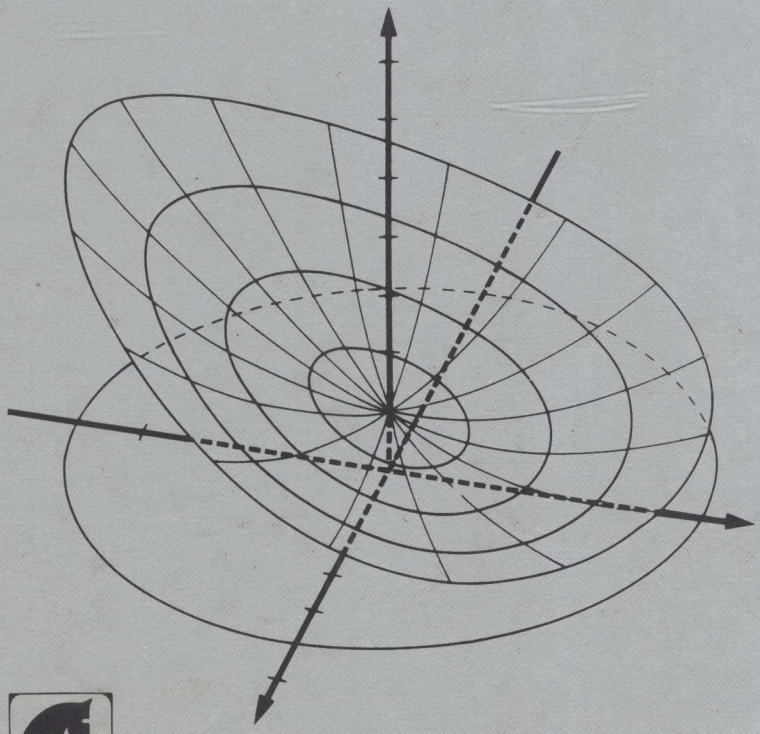
Jack Sklansky
Gustav N. Wassel

Pattern Classifiers and Trainable Machines



Springer-Verlag
New York Heidelberg Berlin

Jack Sklansky
Gustav N. Wassel

Pattern Classifiers and Trainable Machines

With 117 Illustrations

Springer-Verlag
New York Heidelberg Berlin

Jack Sklansky
Department of Electrical Engineering
University of California at Irvine
Irvine, CA 92717
U.S.A.

Gustav N. Wassel
Department of Electronic and
Electrical Engineering
California Polytechnic State
University
San Luis Obispo, CA 93407
U.S.A.

Library of Congress Cataloging in Publication Data
Sklansky, Jack.
Pattern classifiers and trainable machines.

1. Pattern recognition systems. I. Wassel, Gustav N. II. Title.
III. Title: Trainable machines.
TK7882.P3S57 621.3819'598 81-5814
AACR2

Printed in the United States of America.

9 8 7 6 5 4 3 2 1

ISBN 0-387-90435-2 Springer-Verlag New York Heidelberg Berlin
ISBN 3-540-90435-2 Springer-Verlag Berlin Heidelberg New York

Pattern Classifiers and Trainable Machines

To
Gloria and Ruth

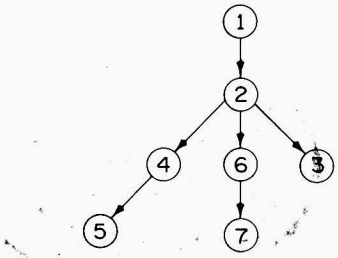
Preface

This book is the outgrowth of both a research program and a graduate course at the University of California, Irvine (UCI) since 1966, as well as a graduate course at the California State Polytechnic University, Pomona (Cal Poly Pomona). The research program, part of the UCI Pattern Recognition Project, was concerned with the design of trainable classifiers; the graduate courses were broader in scope, including subjects such as feature selection, cluster analysis, choice of data set, and estimates of probability densities.

In the interest of minimizing overlap with other books on pattern recognition or classifier theory, we have selected a few topics of special interest for this book, and treated them in some depth. Some of this material has not been previously published. The book is intended for use as a guide to the designer of pattern classifiers, or as a text in a graduate course in an engineering or computer science curriculum. Although this book is directed primarily to engineers and computer scientists, it may also be of interest to psychologists, biologists, medical scientists, and social scientists.

We give special attention in this book to linear trainable classifiers and the extensions of these "linear machines" to nonlinear classifiers. The techniques for designing such classifiers were first developed in the late 1950s and early 1960s. Several inadequacies of these techniques moved us to the discovery of new design concepts—such as window training and close opposed pairs of prototypes—and to the development of the continuous-state model of Markov training processes. These results, as well as several other heretofore inadequately published topics within the scope of pattern classifiers and trainable machines, are included in this book.

A graph of the interchapter dependencies is shown below. In this graph, the circle at each node encloses a chapter number. A branch directed from node i to node j indicates that Chapter j depends on Chapter i.



Thus for a short course the reading may be restricted, for example, to Chapters 1, 2, 4, and 5. Further shortening is possible by deletion of portions of chapters. For example Sections 2.2, 2.3, and 2.4 may be omitted in chapter sequence 1, 2, 4, 5.

Exercises for the student are placed at the end of each chapter.

We thank three successive deans of the UCI School of Engineering, Dr. Robert M. Saunders, Dr. James H. Mulligan, Jr., and Dr. Allen R. Stubberud; as well as Dr. Beamont Davison, Dean of Engineering at Cal Poly Pomona, for their support and encouragement of the research and course development leading to this book.

We acknowledge with thanks the contributions of Dr. Leo Michelotti to the locally trained piecewise linear classifier described in Chapter 3; and we thank him for his criticisms of the manuscript—particularly Chapter 2. We are indebted to Dr. Phil Merryman for his doctoral thesis on continuous-state models, upon which most of Chapter 7 is based. We thank Lee Kilday for discussions leading to our procedure for choosing the minimal-cost operating point. We thank Ramalingam Chellappa for his careful reading and constructive criticisms of the manuscript; in particular for his contributions to our discussion of Bayesian learning and sufficient statistics. We thank Carolyn Kimme-Smith, Walter Blume, and Dr. Gloria Frankl for their contributions to the design and programming of the breast tissue classifier described in Chapter 5.

We are grateful to our students and research colleagues for their criticisms of the text and constructive suggestions.

We are indebted to the University of California at Irvine, the National Science Foundation, the Air Force Office of Scientific Research, the National Institute of General Medical Sciences, and Spectra Research Systems for support of the research leading to the results included in this book. Much of this support was provided under the following grants: NSF Grant No. GK-4226, NSF Science Faculty Fellowship 60196, U.S. Air Force Grant No. 69-1813, and U.S. Public Service Grant GM-17632.

We thank Audrey Bennett, Frances Candelori, Deborah S. Germain, Edna Nemetz, Mary Phillips, Dorothy Shearer, Martha Sue Spence Campbell, and Suzanne Costelloe for typing the manuscript, and Arlene Sanders for drawing the figures.

Contents

CHAPTER 1

Introduction and Overview

Just as the successful invention of the airplane stimulated the study of aerodynamics, the modern digital computer has stimulated the study of intelligence and learning.

A frequently occurring form of intelligent behavior is the sorting or classification of data. The process of distinguishing poisonous objects from food is an example of such behavior. Extreme forms of the classification process are scientists' transformations of observations of nature into "laws" of nature.

Because of the difficulty of many practical classification tasks, it is no surprise that this form of intelligent behavior often depends on a learning process. For example, the accuracy with which a radiologist classifies the image of a lesion in a radiograph as either benign or malignant is highly dependent on an extended period of training. Since the late 1950s the growth of the technology of digital computers has spurred both a technology of pattern classification machines and a mathematical theory of simple forms of human and machine learning.

Pattern classification is an information-transformation process. That is, a classifier transforms a relatively large set of mysterious data into a smaller set of useful data. It is not surprising, therefore, that computing machines, as well as living organisms, exhibit the ability to detect and classify patterns. Examples of such machines that have been constructed and used effectively—including a few commercial successes—are blood cell classifiers, chromosome analyzers, analyzers of aerial photographs, speech analyzers, postal zone readers, fingerprint analyzers, and radiograph analyzers.

In this book we describe several mathematical methods for the design of artificial pattern classifiers, with special attention given to training techniques. We also describe the mathematics of continuous-state and Markov-chain learning models—which are applicable to both artificially constructed classifiers and human decision processes.

In this first chapter we develop a few preliminary concepts and notation, and discuss the utility of machine learning in a broad sense. We start with a few basic definitions.

1.1 Basic Definitions

A *classifier* is a device or a process that sorts data into categories or classes.

A *trainable* classifier is a classifier that can improve its performance in response to information it receives as a function of time. Let $\mathscr{C}$ denote such a classifier and let I denote the received information. $\mathscr{C}$ may be a machine, a biological organism or human being, a biological species, a man–machine system, a business organization, or a nation's economic system. I is often a mathematical function of $\mathscr{C}$'s past performance; sometimes it is just a special sequence of observations and correct classifications; usually it is a combination of all three: a special set of observations, associated correct classifications, and the value of a function of some or all of the past performance of $\mathscr{C}$.

Training is the process by which the parameters of $\mathscr{C}$ are adjusted in response to I. (If $\mathscr{C}$ is a human, these parameters are usually parts of a psychological model of the learning process.) A *training procedure* is an algorithm—often a computer algorithm—that implements the training process.

Learning is the motion of a system's effectiveness (or performance) from one level to another. Learning is positive if the motion is in the direction of increased effectiveness. (Implicit in this definition is the existence of a procedure for computing effectiveness or performance quantitatively. We will take up the subject of these performance measures later.) Learning is often associated with feedback, and provides a means by which humans may control their technologies, machines may adapt automatically to changing environments, and species may survive.

The social utility of learning in intelligent machines is particularly evident in the following ways:

1. Models of learning can lead to the construction of machines that learn and relearn the users' goals, even when these goals change over a period of time. It is only by such a feedback process that the human users can have assurance that these machines will serve the users' purposes, and avoid the sorcerer's apprentice syndrome.
2. Models of learning can lead to the construction of machines that learn to overcome the inadequacies and failures of the machine's own

structures and parts. These self-organizing and self-repairing properties may contribute significantly to the economy of designing and operating a complex intelligent machine.

3. Models of human learning provide a basis for the construction of machines that can share with humans the task of learning. We envision here a learning loop consisting of two learn–teach sequences. In one sequence the machine learns those aspects of a task that it is best suited for, and teaches some of what it has learned to the human user. In the second learn–teach sequence, the human learns a task at which he or she is efficient, and teaches the machine a task based on what he or she has learned.

1.2 Trainable Classifiers and Training Theory

In this section we refine our view of pattern classifiers and define and discuss training theory. Recall that we defined a classifier as a device that sorts data into categories. These data are often structured as vectors in *feature space.* Every point in this space is called a *feature vector.* Each component x_i of the feature vector $\mathbf{x}$ is usually a feature, attribute, or property of an object under analysis. For example, the classifier may be analyzing the chromosomes of a single human cell, and sorting these chromosomes into 23 pairs (a karyotype). The feature vector of each chromosome may have components consisting of, for example, the width of its centromere (its "waist"), the average length of its lobes, the distance of its convex hull's* centroid to its mass centroid, etc. [3].

It is often assumed that the feature vectors of a given class are in some sense nearer to all feature vectors in that class than to all or most of the feature vectors in other classes. This is the *compactness hypothesis* [4].

The feature vectors in a given class occupy a region in feature space which we call a *class region.* It is often assumed that every class region is bounded. Another common assumption is that none of the class regions overlap. (However, in most practical problems, some overlap exists.) When the class regions don't overlap, the classes are said to be *separable,* and to have the property of *separability.* If, for every class region, a hyperplane can be placed so that it separates that region from all other class regions, the classes are said to be *linearly separable.* Much of the early work on the theory of pattern classification was concerned with linearly separable classes. Recently, however, a significant amount of work has been concerned with *nonseparable* (i.e., overlapping) classes and classes which are not linearly separable.

* The convex hull of a plane region $\mathscr{R}$ is the smallest convex region that contains $\mathscr{R}$. One can visualize the convex hull by imagining a rigid board cut in the exact of $\mathscr{R}$ and a stretched elastic band placed around the edge of the board. The band lies on the boundary of the convex hull.

The classifier assigns every feature vector to a particular *decision region* $\mathscr{R}_j$ in feature space by means of a set of *decision hypersurfaces* (Figure 1.1). Each such assignment may or may not correspond to a correct or desirable classification. A *trainable classifier* is a classifier which attempts to make the number of incorrect classifications small or zero by adjusting the set of decision regions $\{\mathscr{R}_j\}$ in response to observations on a sequence of feature vectors, $\{\mathbf{x}(n)\}$. This sequence of observations is said to take place during a *learning* or *training* phase.

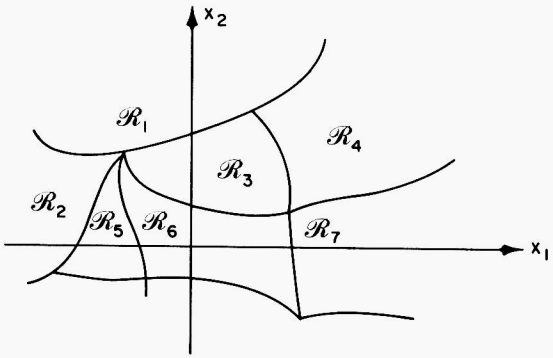


Figure 1.1. An example of a set of decision regions $\{\mathscr{R}_j\}$ in two-dimensional feature space.

Along with the feature vectors, the observations may include information that correctly classifies these feature vectors. If the observations include the correct-classification data, the training is said to be *supervised* or *with a teacher*. These correct-classification data are sometimes referred to as *reinforcements*. If no reinforcements are included in the data, the training is said to be *unsupervised* or *without a teacher*. The process of forming a computer program that classifies various forms of heart enlargements in radiographs in response to radiologists' diagnoses is an example of supervised training [5]. The process by which a botanist learns to classify new forms of plant life into families, genera, species, etc., is an example of unsupervised training.

The procedure or algorithm by which the members of $\{\mathscr{R}_j\}$ are adjusted in response to the observed feature vectors is called a *training procedure*. Each adjustment of $\{\mathscr{R}_j\}$ takes place in response to one or more feature vectors. Each such adjustment, together with the associated observations and reinforcements is called a *trial*, following the terminology of psychologists and psychophysicists. The number of trials is an index of the length of training.

After a classifier is trained, it is usually presented with input data whose classes are unknown. This mode of operation of the classifier is referred to as the *working phase*, while the mode during which training takes place is

known as the *training phase.* The set of feature vectors or observations used as input data during the training phase is referred to as the *training set.* Sometimes the training phase and working phase can coincide or overlap. This is often the case when the training is unsupervised.

Often the decision hypersurfaces of a classifier are determined by a set of *discriminant functions* $\{g_j(\mathbf{x})\}$ as follows:

$$\mathcal{R}_j = \{\mathbf{x} | g_j(\mathbf{x}) \geqslant g_i(\mathbf{x}) \text{ for all } i\}. \tag{1.1}$$

When $g_j(\mathbf{x})$ is of the form

$$g_j(\mathbf{x}) = \mathbf{w}_j^T \mathbf{x} + w_{j0}, \tag{1.2}$$

where $\mathbf{w}_j^T$ denotes the transpose of a *weight vector* $\mathbf{w}_j$, then $g_j(\mathbf{x})$ is a *linear discriminant function* and the classifier is referred to as a *linear classifier.* An advantage of linear over nonlinear classifiers is that the available training procedures for linear classifiers are relatively simple and well understood, and the number of weights w_{ji} to be adjusted during the training phase is relatively small.

We define *training theory* as a body of insights into the relationship between training procedures and learning. The types of insights provided by training theory include relationships among expected performance [6], length of training, stability in the large [7], and the dimensionality of feature space. For example, certain parts of this theory permit us to predict the average performance of trainable classifiers as a function of the length of training [8].

1.3 Assumptions and Notation

It is usually assumed in this book that the data enters the classifier as a sequence of statistically independent d-dimensional random vectors $\{\mathbf{X}(n) | n = 0, 1, \ldots, T-1\}$, each component of $\mathbf{X}(n)$ denoting a feature, property, or measurement of an object under observation. The quantity T denotes the number of trials in the training process. Often we use a lowercase form, namely $\mathbf{x}(n)$ or $\mathbf{x}$, for this vector when the indication of a random process is not important. Unless stated otherwise, it is assumed that the objects under observation change at random as n increases, and that the sequence of classes to which these objects belong is a statistically independent random sequence.

Let $\omega(n)$ denote the class to which $\mathbf{X}(n)$ belongs. In this book the discussion is usually restricted to two classes: ω_1 and ω_2, so that the range of $\omega(n)$ has just two elements. However, two-category classification techniques can often be extended to multiclass cases. One way in which such cases may be handled is by assigning a weight vector to each class, and training each weight vector of the classifier to distinguish between members of the assigned

class and members of all other classes. Another technique, devised by Kesler, constructs a single two-category problem from the multiclass problem. For a description of the Kesler technique see Section 2.6.

1.4 Illustrative Training Process

The fictitious scenario described below illustrates a few of the types of tasks involved in the design of a simple pattern classifier. Real design processes are usually more complex than this one.

Scenario of a Simplified Design Process. A medical instrumentation manufacturer (MIM) wants to construct a pilot model of an artificial classifier that, using microscopic images of blood cells as the only input data, can distinguish malignant cells from benign cells with an accuracy comparable to or better than that of human pathologists. MIM wants the classifier to be capable of being trained by examples of malignant and benign cells, with each example labeled M or B, according as it is malignant or benign, respectively. To obtain these examples MIM enlists the services of a competent pathologist, PA, and a person skilled in the design of automatic pattern classifiers, KL.

Let $\mathscr{M}$ denote the class of all malignant cells, and let $\mathscr{B}$ denote the class of all benign cells. The pathologist analyzes N cells, and labels the ith cell M_i or B_i, denoting membership in $\mathscr{M}$ or $\mathscr{B}$, respectively. (The choice of N is made under the advice of KL.) Let us assume that all of these labels are correct, so that $\{M_i\} \subset \mathscr{M}$ and $\{B_j\} \subset \mathscr{B}$. Let N_M = number of members of $\{M_i\}$, N_B = number of members of $\{B_j\}$. Thus $N_M + N_B = N$. The quantity N_M is chosen equal to the expected number of members of $\mathscr{M}$ in a sample of N cells, and N_B is the expected number of members of $\mathscr{B}$ in a sample of N cells.

PA and KL discuss the problem of assigning physical measurements to the components of a feature space. They decide on the following two-dimensional feature space. (In practice the dimensionality of the feature space is likely to be much higher than two). Let c_n denote a closed plane curve lying on the boundary of the image of the nucleus of the nth cell. Let r_n denote the region enclosed by c_n. Let $X_1(n)$ denote the ratio of the area of r_n to the square of the length of c_n. Let $X_2(n)$ denote the ratio of the area of r_n to the area of the convex hull of r_n. The feature vector $\mathbf{X}(n)$ is

$$\mathbf{X}(n) = [X_1(n), X_2(n)]^T.$$

A *training input sequence* is then formed as follows. Using a table of random numbers or the random number generator of a computer, $\mathscr{M}$ or $\mathscr{B}$ is chosen at random in accordance with the estimated probability of occurrence of $\mathscr{M}$ or $\mathscr{B}$, respectively. For each choice of $\mathscr{M}$ or $\mathscr{B}$ at trial n, an example $\mathbf{X}(n)$ from $\mathscr{M}$ or $\mathscr{B}$, respectively, is selected at random. Let

$\omega(n)$ denote a function of n whose value is the label $\mathscr{M}$ or $\mathscr{B}$ at trial n. For the nth example, the components of $\mathbf{X}(n)$ are measured and recorded. In this way KL obtains the sequence of pairs $\{[\mathbf{X}(n), \omega(n)]\}$ $(n = 0, 1, 2, 3, \ldots, T - 1)$. This is the training input sequence obtained for the artificial classifier.

KL decides on the following form of classifier and training procedure (many others could have been chosen). The classifier is of the form whose output at trial n, namely $R(n)$, is determined by

$$R(n) = \begin{cases} \mathscr{M}, & \text{if } \mathbf{W}(n)^T\mathbf{X}(n) + W_0(n) \geqslant 0, \\ \mathscr{B}, & \text{if } \mathbf{W}(n)^T\mathbf{X}(n) + W_0(n) < 0, \end{cases} \tag{1.3}$$

where $\mathbf{W}(n) = [W_1(n), W_2(n)]^T$ = weight vector of a linear classifier. If we define

$$\mathbf{V}(n) = \begin{bmatrix} W_0(n) \\ W_1(n) \\ W_2(n) \end{bmatrix}$$

and

$$\mathbf{Y}(n) = \begin{bmatrix} 1 \\ X_1(n) \\ X_2(n) \end{bmatrix},$$

then Equation (1.3) may be simplified to

$$R(n) = \begin{cases} \mathscr{M}, & \text{if } \mathbf{V}(n)^T\mathbf{Y}(n) \geqslant 0, \\ \mathscr{B}, & \text{if } \mathbf{V}(n)^T\mathbf{Y}(n) < 0. \end{cases}$$

The chosen training procedure is the form in which the weight vector and $W_0(n)$ are adjusted in accordance with the following rule:

$$\mathbf{V}(n+1) = \begin{cases} \mathbf{V}(n) - \rho\mathbf{Y}(n), & \text{if } R(n) = \mathscr{M}, \omega(n) = \mathscr{B}, \\ \mathbf{V}(n) + \rho\mathbf{Y}(n), & \text{if } R(n) = \mathscr{B}, \omega(n) = \mathscr{M}, \\ \mathbf{V}(n), & \text{if } R(n) = \omega(n), \end{cases} \tag{1.4}$$

where the *step size* ρ is any positive number. This is the *proportional increment* training procedure—one of the most popular training procedures in the early development of the technology of trainable classifiers [9]. The initial values $W_0(0)$, $W_1(0)$, $W_2(0)$ are chosen on the best available evidence for the shapes of the class regions in feature space. Further discussion of this training procedure appears in Section 2.4.

During the training process the augmented weight vector $\mathbf{V}(n)$ moves toward an asymptotic value with a random sequence superimposed on its expected motion. Thus $\mathbf{V}(n)$ does not necessarily achieve a final value with certainty. It may interminably move at random in a neighborhood of the limit of the expected value of $\mathbf{V}(n)$ as $n > \infty$. This phenomenon is discussed in detail in Chapters 4–7.