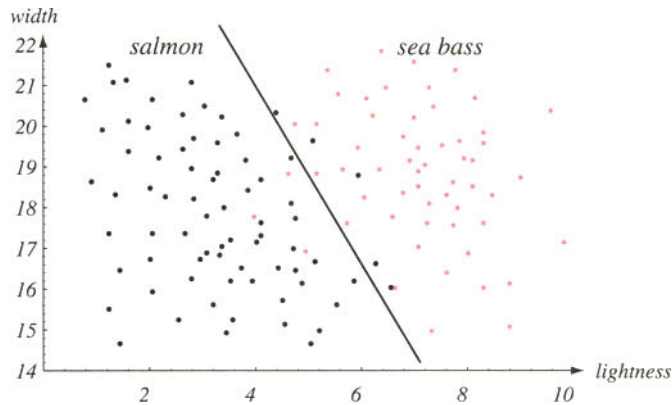
In our search for other features, we might try to capitalize on the observation that sea bass are typically wider than salmon. Now we have two features for classifying fish—the lightness  $x_1$  and the width  $x_2$ . If we ignore how these features might be measured in practice, we realize that the feature extractor has thus reduced the image of each fish to a point or *feature vector*  $\mathbf{x}$  in a two-dimensional *feature space*, where

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

Our problem now is to partition the feature space into two regions, where for all points in one region we will call the fish a sea bass, and for all points in the other we call it a salmon. Suppose that we measure the feature vectors for our samples and obtain the scattering of points shown in Fig. 1.4. This plot suggests the following rule for separating the fish: Classify the fish as sea bass if its feature vector falls above the *decision boundary* shown, and as salmon otherwise.

## DECISION BOUNDARY

This rule appears to do a good job of separating our samples and suggests that perhaps incorporating yet more features would be desirable. Besides the lightness



**FIGURE 1.4.** The two features of lightness and width for sea bass and salmon. The dark line could serve as a decision boundary of our classifier. Overall classification error on the data shown is lower than if we use only one feature as in Fig. 1.3, but there will still be some errors.

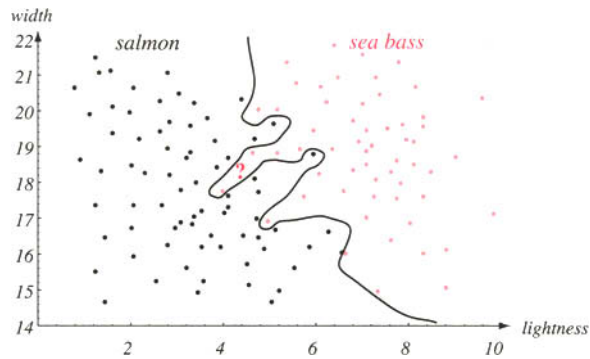
and width of the fish, we might include some shape parameter, such as the vertex angle of the dorsal fin, or the placement of the eyes (as expressed as a proportion of the mouth-to-tail distance), and so on. How do we know beforehand which of these features will work best? Some features might be redundant. For instance, if the eye color of all fish correlated perfectly with width, then classification performance need not be improved if we also include eye color as a feature. Even if the difficulty or computational cost in attaining more features is of no concern, might we ever have *too many* features—is there some “curse” for working in very high dimensions?

Suppose that other features are too expensive to measure, or provide little improvement (or possibly even degrade the performance) in the approach described above, and that we are forced to make our decision based on the two features in Fig. 1.4. If our models were extremely complicated, our classifier would have a decision boundary more complex than the simple straight line. In that case all the training patterns would be separated perfectly, as shown in Fig. 1.5. With such a “solution,” though, our satisfaction would be premature because the central aim of designing a classifier is to suggest actions when presented with *novel* patterns, that is, fish not yet seen. This is the issue of *generalization*. It is unlikely that the complex decision boundary in Fig. 1.5 would provide good generalization—it seems to be “tuned” to the particular training samples, rather than some underlying characteristics or true model of all the sea bass and salmon that will have to be separated.

## GENERALIZATION

Naturally, one approach would be to get more training samples for obtaining a better estimate of the true underlying characteristics, for instance the probability distributions of the categories. In some pattern recognition problems, however, the amount of such data we can obtain easily is often quite limited. Even with a vast amount of training data in a continuous feature space though, if we followed the approach in Fig. 1.5 our classifier would give a horrendously complicated decision boundary—one that would be unlikely to do well on novel patterns.

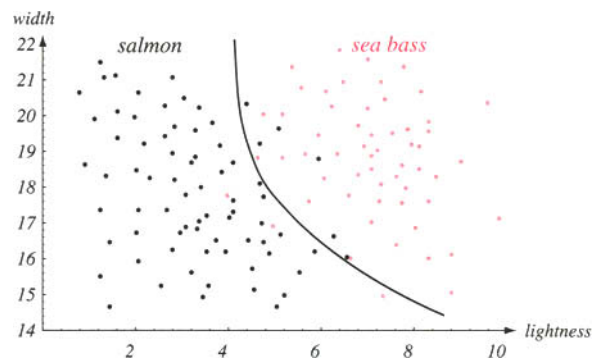
Rather, then, we might seek to “simplify” the recognizer, motivated by a belief that the underlying models will not require a decision boundary that is as complex as that in Fig. 1.5. Indeed, we might be satisfied with the slightly poorer performance on the training samples if it means that our classifier will have better performance



**FIGURE 1.5.** Overly complex models for the fish will lead to decision boundaries that are complicated. While such a decision may lead to perfect classification of our training samples, it would lead to poor performance on future patterns. The novel test point marked ? is evidently most likely a salmon, whereas the complex decision boundary shown leads it to be classified as a sea bass.

on novel patterns.\* But if designing a very complex recognizer is unlikely to give good generalization, precisely how should we quantify and favor simpler classifiers? How would our system automatically determine that the simple curve in Fig. 1.6 is preferable to the manifestly simpler straight line in Fig. 1.4 or the complicated boundary in Fig. 1.5? Assuming that we somehow manage to optimize this tradeoff, can we then *predict* how well our system will generalize to new patterns? These are some of the central problems in *statistical pattern recognition*.

For the same incoming patterns, we might need to use a drastically different task or cost function, and this will lead to different actions altogether. We might, for instance, wish instead to separate the fish based on their sex—all females (of either species) from all males—if we wish to sell roe. Alternatively, we might wish to cull



**FIGURE 1.6.** The decision boundary shown might represent the optimal tradeoff between performance on the training set and simplicity of classifier, thereby giving the highest accuracy on new patterns.

\*The philosophical underpinnings of this approach derive from William of Occam (1284–1347?), who advocated favoring *simpler* explanations over those that are needlessly complicated: *Entia non sunt multiplicanda praeter necessitatem* (“Entities are not to be multiplied without necessity”). Decisions based on overly complex models often lead to lower accuracy of the classifier.