ESTIMATION AND LEARNING ALGORITHMS

IN

PATTERN RECOGNITION

FEBRUARY 1977

RIICHIRO MIZOGUCHI

ESTIMATION AND LEARNING ALGORITHMS IN PATTERN RECOGNITION

by

RIICHIRO MIZOGUCHI

Submitted in partial fulfillment of
the requirement for the degree of

DOCTOR OF ENGINEERING
(Electrical Engineering)

at

OSAKA UNIVERSITY
TOYONAKA, OSAKA, JAPAN

FEBRUARY 1977

ACKNOWLEDGEMENTS

The author would like to express his grateful acknowledgement to Professor Makoto Kizawa, the thesis supervisor, for his constant guidance and encouragement during the entire course of this work. Special thanks are also expressed to Associate Professor Masamichi Shimura (presently with Tokyo Institute of Technology) for his invaluable and instructive advices.

The author is very much grateful to Professor Kokichi Tanaka for his invaluable guidance.

He also wishes to thank to Associate Professor Jun-ichi Toyoda, Mr. Hiroshi Makino, Mr. Tetsuro Ito, and the colleagues of Prof. Kizawa's laboratory.

ABSTRACT

The present thesis is concerned with estimation and learning techniques in pattern recognition, and consists of 8 chapters including an introductory chapter (Chapter 1) and a concluding chapter (Chapter 8). Chapter 2 presents a supervised learning procedure for constructing a piecewise linear discriminant function which is composed of local minimum number of linear discriminant functions. Chapter 3 deals with the problem of nonsupervised signal detection. It is shown that an adaptive signal detector converging to the optimal machine can be designed without knowing the probability of signal occurrence. Chapter 4 treats the problem of self-learning of a finite mixture. An effective decomposition algorithm of a finite mixture, called WDDM, is presented along with its convergence proof. In the following three chapters, nonsupervised algorithms are considered in terms of nonparametric learning. In Chapter 5, the two-category problem is discussed, while Chapters 6 and 7 deal with the multi-category problem. In Chapter 5, a linear discriminant function is obtained, and it is shown to be effective even when the *a priori* probability of each category is unknown. In chapter 6, an algorithm for estimating one of the modes of a multidimensional probability density function is proposed by using hyper-cubic window function. Chapter 7 treats the problem of cluster detection. An efficient cluster detection algorithm is obtained by introducing hierarchical structure into data set.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 GENERAL BACKGROUND

Pattern recognition is one of the most important problems in the area of artificial intelligence. Since the advent of the digital computer, a constant effort has been made to design pattern recognition machines. Mathematically, pattern recognition is a classification problem, and a number of problems can be formulated as those of pattern classification. In signal detection, for example, the problem is to classify observed waveforms into one of the two classes, containing signal and not; in character reading of uppercase alphabet, the observed characters are classified into one of 26 classes. Although there exist various approaches to the problem of classifier design, the common and major goal is to have a low probability of misclassification.

A pattern recognition machine can be divided into two parts, a feature extractor and a classifier, as shown in Fig. 1.1. There is no general theory of feature extraction because the extraction usually depends on the pattern structure of the particular problem under consideration. On the other hand, the problem of classifier design has the mathematical aspects common to all pattern recognition problems, so that the mathematical theory of classifier design has been developed very extensively.

Suppose that we intend to design a pattern classifier. If the probability distributions of the different categories are known, the

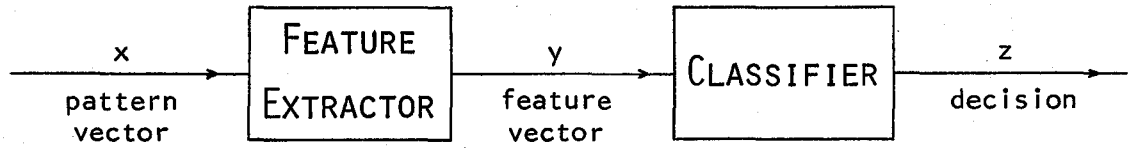


Fig. 1.1 Diagram of a pattern recognition machine.

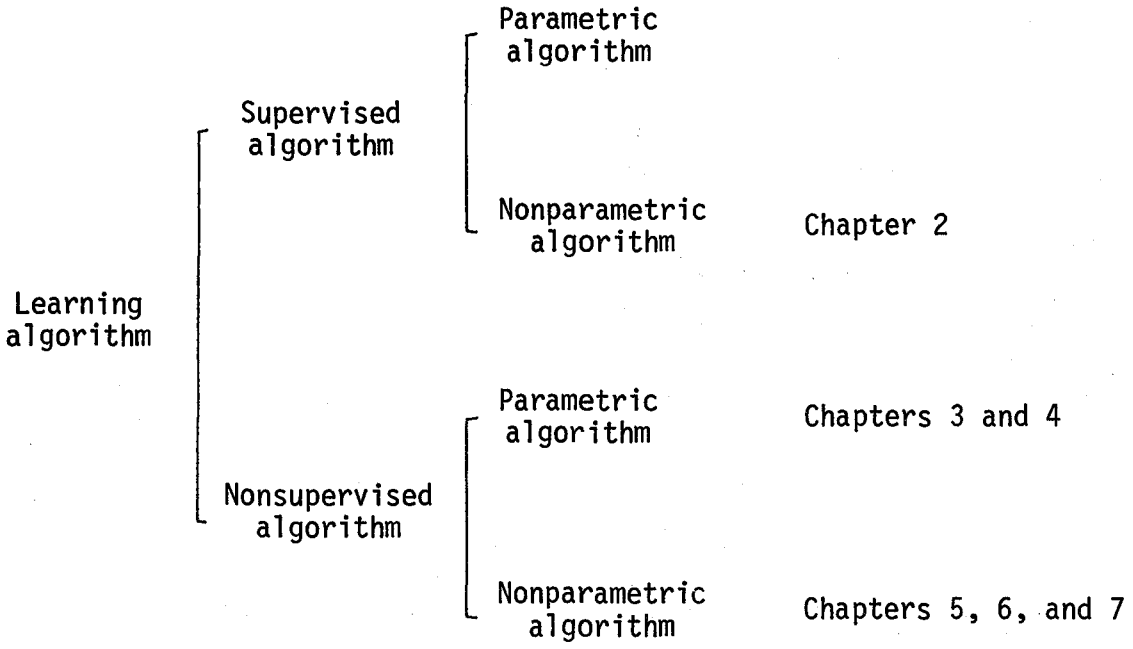


Fig. 1.2 Classification scheme of learning algorithms.

statistical theory of classification may be used to design a classifier. Such a classifier is optimal in terms of probability of misclassification. In practice, however, we rarely have this kind of complete knowledge about the probability structure of the patterns. Usually, we only have some vague, general knowledge about the situation, together with a number of samples — particular representatives of the patterns we want to classify. The problem, then, is to find some way of designing the classifier by using the sample patterns and the *a priori* knowledge. The process of acquisition of necessary knowledge for design from the samples is usually called 'learning'.

In the present thesis, the learning theory of classifier design is studied, and several learning algorithms for designing classifiers are proposed. Learning algorithms are generally divided into four groups according to the information available during the learning period (See Fig. 1.2). In order to clarify the purpose of this thesis we give in the following a brief survey of the learning theory in pattern recognition.

1.2 BRIEF SURVEY OF THE LEARNING THEORY

The problem of supervised parametric learning (Bayesian learning [9],[10],[38]) is completely solved in statistics.

The problem of supervised nonparametric learning was originated by Rosenblatt's perceptron [70]. Since then, there have been proposed many training algorithms for designing linear discriminant functions (LDF's) called error-correction procedures [62]. However, these algorithms do not converge on nonseparable problems. There is another type of algorithms based on stochastic approximation method [95]. The

performance of this type of algorithms is preferable to the error-correction procedures.

The stochastic approximation method was originally proposed by Robbins and Monro [69] in 1951. Since then, a number of contributions to this problem have been made by many authors [5],[17],[18],[24],[39], [40],[88]. The method of stochastic approximation is now an efficient tool in the area of learning theory [13],[74].

Nonsupervised parametric learning began to be studied in the early 1960's. Until 1970, signal detection [11],[12],[65],[75],[76],[83], [97], which is a special case of the two-category problem, had been discussed instead of the multi-category problem. However, since Yakowitz [94] pointed out that the multi-category problem was equivalent to the problem of identification of a finite mixture, nonsupervised parametric learning has been considered as a theory of decomposition of a finite mixture. Although many algorithms have been proposed [32],[36], [68],[79],[96], such an algorithm that can be easily executed independently of the dimensionality of the distributions has not been found yet.

The problem of nonsupervised nonparametric learning is further divided into two groups. One does not store sample patterns, and the other does. The former problem is considerably difficult to solve because of a complete lack of knowledge of the patterns. Therefore, only two-category problem has been studied [6],[81]. The latter problem is well-known as cluster analysis [3],[4],[23],[34],[41],[98]. Although the problem has been considered for many years, satisfactory results have not been known yet.

1.3 ORGANIZATION OF THIS THESIS

This thesis is organized as listed in Table 1.1. In Chapter 2, the problem of designing piecewise linear discriminant functions (PLDF's) is discussed based on a new algorithm for obtaining one of the optimal solutions of linear inequalities. The new algorithm has advantages over the computational time and the storage size. Our PLDF is constructed rather fast and is composed of local minimum number of LDF's, so that the proposed algorithm can be an effective solution to the problem. In Chapter 3, considered is the signal detection problem, which has not been yet solved satisfactorily in spite of its popularity in nonsupervised learning. Two adaptive detectors converging to the optimal machine are obtained without knowing the probability of signal occurrence. In Chapter 4, decomposition of a finite mixture is treated. By extending DDM (decision-directed-machine) a decomposition algorithm of a finite mixture is proposed, which is called WDDM (weighted-decision-directed method). Whereas previous algorithms are rather complex and fail to decompose a multidimensional finite mixture, WDDM has a simple structure and can be easily executed independently of the dimensionality of the distribution under study.

The following three chapters deal with the problem of nonsupervised nonparametric learning. In Chapter 5, a design algorithm of an LDF is discussed by using the first principal component. An efficient threshold value is obtained by estimating the unique minimum point of probability density function, so that our LDF works well even when the *a priori* probability of each category is unknown. In Chapter 6, a mode estimation algorithm of an unknown multidimensional probability

Table 1.1 Organization of the present thesis.

Chapter 2	Supervised nonparametric learning (piecewise linear discriminant function)
Chapter 3	Nonsupervised parametric learning (two-category problem)
Chapter 4	Nonsupervised parametric learning (multi-category problem)
Chapter 5	Nonsupervised nonparametric learning (two-category problem)
Chapter 6	Nonsupervised nonparametric learning (multi-category problem)
Chapter 7	Nonsupervised nonparametric learning (cluster analysis)

density function is proposed by employing a new hyper-cubic window function. This algorithm makes it possible to design a discriminant function for multi-category problem without memorizing patterns. An application of the mode estimation algorithm to nonsupervised nonparametric signal detection is studied and its effectiveness is demonstrated. In Chapter 7, an efficient cluster detection algorithm is presented. In the algorithm, by associating potential with each point, which is an excellent measure of point density, hierarchical structure is introduced into data set. This operation makes it possible to give the algorithm a high ability to detect clusters. Furthermore, it is shown that our algorithm has a flexible structure, that is, it can detect only the specific types of clusters satisfying users' requirements by adjusting parameters appropriately.

All the proposed learning algorithms are verified by computer simulation, and some results are presented at the end of every chapter.

CHAPTER 2

PIECEWISE LINEAR DISCRIMINANT FUNCTIONS

2.1 INTRODUCTION

The purpose of learning classification problems is to construct appropriate discriminant functions by estimating the statistical structure of pattern distribution based on given sample patterns. Since the advent of Rosenblatt's perceptron, a number of researches as to supervised learning have been made. Although excellent results about linear discriminant functions (LDF's) are obtained, many problems are left unsolved concerning nonlinear discriminant functions (NLDF's). It is one of the merits of LDF's that the learning algorithms are very simple and easy to execute. However, their performance is rather poor, since they assume the pattern set to be linearly separable in spite of the fact that most real world patterns are not linearly separable. On the other hand, NLDF's have a great ability to realize any type of decision surfaces, so that the performance is very good. Unfortunately, however, general design algorithms of NLDF's are not established yet.

We know another type of discriminant functions called piecewise linear discriminant function (PLDF). PLDF's are composed of a finite number of LDF's, and they can approximate arbitrary decision surfaces in spite of their simple structure. Therefore, PLDF's can be useful classifiers provided an efficient training algorithm is available. For this reason, we focus our attention on PLDF in this chapter.

In order to improve the performance of perceptron type algorithms in the case where training patterns are not linearly separable, various modifications by memorizing the patterns have been proposed [26], [29],[61],[82],[90],[92]. It is well-known that the problem of constructing an LDF is reduced to that of solving a set of linear inequalities when all the training patterns are memorized. Linear separability and nonlinear separability of the training patterns correspond to the consistency and inconsistency of the linear inequalities, respectively.* Therefore, it is sufficient for constructing a reasonable LDF to obtain such a solution of the linear inequalities that maximizes the number of the satisfied inequalities. In this chapter, we consider the problem of obtaining an optimal solution which locally maximizes the number of the satisfied inequalities.

A number of algorithms for solving linear inequalities have been discussed in the last decade. Ibaraki and Muroga [29] and Warmack and Gonzalez [90] have proposed algorithms for obtaining the optimal solutions. However, their algorithms take a rather long computational time and need the extra-storage requirements. Many results have also been reported as to design algorithms of PLDF's [7],[8],[25],[42]. However, all the proposed algorithms construct rather redundant PLDF's.

In the next section the author proposes an algorithm for obtaining one of the optimal solutions of a set of linear inequalities. It has advantages over the computational time and the storage. In section 2.3,

---

* A set of linear inequalities is said to be consistent when it has a solution and said to be inconsistent when it does not have any solution.

a PLDF is constructed by connecting LDF's in a tree structure, where each LDF is determined by the algorithm mentioned above. If the training pattern set is linearly separable, our PLDF is composed of one LDF. Otherwise, it is composed of local minimum number* of LDF's, since every LDF is determined according to the optimal solution of the linear inequalities. In section 2.4, a learning algorithm for designing a PLDF without memorizing patterns is considered based on an LDF with a window. Finally, some results of computer simulation of our algorithms are shown in section 2.5.

2.2 LINEAR INEQUALITIES

Our algorithm proposed in this section is based on the idea that if the training patterns are linearly separable, then an arbitrary weight vector** can reach the solution region without going out any correct region of pattern which it entered before, otherwise, there exists at least one correct region of pattern such that the weight vector cannot reach without going out a certain correct region of pattern.

Now, let us define a matrix X_N as

$$X_N^T = (x_1, x_2, \cdots, x_M, -x_{M+1}, -x_{M+2}, \cdots, -x_N)$$

where x_i are d-dimensional patterns, and M and N - M are total numbers

---

* This is not global minimum, so that more LDF's than the LDF's of the global minimum number may be needed when bad initial values are used.

** All the discussions in this chapter are made in the weight space.

of patterns of categories 1 and 2, respectively. Then, a set of linear inequalities can be written as

$$X_N W > 0. \tag{2.1}$$

where W denotes the weight vector.

We here consider the following subproblem instead of solving (2.1) directly:

Subproblem 2.1: Given the following k linear inequalities and one of the solutions W_0: $X_k W > 0$. Suppose that a new linear inequality $x_{k+1}^T W > 0$ is added to $X_k W > 0$. If $X_{k+1} W > 0$ has some solutions, then find one of them. Otherwise, let W_0 be the solution of $X_{k+1} W > 0$.

Assume that there exists an algorithm for solving the above subproblem, and we call it Algorithm 2.0. Then, one of the optimal solutions can be found as follows:

Algorithm 2.1:

Step 1: Set n = 0, and choose W_0 arbitrarily.

Step 2: $\chi_t = \{x_i \mid x_i^T W_0 > 0\}$ and $\chi_f = \{x_i \mid x_i^T W_0 \leq 0\}$.

Step 3: If χ_f is empty, then terminate. Otherwise, go to Step 4.

Step 4: Choose a pattern arbitrarily, say x_j, from χ_f.

Step 5: If $x_j^T W_n > 0$, then $\chi_t = \chi_t \cup \{x_j\}$ and go to Step 9.
Otherwise, go to Step 6.

Step 6: n = n + 1.

Step 7: Call Algorithm 2.0 and store the solution in W_n.

Step 8: If $W_n = W_{n-1}$, then go to Step 9. Otherwise,
$\chi_t = \chi_t \cup \{x_j\}$ and go to Step 9.

Step 9: $\chi_f = \chi_f - \{x_j\}$ and go to Step 3.

One can see that the problem of finding one of the optimal solutions of (2.1) is reduced to that of constructing Algorithm 2.0. Note that Subproblem 2.1 can be reformulated as follows:

Subproblem 2.2: Find $W^{\#}$ such that

$$x_{k+1}^T W^{\#} = \max_W [x_{k+1}^T W \mid X_k W \geq 0 \text{ and } x_i^T W = x_i^T W_0]$$

where W_0 is one of the solutions of $X_k W > 0$ and i $(1 \leq i \leq k)$ is arbitrarily fixed. If $x_{k+1}^T W^{\#} > 0$, then $W = W^{\#}$. Otherwise, $W = W_0$.

Of course, Subproblems 2.1 and 2.2 are equivalent to each other, so that we can obtain Algorithm 2.0 by solving Subproblem 2.2 instead of Subproblem 2.1 as follows:

Algorithm 2.0:

Step 1: $m = 1$ and $W^{\#} = W_0$, and store x_i^T in the first row of the matrix X_{min}.

Step 2: $y = (I - X_{min}^T (X_{min} X_{min}^T)^+ X_{min}) x_{k+1}$ *.

Step 3: If $y = 0$, then go to Step 8. Otherwise, go to Step 4.

Step 4: $x_{min} = \arg[\min_{x \in \chi_y} \delta(x)]$ where $\delta(x) = x^T W^{\#} / |x^T y|$ and $\chi_y = \{x \mid x^T y < 0\} \cap \{x_1, x_2, \ldots, x_k\}$.

Step 5: $W^{\#} = W^{\#} + \delta(x_{min}) y$.

Step 6: $m = m + 1$.

Step 7: Store x_{min}^T in the m-th row of the matrix X_{min}, and go to Step 2.

Step 8: $Z^{j-1} = X_{min}^j - (X_{min}^{jT} x_i) x_i / \|x_i\|^2$ where $2 < j < m$ and Z^j

---

* A^+ denotes the Moore-Penrose pseudoinverse of matrix A [2] and I denotes identity matrix.

denotes the j-th row vector of the matrix Z.

Step 9: If $(ZZ^T)^+Z(x_{k+1} - (x_{k+1}^T x_i)x_i/\|x_i\|^2) \leq 0$, then go to Step 12. Otherwise, go to Step 10.

Step 10: Delete the row vector corresponding to the maximal element of $(ZZ^T)^+Z(x_{k+1} - (x_{k+1}^T x_i)x_i/\|x_i\|^2)$ from the matrix X_{min}.

Step 11: $m = m - 1$, and go to Step 2.

Step 12: If $x_{k+1}^T W^{\#} \leq 0$, then $W = W_0$ and terminate. Otherwise, $W = W^{\#} + \alpha(W_0 - W^{\#})/2$ and terminate, where $\alpha = x_{k+1}^T W^{\#}/|x_{k+1}(W_0 - W^{\#})|$.

Before proceeding to the proof of the convergence of Algorithm 2.0, a lemma is presented.

Lemma 2.1: (Farkas) Let A and ζ be an m×n matrix and an m-dimensional vector, respectively, and let ξ and η be n-dimensional vectors. Then, a necessary and sufficient condition of $\xi^T\eta \geq 0$ for any ξ such that $A\xi \geq 0$ is that η can be written as $\eta = A^T\zeta$ where $\zeta \leq 0$.

Proof: See [48].

Now, we have the following theorem:

Theorem 2.1: Algorithm 2.0 can find one of the solutions of the equivalent Subproblems 2.1 and 2.2 in a finite number of steps.

Proof: Algorithm 2.0 is based on gradient projection method [48]. From the theory of generalized inverse matrix [2] it is well-known that $A^+\zeta$ gives the least squares error solution of the minimum norm of $A\xi = \zeta$, and that

$$A^{+} = A^{T}(AA^{T})^{+}$$

$$A = AA^{+}A.$$

One can easily see that $I - A^{+}A$ is the projection matrix on $N(A)^{*}$ from

$$\begin{aligned} & A((I - A^{+}A)\xi) \\ &= A\xi - AA^{+}A\xi \\ &= A\xi - A\xi \\ &= 0. \end{aligned}$$

Therefore, the vector y defined in Step 2 is the orthogonal projection of x_{k+1} on the intersection of all the patterns contained in X_{min}. It is seen from the above discussion that moving W in the direction y can increase $x_{k+1}^{T}W$ without going out the subspace M defined as

$$M = \{W \mid W \in R^{d},\ X_{k}W > 0 \text{ and } x_{i}^{T}W = x_{i}^{T}W_{0}\}.$$

Next, we examine the termination conditions. $W^{\#}$ corresponds to one of the vertexes of the convex set M when y = 0. Therefore, if y = 0, then $W^{\#}$ is tested whether it is a solution or not in Steps 8 and 9, since our solution must be one of the vertexes of M. In order to employ the Farkas' lemma we here show the correspondence between our notations and those used in the lemma:

$$Z \longrightarrow A$$

$$W - W^{\#} \longrightarrow \xi$$

---

* $N(A)$ denotes the null space of the matrix A.

$$x_{k+1} - (x_{k+1}^T x_i)x_i/\|x_i\|^2 \longrightarrow \eta$$

$$(ZZ^T)^+ Z(x_{k+1} - (x_{k+1}^T x_i)x_i/\|x_i\|^2) \longrightarrow \zeta \qquad (2.2)$$

Considering that $y = 0$, one can see from (2.2) that ζ is the solution of

$$Z^T\zeta = \eta \qquad (2.3)$$

where η is the orthogonal projection of x_{k+1} on the hyperplane $x_i^T W = 0$. Therefore, if $\zeta \leq 0$, then from the lemma we have

$$(W - W^{\#})^T \eta \leq 0$$

for an arbitrary vector $W - W^{\#}$ such that

$$Z(W - W^{\#}) \geq 0.$$

In other words, it is impossible to increase $x_{k+1}^T W$ by moving $W^{\#}$ in any interior direction of the convex set M. Hence, the weight vector $W^{\#}$ is a solution when $\zeta \leq 0$.

We next examine the case where $\zeta \nleq 0$, say $\zeta_\ell > 0$, for some ℓ. From (2.3) we have

$$\eta = \sum_{j=1}^{d} \zeta_j Z^j. \qquad (2.4)$$

where Z^j denotes the j-th row vector of the matrix Z. From $\zeta_\ell > 0$ and (2.4), $W^{\#}$ turns out to be not a solution, since $x_{k+1}^T W$ can be increased by moving $W^{\#}$ in the direction where Z^ℓ increase. In this case, therefore, the searching process can be continued by deleting the pattern corresponding to Z^ℓ.

Note that the convex set M is contained in the hyperplane $x_i^T W = x_i^T W_0$. This hyperplane intersects all other hyperplanes, but $x_i^T W = 0$, supporting the convex cone composed of $X_k W \geq 0$, since the hyperplane $x_i^T W = 0$ is one of the supporting hyperplanes of the convex cone. Therefore, it is sufficient for our purpose to seek for a solution on the hyperplane $x_i^T W = x_i^T W_0$. Hence, considering the operation in Step 12, Algorithm 2.0 can find a solution of Subproblems 2.1 and 2.2 with a finite number of steps. (Q.E.D.)

We also have a theorem about Algorithm 2.1.

Theorem 2.2: Algorithm 2.1 finds one of the optimal solutions of the set of linear inequalities (2.1) with a finite number of steps.

Proof: Proof is omitted, since it is obvious from Theorem 2.1 and the procedures of Algorithm 2.1. (Q.E.D.)

2.3 PIECEWISE LINEAR DISCRIMINANT FUNCTION

There are two approaches to the problem of constructing PLDF's. One [8] is based on the adjustment of a set of LDF's whose number and functional form are determined beforehand. The other [7],[25],[47] employs the method of generating LDF's sequentially. In this chapter the latter approach is taken, that is, we construct a PLDF by connecting LDF's in a tree structure, where each LDF is determined one by one according to the optimal solution of the linear inequalities.

First, the training patterns are divided into two groups by an LDF obtained by Algorithm 2.1. Next, each of the two subgroups is again divided into two groups by Algorithm 2.1. This process is continued until all the subgroups of the training patterns consist of patterns of just one category. An example of a PLDF constructed in

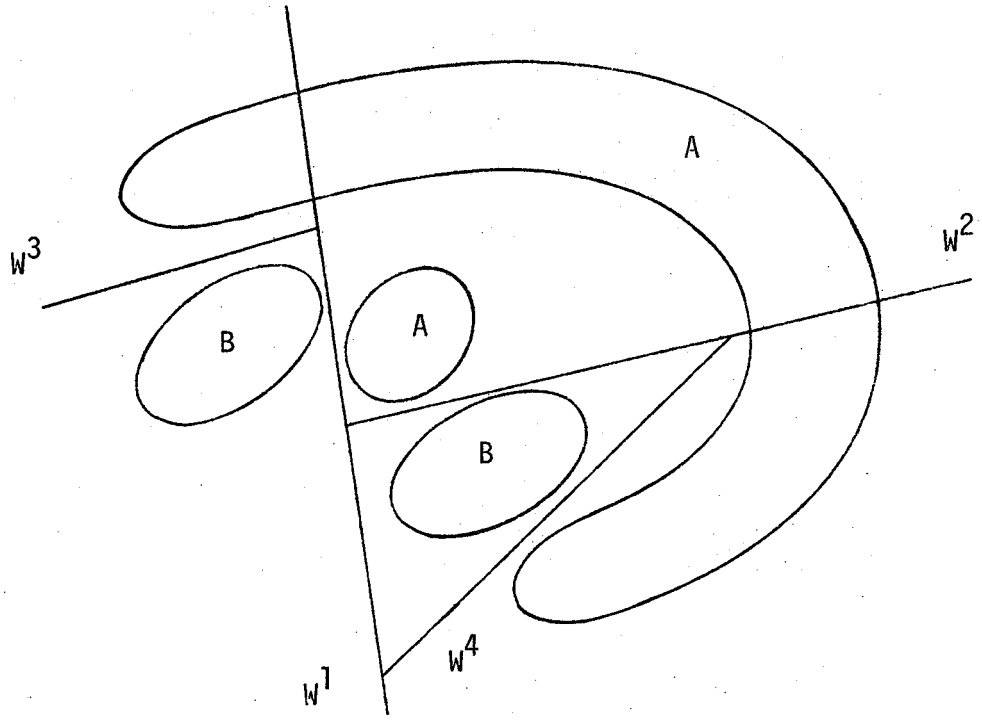
the above manner is shown in Fig. 2.1.

Let us investigate the characteristics of an optimal solution of linear inequalities. When the set of linear inequalities is consistent, that is, when the training patterns are linearly separable, it is obvious that an optimal solution linearly separates the patterns. Note that Algorithm 2.1 is a procedure for finding a weight vector maximizing the number of correctly classified patterns under the constraint that the weight vector must not go out of the correct regions of the patterns which were correctly classified by the weight vector. In Fig. 2.2 (a) and (b), the optimal solutions W^1 and W^2 are obtained by using the initial vectors W_0^1 and W_0^2, respectively. These examples demonstrate that an optimal solution of linear inequalities corresponds to a locally optimal LDF, and this fact shows the effectiveness of Algorithm 2.1 in constructing a PLDF. We here note, however, that the above correspondence does not always hold. For example, in Fig. 2.2 (b) no desirable solution is obtained when W_0^3 is used as an initial vector.

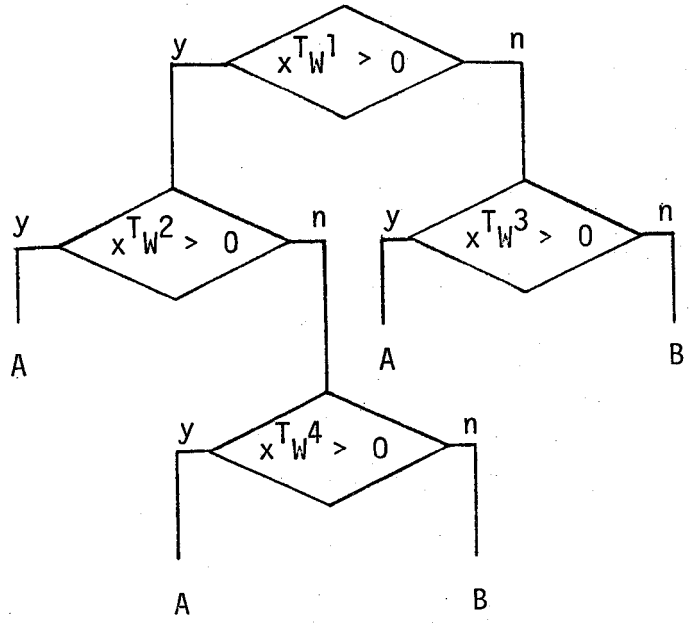
From the above discussion, it is seen that our algorithm can design a PLDF with the global minimum number of LDF's as shown in Fig. 2.3 in the case where appropriate initial weight vectors are available. Even if such weight vectors are not available, a PLDF with a local minimum number of LDF's is obtained, since every LDF is determined so as to locally maximize the number of patterns classified correctly.

2.4 LINEAR DISCRIMINANT FUNCTION WITH A WINDOW

All the algorithms we have discussed thus far are based on the analyses of the stored training patterns. In this section, we consider



(a)



(b)

Fig. 2.1 Piecewise linear discriminant function with a tree structure.

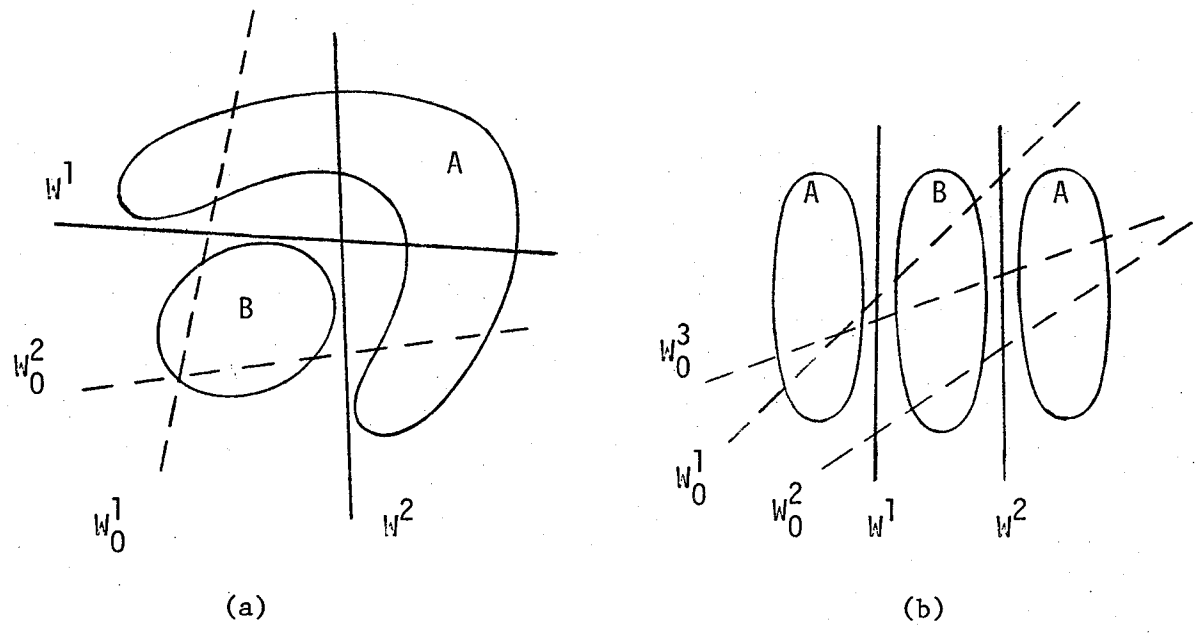


Fig. 2.2 Two-dimensional patterns and optimal solutions.

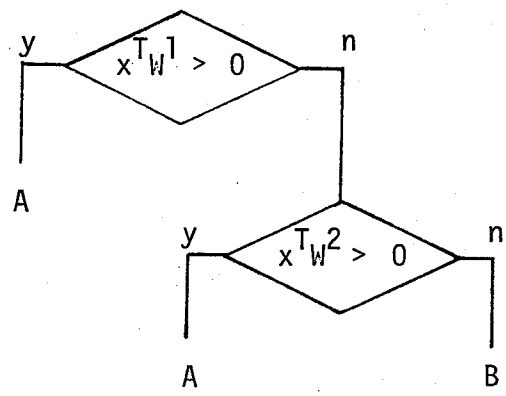


Fig. 2.3 Piecewise linear discriminant function.

the problem of constructing a PLDF without memorizing patterns. In order to do this a new learning algorithm of an LDF with a window (WLDF) is proposed, which corresponds to that for obtaining an optimal solution of a set of linear inequalities discussed in the previous section.

It is well-known that the so-called error-correction procedure cannot find any reasonable solution in the linearly nonseparable case. What we need for design of a PLDF is the locally optimal LDF's such as W^1 and W^2 shown in Fig. 2.2. Therefore, 'local learning' seems to be more useful than 'global learning' like the error-correction learning algorithm. The fact that W^1 and W^2 in Fig. 2.2 correctly discriminate the patterns near them suggest the necessity of 'local learning'.

Now the author proposes the learning algorithm of WLDF.

Algorithm 2.2:

$$W_k' = \begin{cases} W_{k-1} - (1+\alpha)\delta_k d_k x_k & \text{if } d_k < 0 \text{ and } x_k \in C_A \\ & \text{or } d_k > 0 \text{ and } x_k \in C_B \\ W_{k-1} & \text{otherwise} \end{cases}$$

$$W_k = W_k'/\|W_k'\|$$

where

$$d_k = x_k^T W_{k-1}$$

$$\delta_k = \begin{cases} 1 & \text{if } |d_k| \le D_k \text{ and } |(M_{k-1} - x_k)^T W_{k-1}| \le L_k \\ 0 & \text{otherwise} \end{cases}$$

$$D_k = D/\gamma_k^\beta, \qquad L_k = L/\gamma_k^\beta, \qquad 0 < \alpha, \beta < 1$$

$$M_k = M_{k-1} + \delta_k(x_k - M_{k-1})/\gamma_k$$

$$\gamma_k = \gamma_{k-1} + \delta_k$$

C_A and C_B denote the pattern categories A and B, respectively.

WLDF is literally an LDF having a window-like region perpendicular to its hyperplane. The adjustment of the weight vector is made only when a pattern is observed within the window. Let us take the patterns in Fig. 2.4 as examples. The patterns of the category A consist of two clusters, and all the patterns of the right cluster are misclassified by the LDF $x^TW_k = \theta_k$. However, these misclassified patterns are neglected, since they are outside the window. In this case it is clear that the LDF $x^TW_k = \theta_k$ converges to such an LDF that discriminates between the left cluster of the category A and the category B.

As is seen from the above algorithm, the window is made smaller when a pattern is observed within it. This reduction operation enables us to obtain a reasonable LDF even when the training patterns are not linearly separable. The learning process is terminated when training patterns within the window become linearly separable. After obtaining an LDF, its window is removed. Then, a PLDF is constructed in the same way as in Section 2.3. Thus, a PLDF can be obtained without memorizing patterns.

2.5 COMPUTER EXPERIMENTS

Computer simulation of our algorithms was made, and reasonable PLDF's were obtained in every case. It took 12.7 seconds and 2.9 seconds to construct a PLDF composed of 3 LDF's to 72 two-dimensional

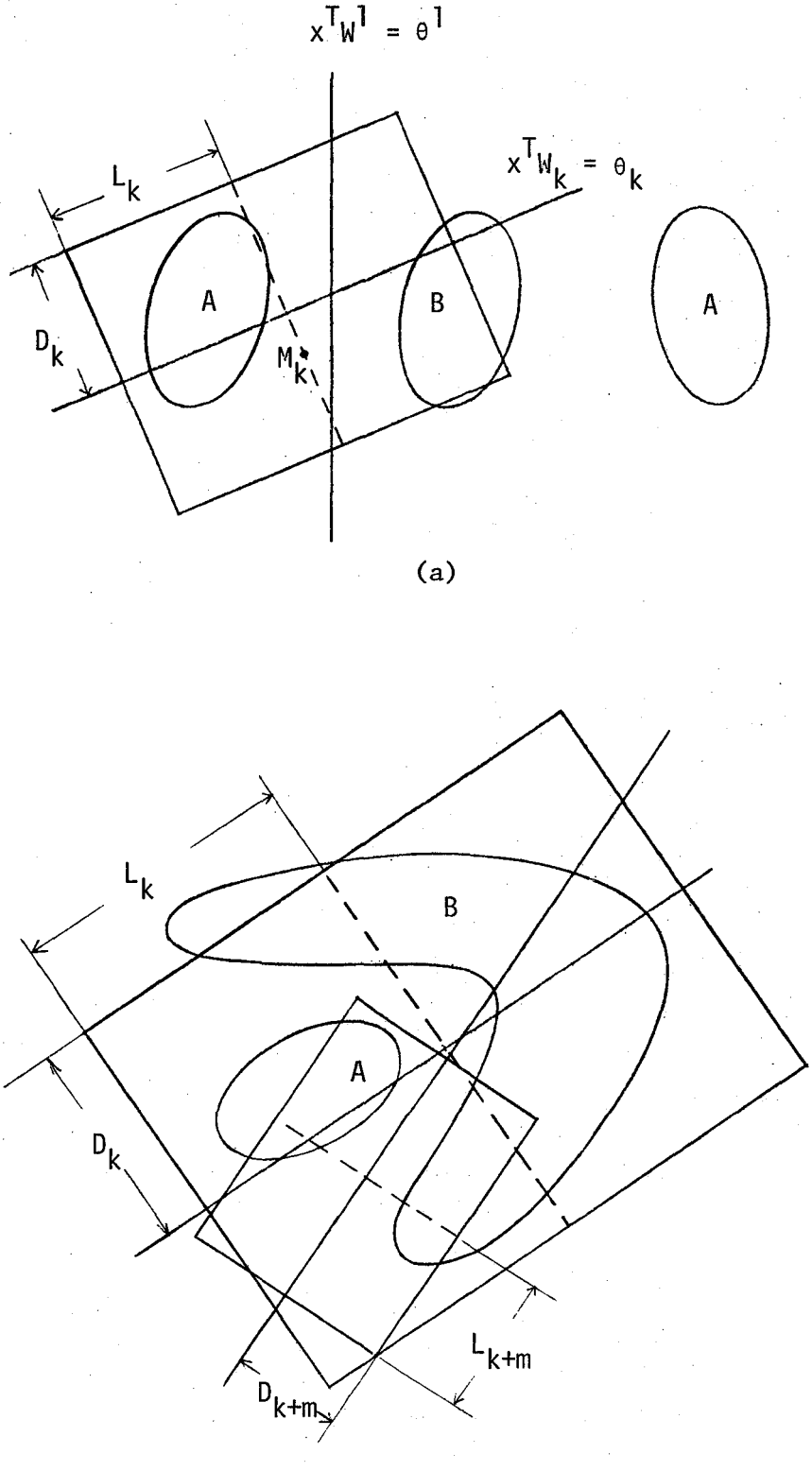


Fig. 2.4 Linear discriminant function with a window.

patterns by Algorithm 2.1 and Algorithm 2.2, respectively. Furthermore, the precise experiments showed that the required learning iterations for Algorithm 2.1 and Algorithm 2.2 were in proportion to N^2 and N, respectively.

2.6 CONCLUSION

In this chapter we have discussed nonparametric algorithms for constructing a PLDF. Our PLDF has been obtained by connecting LDF's in a tree structure, where each LDF is determined by solving linear inequalities. We have proposed an algorithm for finding one of the optimal solutions of a set of linear inequalities using gradient projection method. This algorithm can find it in a finite number of steps in proportion to N^2. Furthermore, a design algorithm of a PLDF without memorizing training patterns has also been proposed by employing a learning algorithm of an LDF with a window. Although the required steps for this algorithm is in proportion to N, much of its performance has been left unknown.

The algorithms proposed in this chapter seem to be rather complicated. This is because the training patterns of interest are not simple, that is, each category is composed of a nonglobular cluster or many clusters. In these cases, our design algorithms of a PLDF come to be useful, since the performance of an LDF is unacceptably poor.

Our PLDF has been constructed with the aid of a teacher. In the succeeding chapters, we shall deal with the nonsupervised problems.

CHAPTER 3

AN APPROACH TO NONSUPERVISED LEARNING CLASSIFICATION

—— TWO-CATEGORY PROBLEM ——

3.1 INTRODUCTION

In nonsupervised learning for two-category problems, the decision-directed type of algorithm has been frequently used, because it can provide a means of nonsupervised adaptation without complexity associated with other nonsupervised learning techniques. Scudder [75],[76] and Tanaka [83] have discussed the learning algorithms for binary detection of unknown signal versus null signal embedded in Gaussian noise, and shown that though asymptotically biased, the estimate of unknown signal converges. Patrick and Costello [65] have extended the scheme for detection of two unknown signals. In their systems, however, the *a priori* probability of signal occurrence is assumed to be known.

Davisson and Schwartz [12] have discussed the behavior of the decision-directed algorithm in the case where the *a priori* probability is unknown. They obtained the estimate of the probability of signal occurrence using its relative frequency. Furthermore, the probability of a runaway (the estimate converges to 1 or 0) is analyzed, and it is shown that if the signal-to-noise ratio is below a critical value, the probability of a runaway is equal to 1.

Young and Coraluppi [96] have discussed a simple self-learning algorithm for decomposing a Gaussian mixture. The algorithm is derived from an information criterion by using stochastic approximation. In

their method, however, one of the local maxima is sought instead of the global maximum of the criterion function. Therefore, there is positive probability of not converging to the correct value.

This chapter discusses a class of nonsupervised pattern classifiers. The classifiers show an asymptotically optimal behavior without knowing the *a priori* probability of the occurrence of each category. The most related work is that of Chien and Fu [10]. They discussed the scheme of obtaining the moments of the mixture distribution using stochastic approximation and applied to the pattern classification problem similar to that treated here. In their method, however, the input patterns are assumed to be one-dimensional, while the classifiers discussed here can be used for multi-dimensional patterns and the classification algorithm can be easily executed. In our method, the mean vector and covariance matrix of the mixture distribution are estimated by using the law of large numbers. The estimate of the probability of each category's occurrence is calculated by using the above estimates of the mean vector and covariance matrix, and is shown to be consistent. Using the consistent estimate of the probability obtained in the above manner, we have the estimates of the mean vector and covariance matrix of each category, which are also proved to be consistent. The discriminant function is then constructed by using the consistent estimates of all unkown statistics of input patterns.

The analytical result of the learning process shows that the classifiers converge probabilistically to the Bayes' minimum error classifier, which is also verified by some computer experiments.

3.2 DESCRIPTION OF THE PROBLEM

Consider the problem of classifying the Gaussian patterns with common covariance matrix into two categories. Let $X_t = \theta_t S + N_t$ be the t-th pattern, where S is the unknown mean vector of category C_1, N_t is the t-th random sample from a Gaussian distribution with mean vector 0 and covariance matrix Σ, and θ_t is a binary variable such that $\theta_t = 1$ indicates the occurrence of C_1 and $\theta_t = 0$ the occurrence of C_0.

According to unknown parameters of input patterns, the following two cases arise.

1) In Case 1, both mean vector S and the probability of category C_1's occurrence P are unknown, but covariance matrix Σ is known.
2) In Case 2, both probability P and covariance matrix Σ are unknown, but the mean vector's power W is known.

If statistics P, S and Σ are given, classification of patterns is performed based on the Bayes' rule as follows:

$$\text{decide:} \quad X \in \begin{cases} C_1, & \text{if } \lambda > 0 \\ C_0, & \text{otherwise} \end{cases} \tag{3.1}$$

where

$$\lambda = (X - S/2)^T \Sigma^{-1} S + \log P/(1 - P). \tag{3.2}$$

3.3 NONSUPERVISED LEARNING ALGORITHMS

Let ν_t be the frequency of occurrence of the category C_1 during the learning period up to time t ($\nu_t \leq t$). Then, from the definition the following relations are obtained:

$$\begin{cases} \dfrac{t}{\nu_t} M_t = \dfrac{1}{\nu_t} \sum_{k=1}^{\nu_t} X_{t_k}^{(1)} + \dfrac{1}{\nu_t} \sum_{k=1}^{t-\nu_t} X_{t_k}^{(0)} \xrightarrow[t \to \infty]{p} S \\ \dfrac{\nu_t}{t} \xrightarrow[t \to \infty]{p} P \end{cases} \tag{3.3}$$

where

$$M_t = \sum_{k=1}^{t} X_k / t,$$

and $X^{(1)}$ and $X^{(0)}$ are the input pattern vectors of the categories C_1 and C_0, respectively.

We here assume that the input pattern X is a sample from the mixture distribution with mean vector S_m and covariance matrix Λ. That is,

$$\begin{cases} S_m = \lim_{t \to \infty} M_t \\ \Lambda = \lim_{t \to \infty} \dfrac{1}{t-1} \sum_{k=1}^{t} (X_k - M_t)(X_k - M_t)^T. \end{cases} \tag{3.4}$$

Note that

$$\mathrm{tr}\ \left(\left(\frac{t}{\nu_t} \right)^2 M_t M_t^T \right) \xrightarrow[t \to \infty]{p} W \tag{3.5}$$

where W is the mean vector's power and tr (A) is the trace of the matrix A.

Now we state the following lemmata.

Lemma 3.1: Define a matrix $U_t(\nu_t)$ by

$$U_t(\nu_t) = \frac{1}{t-1} \sum_{k=1}^{t} (X_k - M_t)(X_k - M_t)^T - \Sigma - \frac{t - \nu_t}{\nu_t} M_t M_t^T. \tag{3.6}$$

Then,

$$U_t(\nu_t) \xrightarrow[t \to \infty]{p} 0 \tag{3.7}$$

where 0 is an n×n zero matrix.

Proof: See Appendix 3.1.

Lemma 3.2: Define a value $V_t(\nu_t)$ by

$$V_t(\nu_t) = W - \mathrm{tr}\,\left(\left(\frac{t}{\nu_t}\right)^2 M_t M_t^T\right). \tag{3.8}$$

Then,

$$V_t(\nu_t) \xrightarrow[t \to \infty]{p} 0. \tag{3.9}$$

Proof: From (3.5) the proof is trivial.

Now let us consider the learning algorithms for obtaining the probability of C_1's occurrence.

Case 1:

Theorem 3.1: Define a value $\bar{\nu}_t$ by

$$\mathrm{tr}\, U_t(\bar{\nu}_t) = \min_{1 \le k \le t} |\mathrm{tr}\, U_t(k)| \tag{3.10}^*$$

and also define a vector $\bar{S}_t$ and a probability $\bar{P}_t$ by

---

* It is important to calculate the estimate of ν_t. However, it takes a comparatively long time to calculate $\bar{\nu}_t$ defined here. In the actual calculation, $\bar{\nu}_t'$ defined in Appendix 3.2 is much easier to obtain and can be used instead of $\bar{\nu}_t$. See Appendix 3.2.

$$\begin{cases} \bar{S}_t = \dfrac{t}{\nu_t} M_t \\ \bar{P}_t = \dfrac{\bar{\nu}_t}{t} . \end{cases} \tag{3.11}$$

Then, we have

$$\begin{cases} \bar{S}_t \xrightarrow[t \to \infty]{p} S \\ \bar{P}_t \xrightarrow[t \to \infty]{p} P. \end{cases} \tag{3.12}$$

Proof: From (3.10)

$$|\text{tr } U_t(\nu_t)| \geq |\text{tr } U_t(\bar{\nu}_t)|. \tag{3.13}$$

From Lemma 3.1

$$|\text{tr } U_t(\nu_t)| \xrightarrow[t \to \infty]{p} 0. \tag{3.14}$$

Then we have

$$\text{tr } U_t(\bar{\nu}_t) = \text{tr } U_t(\nu_t) + \left(1 - \frac{\nu_t}{\bar{\nu}_t}\right) \text{tr } \left(\frac{t}{\nu_t} M_t M_t^T\right) \xrightarrow[t \to \infty]{p} 0. \tag{3.15}$$

Using (3.14) and (3.15), and considering that

$$\text{tr } \left(\frac{t}{\nu_t} M_t M_t^T\right) \geq \text{tr } M_t M_t^T = \left(\frac{\nu_t}{t}\right)^2 \text{tr}\left(\left(\frac{t}{\nu_t}\right)^2 M_t M_t^T\right)$$

$$\xrightarrow[t \to \infty]{p} P^2 W \ (>0),$$

we have

$$\frac{\nu_t}{\bar{\nu}_t} \xrightarrow[t \to \infty]{p} 1. \tag{3.16}$$

Therefore, the following relations are obtained:

$$\begin{cases} \bar{S}_t = \dfrac{M_t}{\nu_t}\dfrac{\nu_t}{\bar{\nu}_t}\, t \xrightarrow[t\to\infty]{p} S \\ \bar{P}_t = \dfrac{\nu_t}{t}\dfrac{\bar{\nu}_t}{\nu_t} \xrightarrow[t\to\infty]{p} P \end{cases}$$

which prove the theorem. (*Q.E.D.*)

Case 2:

Theorem 3.2: Define a value $\tilde{\nu}_t$ by

$$|V_t(\tilde{\nu}_t)| = \min_{1\le k\le t} |V_t(k)| \qquad (3.17)^*$$

and also define a probability $\tilde{P}_t$, vector $\tilde{S}_t$, and matrix $\tilde{\Sigma}_t$ by

$$\begin{cases} \tilde{P}_t = \dfrac{\tilde{\nu}_t}{t}, \qquad \tilde{S}_t = \dfrac{t}{\tilde{\nu}_t}\, M_t \\ \tilde{\Sigma}_t = \dfrac{1}{t-1}\sum_{k=1}^{t} (X_k - M_t)(X_k - M_t)^T - \dfrac{t-\tilde{\nu}_t}{\tilde{\nu}_t} M_t M_t^T. \end{cases} \qquad (3.18)$$

Then the following relations are obtained:

$$\begin{cases} \tilde{P}_t \xrightarrow[t\to\infty]{p} P \\ \tilde{S}_t \xrightarrow[t\to\infty]{p} S \\ \tilde{\Sigma}_t \xrightarrow[t\to\infty]{p} \Sigma. \end{cases}$$

---

* See Appendix 3.2.

Proof: From (3.17)

$$|V_t(\nu_t)| \geq |V_t(\tilde{\nu}_t)|. \tag{3.19}$$

Then, from Lemma 3.2 we have

$$|V_t(\tilde{\nu}_t)| = |V_t(\nu_t) + \left(1 - \left(\frac{\nu_t}{\tilde{\nu}_t}\right)^2\right)\left(\frac{t}{\nu_t}\right)^2 M_t M_t^T| \xrightarrow[t \to \infty]{p} 0 \tag{3.20}$$

and hence

$$\frac{\nu_t}{\tilde{\nu}_t} \xrightarrow[t \to \infty]{p} 1. \tag{3.21}$$

Therefore, we obtain

$$\begin{cases} \tilde{S}_t = \dfrac{t}{\nu_t} M_t \dfrac{\nu_t}{\tilde{\nu}_t} \xrightarrow[t \to \infty]{p} S \\ \dfrac{\tilde{\nu}_t}{t} = \dfrac{\nu_t}{t} \dfrac{\tilde{\nu}_t}{\nu_t} \xrightarrow[t \to \infty]{p} P. \end{cases}$$

By using Lemma 3.1 we have

$$\tilde{\Sigma}_t = \frac{1}{t-1} \sum_{k=1}^{t} (X_k - M_t)(X_k - M_t)^T - \frac{t - \nu_t}{\nu_t} M_t M_t^T$$

$$+ \left(1 - \frac{\nu_t}{\tilde{\nu}_t}\right)\left(\frac{t}{\nu_t}\right) M_t M_t^T \xrightarrow[t \to \infty]{p} \Sigma.$$

These results prove the **theorem**. (*Q.E.D.*)

We have shown the method of obtaining consistent estimates, $\bar{S}_t$ and $\bar{P}_t$ by (3.10) and (3.11) in Case 1, and $\tilde{S}_t$, $\tilde{P}_t$, and $\tilde{\Sigma}_t$ by (3.17) and (3.18) in Case 2. If these estimates are used in the optimal decision rule (3.1) instead of the true statistics S, P, and Σ, we obtain the following decision rule:

$$\text{decide: } X \in \begin{cases} C_1, & \text{if } \lambda_t > 0 \\ C_0, & \text{otherwise} \end{cases} \tag{3.22}$$

where

$$\lambda_t = \begin{cases} (X - \bar{S}_t/2)^T \Sigma^{-1} \bar{S}_t + \log \bar{P}_t/(1 - \bar{P}_t), & \text{for Case 1} \\ (X - \tilde{S}_t/2)^T \tilde{\Sigma}_t^{-1} \tilde{S}_t + \log \tilde{P}_t/(1 - \tilde{P}_t), & \text{for Case 2.} \end{cases}$$

One can easily see that this decision rule converges probabilistically to the Bayes' optimal decision rule after a sufficient large number of iterations, since $\bar{S}_t$(or $\tilde{S}_t$), $\bar{P}_t$(or $\tilde{P}_t$), and $\tilde{\Sigma}_t$ converge probabilistically to the mean vector S, probability of C_1's occurrence P, and covariance matrix Σ, respectively.

3.4 COMPUTER EXPERIMENTS

Some results of a computer study on the classifiers are presented below. In the experiment, 20-dimensional normal patterns are classified into two categories C_1 and C_0, where P = 0.5 and $\Sigma = \sigma^2 I$. The signal-to-noise ratio is defined by $S/N = 10 \log_{10}(S^T S/\sigma^2)$, and the mean vector S is a random sample from a normal distribution with mean vector 0 and covariance matrix $\sigma_s^2 I$.

The learning processes of $\bar{\nu}_t'$ and $\tilde{\nu}_t'$ are shown in Figs. 3.1 and 3.2, respectively. The performance of the classifiers is shown in Fig. 3.3 as the probability of error versus times. From Fig. 3.3 it is seen that the learning procedures in Case 2 takes less time than that in Case 1, though the probability of error in both Cases 1 and 2 converges to the minimum error probability which is indicated by arrows in the figure. This arises from the fact that the term $1/(t-1)$ $\sum_k (X_k-M_t)(X_k-M_t)^T$ in $U_t(\nu_t)$ causes some error in estimating $\bar{\nu}_t'$, especially when the signal-to-noise ratio is comparatively low, while in estimating $\tilde{\nu}_t'$, only the term $tr(t/\nu_t)^2 M_t M_t^T$ is used. However, the method proposed here compare favorably with the decision-directed method as shown in Fig. 3.4.

3.5 CONCLUSION

In this chapter, a new type of nonsupervised adaptive pattern classifiers has been discussed. The main mechanism of the classifiers is based on estimation of the probability of each category's occurrence under the assumption that the input patterns are of a mixture distribution. Utilizing this mechanism, the consistent estimates of unknown statistics of the input patterns were obtained, and then discriminant functions were constructed. It has been shown that the machines with these discriminant functions converge to the Bayes' minimum error classifier. In order to verify their learning processes, some computer experiments have been made and satisfactory results have been obtained.

This chapter has dealt with signal detection problem as a special case of the two-category problem. In the next chapter, we shall discuss the multi-category problem.

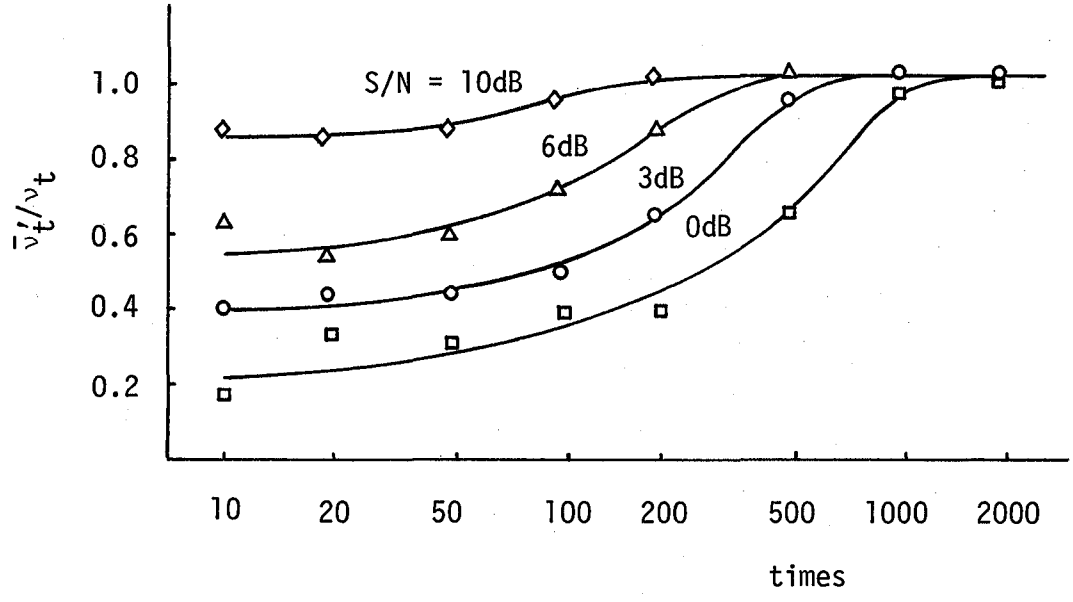


Fig. 3.1 Learning process of $\bar{\nu}_t'$.

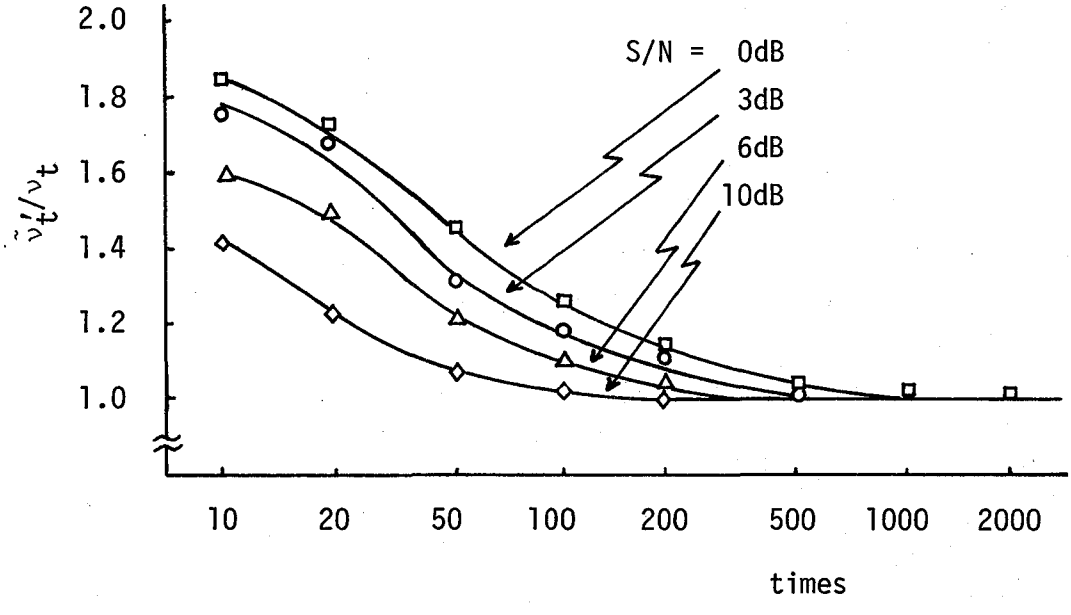


Fig. 3.2 Learning process of $\tilde{\nu}_t'$.

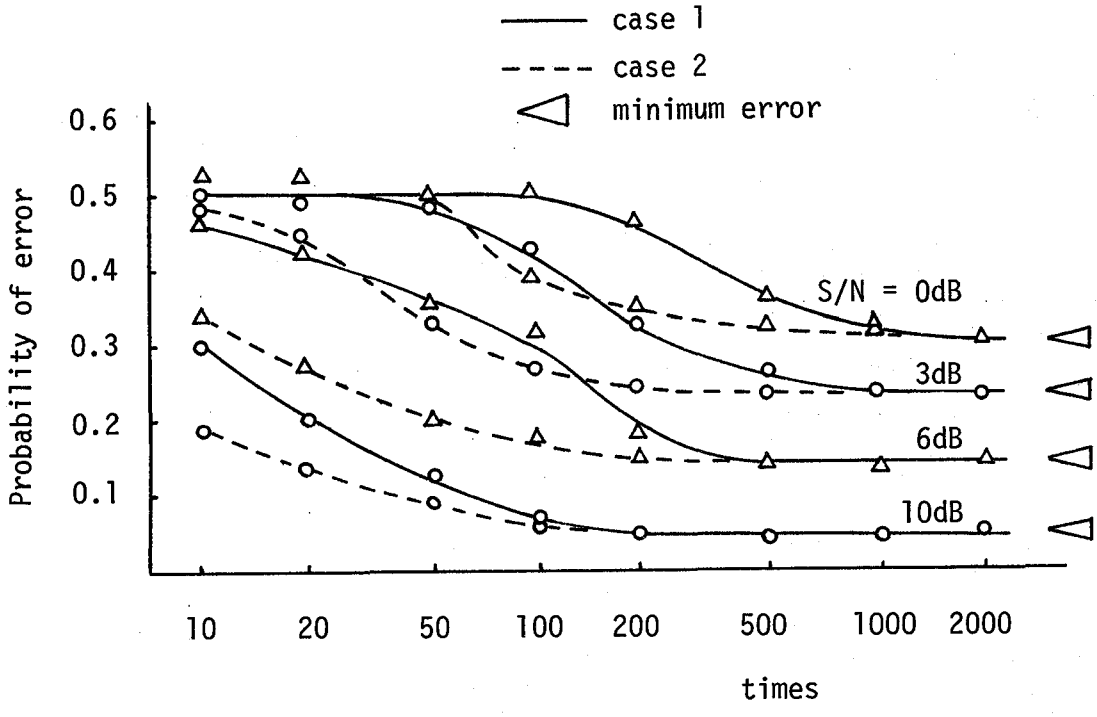


Fig. 3.3 Probability of error versus time in learning process of the machine.

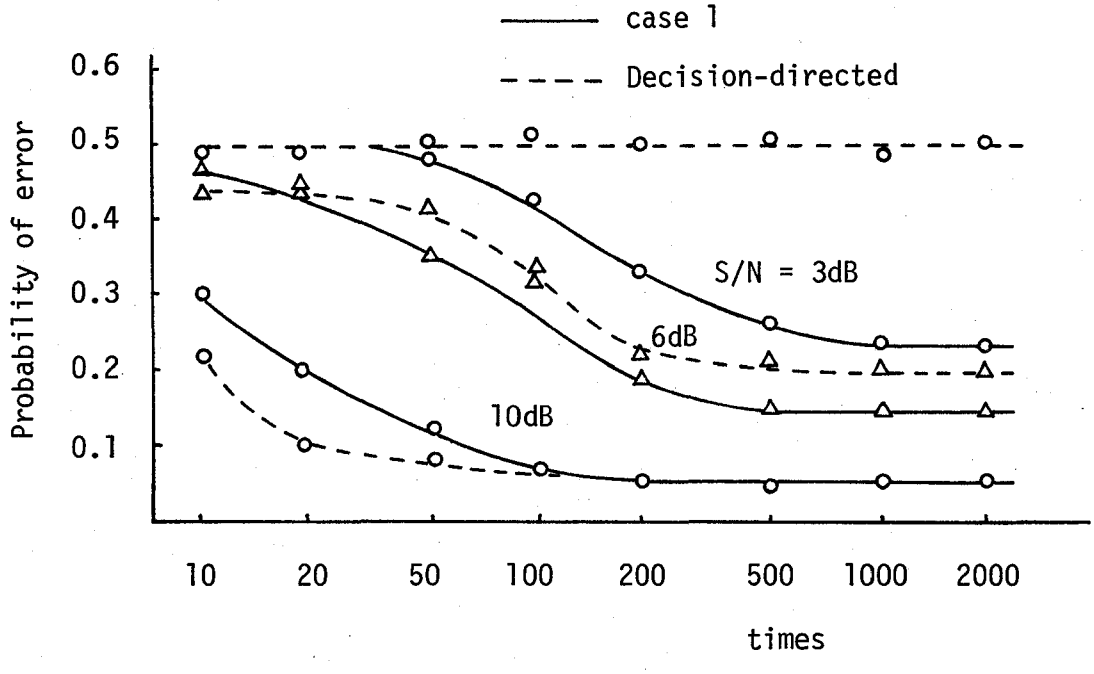


Fig. 3.4 Learning process of the machine for comparison with other methods.

APPENDIX 3.1 PROOF OF *Lemma 3.1*

From the definition of a mixture, we obtain

$$S_m = \frac{P}{|2\pi\Sigma|^{1/2}} \int X \exp[-1/2\ (\ X - S\ )^T\Sigma^{-1}(\ X - S\ )]dX$$

$$+ \frac{1 - P}{|2\pi\Sigma|^{1/2}} \int X \exp[\ -1/2\ X^T\Sigma^{-1}X\ ]dX$$

$$= PS \tag{3.23}$$

$$\Lambda = \frac{P}{|2\pi\Sigma|^{1/2}} \int (\ X - S_m)(\ X - S_m)^T$$

$$\cdot\exp[\ -1/2\ (\ X - S\ )^T\Sigma^{-1}(\ X - S\ )]dX$$

$$+ \frac{1 - P}{|2\pi\Sigma|^{1/2}} \int (\ X - S_m)(\ X - S_m)^T\cdot\exp[\ -1/2\ X^T\Sigma^{-1}X\ ]dX$$

$$= \Sigma + P(\ 1 - P\ )SS^T. \tag{3.24}$$

Substituting (3.3), (3.4), and (3.23) into (3.24), we obtain the proof.

APPENDIX 3.2 ACTUAL CALCULATION OF $\bar{\nu}_t$ AND $\tilde{\nu}_t$

It has been pointed out that tr $U_t(k)$ $(1 \leq k \leq t)$ is a monotone increasing function with respect to k for arbitrarily fixed t, and that $\bar{\nu}_t$, the estimate of ν_t, must satisfy the following relation in order to get the optimal performance:

$$\frac{\bar{\nu}_t}{\nu_t} \xrightarrow[t \to \infty]{p} 1.$$

We now define a positive integer $\bar{\nu}'_t$ in Case 1 by

$$\text{tr } U_t(k) \begin{cases} \geq 0, & \text{if } k \geq \bar{\nu}'_t \\ < 0, & \text{otherwise.} \end{cases}$$

Note that $\bar{\nu}'_t$ is calculated more easily than $\bar{\nu}_t$. We show that $\bar{\nu}'_t$ defined above can be used instead of $\bar{\nu}_t$. From the monotony of tr $U_t(k)$, we have

$$\bar{\nu}_t = \begin{cases} \bar{\nu}'_t - 1, & \text{if } \left|\text{tr } U_t(\bar{\nu}'_t - 1)\right| < \text{tr } U_t(\bar{\nu}'_t) \\ \bar{\nu}'_t, & \text{otherwise} \end{cases}$$

and hence,

$$\frac{\bar{\nu}'_t}{\nu_t} \xrightarrow[t \to \infty]{p} 1.$$

Therefore, the mean vector of the category C_1, $\bar{S}'_t$ estimated and the

probability of C_1's occurrence, $\bar{P}'_t$ calculated by using $\bar{\nu}'_t$ converge probabilistically to S and P, respectively as follows:

$$\begin{cases} \bar{S}'_t = \dfrac{t}{\bar{\nu}'_t} M_t = \dfrac{t}{\nu_t} M_t \dfrac{\nu_t}{\bar{\nu}'_t} \xrightarrow[t \to \infty]{p} S \\ \bar{P}'_t = \dfrac{\bar{\nu}'_t}{t} = \dfrac{\nu_t}{t} \dfrac{\bar{\nu}'_t}{\nu_t} \xrightarrow[t \to \infty]{p} P. \end{cases}$$

This result guarantees that $\bar{\nu}'_t$ can be used instead of $\bar{\nu}_t$ in our algorithms.

We also define a positive integer $\tilde{\nu}'_t$ in Case 2 by

$$V_t(k) \begin{cases} \geq 0, & \text{if } k \geq \tilde{\nu}'_t \\ < 0, & \text{otherwise.} \end{cases}$$

In the same manner as in Case 1, it can be shown that $\tilde{\nu}'_t$ can be used instead of $\tilde{\nu}_t$.

CHAPTER 4

A PARAMETRIC LEARNING METHOD WITHOUT A TEACHER — WDDM —— MULTI-CATEGORY PROBLEM ——

4.1 INTRODUCTION

Self-learning of a finite mixture belongs to nonsupervised parametric learning and is one of the most significant problems in pattern recognition. Recently, some self-learning algorithms are reported, which are divided into two groups. Algorithms of one group are based on the empirical cumulative distribution function (ecdf) constructed beforehand [36],[79], and those of the other group employ stochastic approximation method without using ecdf [32],[36]. The former algorithms need the extra-storage requirements for obtaining ecdf, so that it is rather difficult to execute them in multidimensional cases. On the other hand, the latter algorithms are superior to the former ones in point of economical use of storage. However, they are also difficult to execute in mutidimensional cases, because the integral over variable domain is included in them.

In this chapter a nonsupervised learning algorithm called WDDM (weighted-decision-directed method) is proposed. This algorithm is applicable to all the identifiable distributions having the second moment and is easy to execute even in multidimensional cases. Although WDDM may be considered as an extended version of DDM (decision-directed machine) [75],[76], it differs from DDM in the way of processing input patterns. DDM performs self-learning by using its own decision on the

category which the k-th pattern x_k belongs to. On the other hand, WDDM performs self-learning by not deciding on x_k's category but regarding it as belonging to every category with the weight $p(c^i|\ x_k, \chi_{k-1})$, where c^i, χ_k and $p(c^i|\ x_k, \chi_{k-1})$ denote the category i, the sequence of k patterns, and the conditional probabilities of x_k belonging to c^i given χ_{k-1}, respectively.

The convergence theorem of WDDM is proved by showing the fact that every sequence of learning parameters is a bounded martingale. Some computer simulation of the learning processes of WDDM is made in the problems of decomposition of 4 and 10-dimensional normal mixtures and of detecting a signal embedded in 10-dimensional Gaussian noise. The results of the computer experiments are shown in Section 4.5, where the influence of initial estimates on the performance of WDDM are also considered along with its modification.

4.2 DESCRIPTION OF THE PROBLEM

Let N, P^i, and θ^i be the number of categories, the *a priori* probabilities of the categories c^i, and unknwon parameters of c^i, respectively. Then, a finite mixture F(x) can be written as

$$F(x) = \sum_{i=1}^{N} P^i f(x|\theta^i)$$

where $f(x|\theta^i)$ are density functions of x with parameters θ^i. Now, suppose that unlabeled samples from F(x) are given successively, and functional form of $f(x|\theta^i)$ and $N^{\#}$, the upper bound of the number of the categories are known. Under these circumstances the problem of decomposition of a finite mixture can be stated as follows:

Problem: Estimate all the unknown parameters of F(x) without memorizing any sample pattern.

We here consider an algorithm for estimating all the unknown parameters of a given finite mixture in the case where they are the *a priori* probabilities P^i, the mean vectors μ^i, and the covariance matrices Σ^i of all categories.

Agrawala [1] has proposed a nonsupervised parametric learning algorithm by introducing a probabilistic teacher based on Bayesian learning theory, where the distribution of interest is assumed to have the reproducing property. In Bayesian learning, the estimation of the unknown parameter θ is performed by obtaining *a posteriori* density of θ given the pattern sequence χ_k under the assumption that the unknown parameter θ is with the *a priori* density $p(\theta)$. Therefore, the conditional density of the pattern x_{k+1} given χ_k must be calculated as

$$f(x_{k+1}|\ \chi_k) = \int_\Phi f(x_{k+1}|\ \theta)p(\theta|\chi_k)d\theta$$

where Φ denotes the domain of θ.

On the other hand, there is another way of estimation, stochastic approximation for example, that obtains the k-th estimate of θ, $\bar{\theta}_k$, without assuming the density function of θ. In this type of algorithms, $\bar{\theta}_k$ is determined directly when χ_k is given as $p(\theta|\ \chi_k)$ is determined in Bayesian learning. Therefore, the conditional density of x_{k+1} given χ_k is obtained as

$$f(x_{k+1}|\ \chi_k) = f(x_{k+1}|\bar{\theta}_k).$$

The purpose of this chapter is to consider a self-learning algorithm applicable to the distributions including the ones without the reproducing property. Hence, we employ stochastic approximation type of method instead of Bayesian learning.

4.3 WDDM

Before showing the learning algorithm of WDDM, we present an example of DDM type of algorithm for estimating the *a priori* probabilities P^i, the mean vectors μ^i, and the covariance matrices Σ^i of categories c^i.

$$P^i_{k+1} = P^i_k + \frac{1}{k+1}(D^i_{k+1} - P^i_k) \tag{4.1}$$

$$\mu^i_{k+1} = \mu^i_k + \frac{D^i_{k+1}}{K^i_{k+1}}(x_{k+1} - \mu^i_k) \tag{4.2}$$

$$\Sigma^i_{k+1} = \Sigma^i_k + \frac{D^i_{k+1}}{K^i_{k+1} - 1} \cdot \left(\frac{K^i_{k+1}}{K^i_{k+1} - 1}(x_{k+1} - \mu^i_k)(x_{k+1} - \mu^i_k)^T - \Sigma^i_k \right) \tag{4.3}$$

where the superscripts i indicate the categories $c^i (i = 1, 2, \ldots, N^{\#})$ and

$$D^i_{k+1} = \begin{cases} 1, & \text{if } p(c^i \mid x_{k+1}, \chi_k, \bar{L}_k) \\ & \quad = \max_{1 \le j \le N^{\#}} p(c^j \mid x_{k+1}, \chi_k, \bar{L}_k) \\ 0, & \text{otherwise} \end{cases}$$

$$K^i_{k+1} = \sum_{t=1}^{k+1} D^i_t$$

$$\chi_k = \{x_1, x_2, \ldots, x_k\}$$

$$\bar{L}_k = \{\bar{l}_1, \bar{l}_2, \ldots, \bar{l}_k\}$$

$\bar{l}_t$: x_t's label (category) decided by the machine itself.

It is clear from the law of large numbers that all the parameters defined in (4.1)-(4.3) converge to the correct values with probability one if all the assignments of input patterns to the categories are correct. In practice, however, the above parameters are known not to converge to the correct values because of an infinite number of false assignments.

Now, we present WDDM in the following:

$$P^i_{k+1} = P^i_k + \frac{1}{k+1}\left(p(c^i | x_{k+1}, \chi_k) - P^i_k \right) \tag{4.4}$$

$$\mu^i_{k+1} = \mu^i_k + \frac{p(c^i | x_{k+1}, \chi_k)}{K^i_k}\left(x_{k+1} - \mu^i_k \right) \tag{4.5}$$

$$\Sigma^i_{k+1} = \Sigma^i_k + \frac{p(c^i | x_{k+1}, \chi_k)}{K^i_k} \cdot \left((x_{k+1} - \mu^i_k)(x_{k+1} - \mu^i_k)^T - \Sigma^i_k \right) \tag{4.6}$$

where

$$K_k^i = K_0^i + \sum_{t=1}^{k} p(c^i | x_t, \chi_{t-1})$$

$$\chi_0 = \phi, \quad i = 1, 2, \ldots, N^{\#}.$$

From (4.4)-(4.6) it is seen that WDDM is applicable to all the identifiable distributions with finite mean vectors and covariance matrices, and that it is easily carried out independent of the dimensionality. Furthermore, the above algorithms show that the main difference between WDDM and DDM lies in that learning mechanism of DDM is based on its own decisions on the observed patterns' categories, while WDDM does not make any decisions but regards the k-th pattern x_k as belonging to every category with the weight $p(c^i | x_k, \chi_{k-1})$. The following discussion will reveal the fact that this difference is essential. In the next section, we shall show that the parameters defined in (4.4)-(4.6) are bounded martingales with respect to the pattern sequence χ_k. In Section 4.5, some results of computer simulation of the learning processes of WDDM are presented.

4.4 CONVERGENCE OF THE ALGORITHM

We here present the convergence theorem of WDDM.

Theorem 4.1: The sequences P_k^i, μ_k^i, and Σ_k^i defined in (4.4)-(4.6) are bounded martingales in element-wise with respect to χ_k.

Proof: From (4.4)

$$E[|P_{k+1}^i|] = E[|P_k^i + \frac{1}{k+1}(p(c^i | x_{k+1}, \chi_k) - P_k^i)|]$$

$$= E[|\frac{1}{k+1} \sum_{j=0}^{k} p(c^i | x_{j+1}, \chi_j)|] \leq 1.$$

By similar manipulation, we have

$$E[|(\mu^i_{k+1})_m|] < \infty$$

$$E[|(\Sigma^i_{k+1})_{mn}|] < \infty.$$

where $(\cdot)_m$ and $(\cdot)_{mn}$ denote the m-th entry of the vector and the mn-th entry of the matrix, respectively.

We next show the sequences to be martingales.

$$E[\, P^i_{k+1} |\, \chi_k]$$

$$= E[\, P^i_k + \frac{1}{k+1}\,(\, p(c^i|\, x_{k+1},\, \chi_k) - P^i_k)\,|\, \chi_k]$$

$$= P^i_k + \frac{1}{k+1}\,(\, \int_X \frac{p(x_{k+1},\, c^i|\chi_k)}{p(x_{k+1}|\chi_k)}\, p(x_{k+1}|\chi_k)dx_{k+1} - P^i_k)$$

$$= P^i_k + \frac{1}{k+1}\,(\, p(c^i|\chi_k)\, \int_X p(x_{k+1}|\, c^i,\, \chi_k)dx_{k+1} - P^i_k)$$

$$= P^i_k + \frac{1}{k+1}\,(\, p(c^i|\chi_k) - P^i_k)$$

$$= P^i_k$$

where we used the relation $p(c^i|\chi_k) = P^i_k$ *

---

* See the footnote on the next page.

$$E[\ \mu_{k+1}^{i} | \chi_k]$$

$$= E[\ \mu_k^i + \frac{p(c^i |\ x_{k+1},\ \chi_k)}{K_k^i}\ (\ x_{k+1} - \mu_k^i) | \chi_k\ ]$$

$$= \mu_k^i + \frac{1}{K_k^i} \int_X (\ x_{k+1} - \mu_k^i)\ p(c^i |\ x_{k+1},\ \chi_k) p(x_{k+1} | \chi_k) dx_{k+1}$$

$$= \mu_k^i + \frac{1}{K_k^i} \int_X (\ x_{k+1} - \mu_k^i) \frac{p(x_{k+1},\ c^i | \chi_k)}{p(x_{k+1} | \chi_k)}\ p(x_{k+1} | \chi_k) dx_{k+1}$$

$$= \mu_k^i + \frac{p(c^i | \chi_k)}{K_k^i} \int_X (x_{k+1} - \mu_k^i)\ p(x_{k+1} |\ c^i,\ \chi_k) dx_{k+1}$$

$$= \mu_k^i + \frac{p(c^i | \chi_k)}{K_k^i} \int_X (x_{k+1} - \mu_k^i)\ p(x_{k+1} |\ c_i,\ \mu_k^i,\ \Sigma_k^i)\ dx_{k+1}$$

$$= \mu_k^i$$

---

* $p(c^i | \chi_k)$ denote the conditional probabilities of the occurrence of the categories c^i given χ_k, and $p(c^i |\ x_{k+1},\ \chi_k)$ denote the conditional probabilities of x_{k+1} belonging to the categories c^i given χ_k. In other words, the difference between these two notations is that the former denote *a priori* probabilities, while the latter denote *a posteriori* probabilities with respect to x_{k+1}.

$$E[\ \Sigma_{k+1}^{i} | \chi_k]$$

$$= E[\ \Sigma_k^i + \frac{p(c^i |\ x_{k+1},\ \chi_k)}{K_k^i} ((x_{k+1} - \mu_k^i)(x_{k+1} - \mu_k^i)^T - \Sigma_k^i) |\ \chi_k]$$

$$= \Sigma_k^i + \frac{1}{K_k^i} \int_X ((x_{k+1} - \mu_k^i)(x_{k+1} - \mu_k^i)^T - \Sigma_k^i)$$

$$\cdot \frac{p(x_{k+1},\ c^i | \chi_k)}{p(x_{k+1} | \chi_k)}\ p(x_{k+1} | \chi_k) dx_{k+1}$$

$$= \Sigma_k^i + \frac{p(c^i | \chi_k)}{K_k^i} \int_X ((x_{k+1} - \mu_k^i)(x_{k+1} - \mu_k^i)^T - \Sigma_k^i)$$

$$\cdot p(x_{k+1} |\ c^i,\ \mu_k^i,\ \Sigma_k^i)\ dx_{k+1}$$

$$= \Sigma_k^i \qquad (Q.E.D.)$$

From the above theorem and convergence of bounded martingales, P_k^i, μ_k^i, and Σ_k^i converge with probability one. However, whether or not their limiting values agree with those of the unknown parameters still remains unknown.

4.5 COMPUTER SIMULATION

In order to verify the effectiveness of WDDM we made computer simulation of its learning processes. Some results are presented below.

4.5.1 DECOMPOSITION OF A MULTIDIMENSIONAL NORMAL MIXTURE

In computer study, a 3-category normal mixture

$$F(x) = \sum_{i=1}^{3} P^i \cdot N(x|\mu^i, \Sigma^i)$$

was used. Fig. 4.1 (a) depicts the learning processes of WDDM in decomposing a 10-dimensional mixture with the following parameters:

$$P^1 = 0.2 \qquad P^2 = 0.3 \qquad P^3 = 0.5$$

$$\mu^1 = (2,2,\ldots,2)^T \qquad \mu^2 = (0,0,\ldots,0)^T$$

$$\mu^3 = (-2,-2,\ldots,-2)^T \qquad \Sigma^i = I \ (i=1,2,3).$$

Initial values of the estimators are as follows:

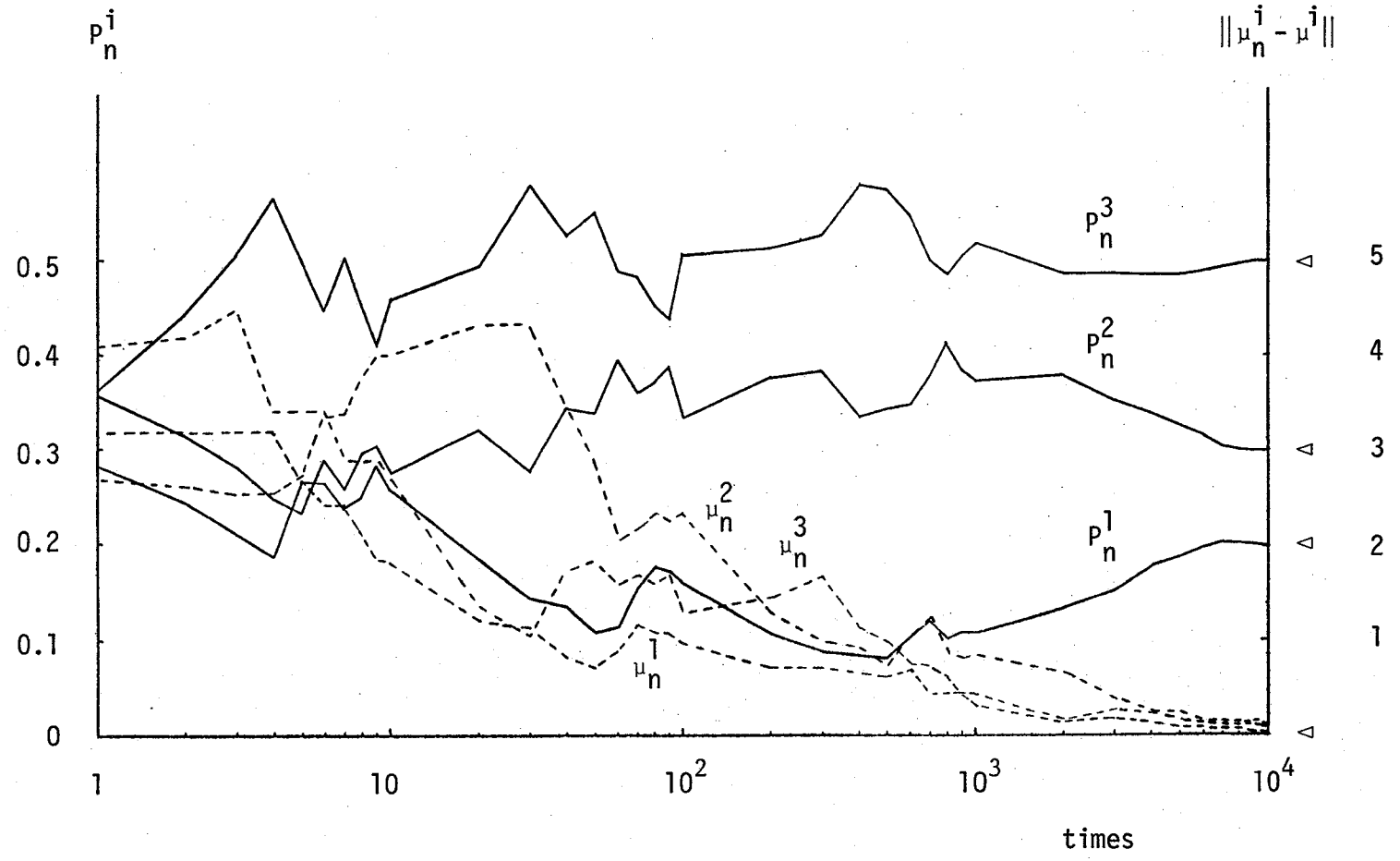
$$P_0^i = 1/3, \quad \Sigma_0^i = 3I \quad (i=1,2,3)$$

$$\mu_0^1 = (3,3,\ldots,3)^T \qquad \mu_0^2 = (1,1,\ldots,1)^T$$

$$\mu_0^3 = (-1,-1,\ldots,-1)^T.$$

An example of the estimated value of the covariance matrix is

$$\Sigma_{10000}^2 = \begin{bmatrix} 1.04 & 0.02 & 0.03 & 0.03 & 0.01 & 0.06 & -0.00 & -0.05 & 0.04 & 0.02 \\ & 0.97 & 0.03 & 0.01 & 0.01 & 0.02 & 0.07 & -0.01 & -0.01 & -0.02 \\ & & 0.93 & 0.01 & -0.00 & 0.07 & 0.03 & 0.02 & -0.03 & -0.05 \\ & & & 1.01 & 0.01 & -0.02 & 0.01 & -0.01 & 0.00 & 0.04 \\ & & & & 0.97 & 0.00 & -0.03 & 0.01 & -0.01 & 0.02 \\ & & & & & 0.97 & 0.01 & -0.02 & -0.01 & 0.01 \\ & & & & & & 1.01 & -0.01 & 0.02 & 0.01 \\ & & * & & & & & 0.99 & 0.03 & 0.04 \\ & & & & & & & & 1.01 & 0.01 \\ & & & & & & & & & 0.98 \end{bmatrix}$$



(a) $N^{\#} = N = 3$ (10-dimensional patterns).

Fig. 4.1 Decomposition of a finite mixture (learning processes of P_n^i and μ_n^i).

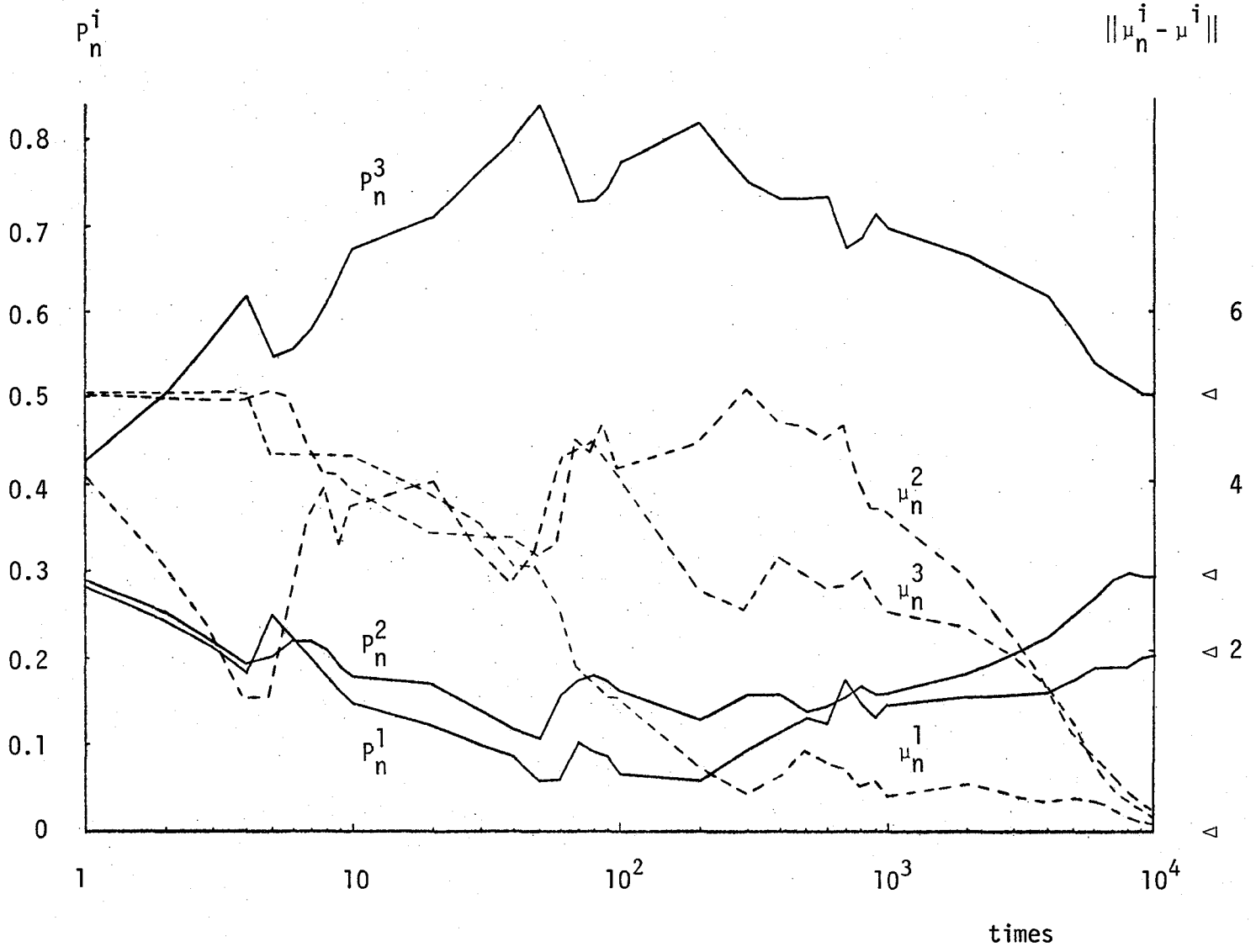


Fig. 4.1 (continued) (b) $N^{\#} = N = 3$ (10-dimensional patterns).

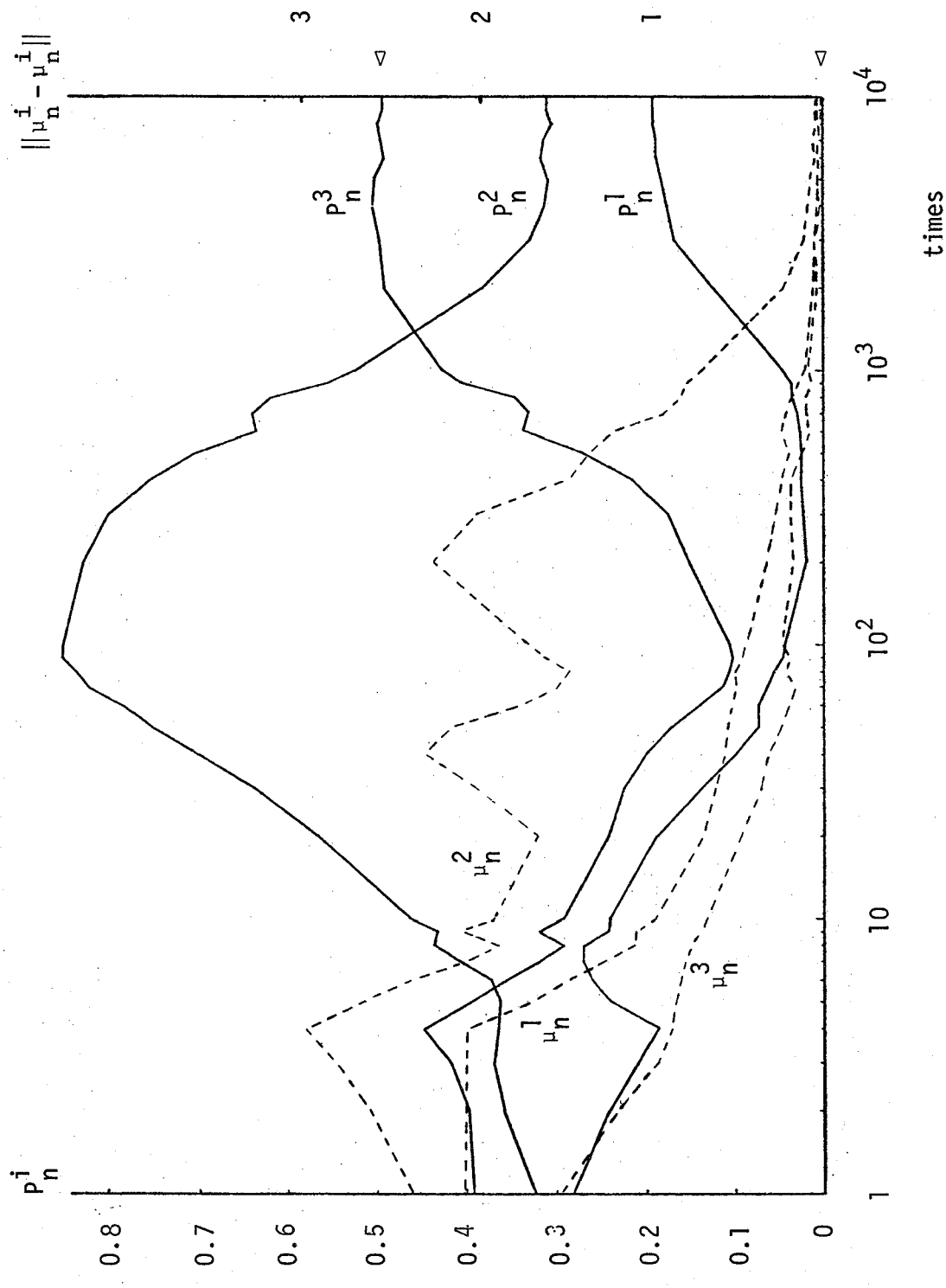


Fig. 4.1 (continued) (c) $N^{\#} = 3$, $N = 2$ (4-dimensional patterns).

Fig. 4.1 (b) depicts another result of the simulation of the same problem as (a) using different random samples. These results show that decomposition of 3-category mixture is performed successfully. Moreover, it is seen from both figures that the estimation of mean vectors proceeds in the desirable direction even when the estimate of *a priori* probability is decreasing. This is one of the remarkable features of WDDM. Fig. 4.1 (c) depicts the result under the following conditions:

$$P^1 = P^2 = 0.5 \qquad P^3 = 0$$

$$\mu^1 = (2,2,2,2)^T \qquad \mu^2 = (0,0,0,0)^T$$

$$\Sigma^1 = \Sigma^2 = 0.25I$$

where 4-dimensional distribution was used. Initial estimates are

$$P_0^1 = P_0^2 = P_0^3 = 1/3 \qquad \mu_0^1 = (3,3,3,3)^T$$

$$\mu_0^2 = (1,1,1,1)^T \qquad \mu_0^3 = (-1,-1,-1,-1)^T$$

$$\Sigma_0^i = 3I \quad (i=1,2,3).$$

The estimates of the covariance matrices at k = 10000 are

$$\Sigma_{10000}^1 = \begin{bmatrix} 0.24 & -0.03 & -0.03 & -0.02 \\ & 0.24 & -0.03 & -0.04 \\ & * & 0.22 & -0.03 \\ & & & 0.24 \end{bmatrix} \quad \Sigma_{10000}^1 = \begin{bmatrix} 0.25 & 0.03 & -0.01 & 0.02 \\ & 0.24 & 0.01 & 0.03 \\ & * & 0.23 & 0.02 \\ & & & 0.25 \end{bmatrix}$$

$$\Sigma^3_{10000} = \begin{bmatrix} 0.24 & 0.00 & -0.00 & -0.00 \\ & 0.24 & -0.01 & -0.00 \\ & * & 0.24 & 0.01 \\ & & & 0.24 \end{bmatrix}$$

At the first glance this result appears to be wrong, that is, the given 2-category mixture seems to be decomposed into 3 categories. However, the detailed results of $P^1_{10000} + P^2_{10000} = 0.504$, $\mu^1_{10000} \doteqdot \mu^2_{10000}$, and $\Sigma^1_{10000} \doteqdot \Sigma^2_{10000}$ show the success on the decomposition of the 2-category mixture.

4.5.2 SIGNAL DETECTION

We also made computer simulation of the learning processes of WDDM in nonsupervised detection of a signal embedded in 10-dimensional Gaussian noise where the signal vector, the probability of signal occurrence, and the covariance matrix of noise were unknown. The results are shown in Fig. 4.2. The probabilities of error of a signal detector constructed by WDDM are depicted against learning iterations. From Fig. 4.2 it is seen that the probabilities of error of the signal detector converge to those of the optimal machine in all SN-ratios 13, 9, 6, and 3dB. Parameters used in the simulation are as follows:

$$P = 0.3 \qquad \mu = (2,2,\ldots,2)^T \qquad \Sigma = \sigma^2 I$$

$$P_0 = 0.5 \qquad \Sigma_0 = 10I$$

$$\mu_0 = (-0.9,-1.4,-0.4,-0.9,0.6,-0.8,-1.6,-1.2,0.6,-0.8)^T.$$

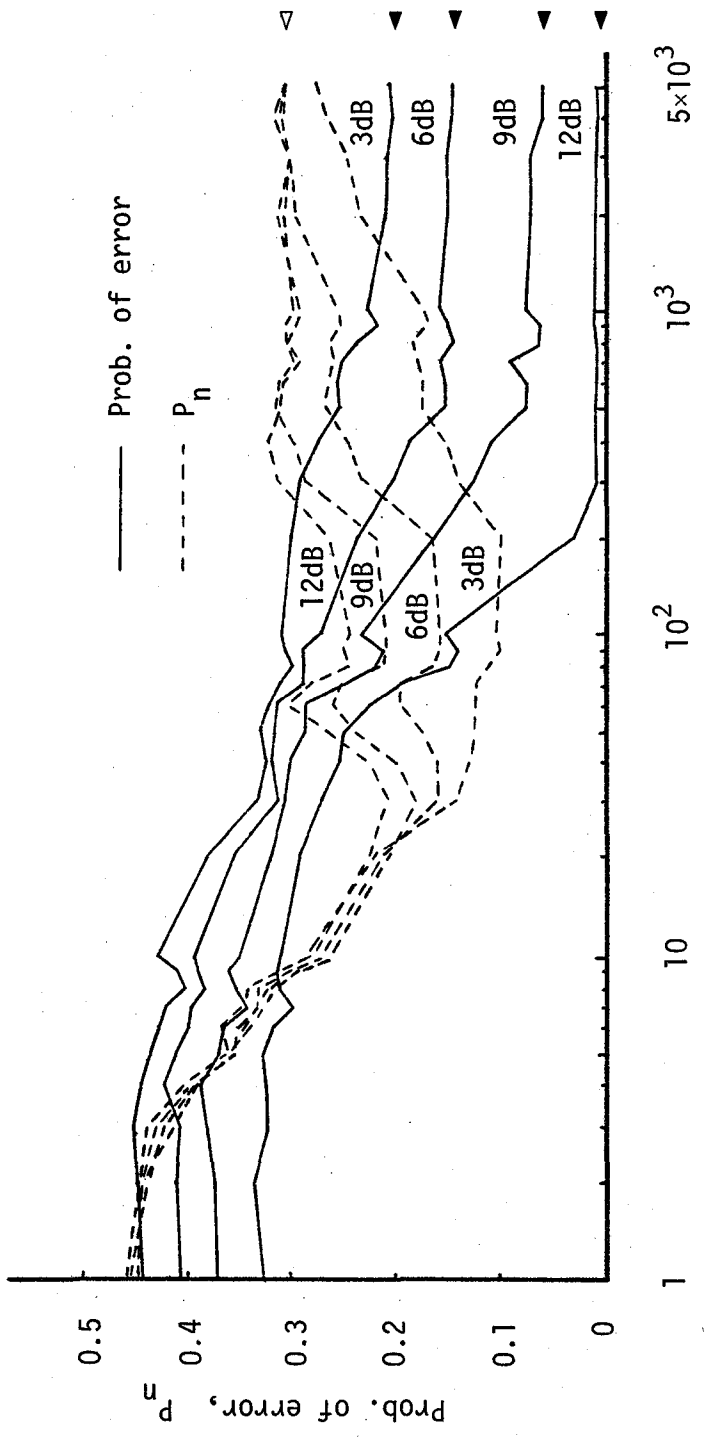


Fig. 4.2 Signal detection by WDDM.

Examples of the estimates of the signal vector and the covariance matrix at k = 5000 are

$$\mu_{5000} = (2.08,\ 2.02,\ 2.00,\ 2.11,\ 1.98,\ 2.18,\ 1.89,\ 2.03,\ 2.26,\ 1.86)^T$$

$$\Sigma_{5000} = \begin{bmatrix} 9.72 & 0.50 & -0.12 & 0.07 & -0.01 & 0.10 & -0.07 & 0.29 & 0.08 & 0.46 \\ & 10.05 & -0.46 & -0.19 & -0.51 & 0.09 & -0.36 & 0.29 & -0.04 & 0.13 \\ & & 9.70 & -0.04 & -0.14 & -0.11 & 0.06 & -0.14 & 0.04 & 0.44 \\ & & & 9.31 & -0.12 & -0.17 & 0.04 & -0.44 & -0.00 & 0.35 \\ & & & & 9.40 & 0.11 & 0.13 & 0.10 & 0.36 & -0.04 \\ & & & & & 10.22 & 0.02 & 0.06 & 0.10 & 0.05 \\ & & & & & & 9.33 & 0.13 & 0.14 & -0.15 \\ & & * & & & & & 9.60 & -0.04 & -0.55 \\ & & & & & & & & 9.52 & 0.07 \\ & & & & & & & & & 9.49 \end{bmatrix}$$

where S/N = 6dB. We here refer to the experimental result that the performance of DDM was very poor in both cases of decomposition of a finite mixture and signal detection.

4.5.3 INITIAL ESTIMATES PROBLEM

In the simulation of signal detection, various initial estimates of signal were used, and satisfactory results were obtained in every case. Especially, the signal vector $(2,2,\ldots,2)^T$ was estimated successfully even when $\mu_0 = (-0.5,-0.5,\ldots,-0.5)^T$. These results show that nonsupervised signal detection by WDDM without the knowledge of the signal vector, the probability of signal occurrence, and the covariance matrix of noise is made successfully almost independent of initial estimates of these parameters.

In the case of decomposition of a finite mixture, however, it may happen that the performance of WDDM is influenced by initial estimates particularly by those of mean vectors. This is of course

undesirable, though it is common to all nonsupervised learning algorithms based on minimization of certain criterion functions.

In order to investigate the influence of initial estimates of mean vectors on the performance of WDDM, computer experiments were made. In the experiments, a 2-category and 2-dimensional normal mixture with parameters

$$P^1 = P^2 = 0.5 \qquad \Sigma^1 = \Sigma^2 = 0.25I$$

$$\mu^1 = (1, 1)^T \qquad \mu^2 = (-1, -1)^T$$

was used. Initial estimates were

$$P_0^i = 1/3 \qquad \Sigma_0^i = I \quad (i=1,2,3) \ (N^{\#} = 3, N = 2).$$

Uniform random numbers in (-2, 2) were used as the initial estimates μ_0^i to give the results objectivity, and 97 successful results were obtained out of 100 sets of initial estimates. In three cases where WDDM failed in decomposing the mixture, great differences of the values of P_n^i arose at about n = 100. Considering these results, the following modification of our algorithm was made. That is, estimation of the *a priori* probabilities P^i is not performed until n = 100, and the other parameters are estimated as usual by using the fixed *a priori* probabilities $P_n^i = P_0^i$. Then, after n = 100 the full scale estimation of all parameters is performed by using P_0^i, μ_{100}^i, and Σ_{100}^i as initial estimates. This modified algorithm succeeded on decomposing the mixture in all the cases of the above 100 initial estimates. Fig. 4.3 shows

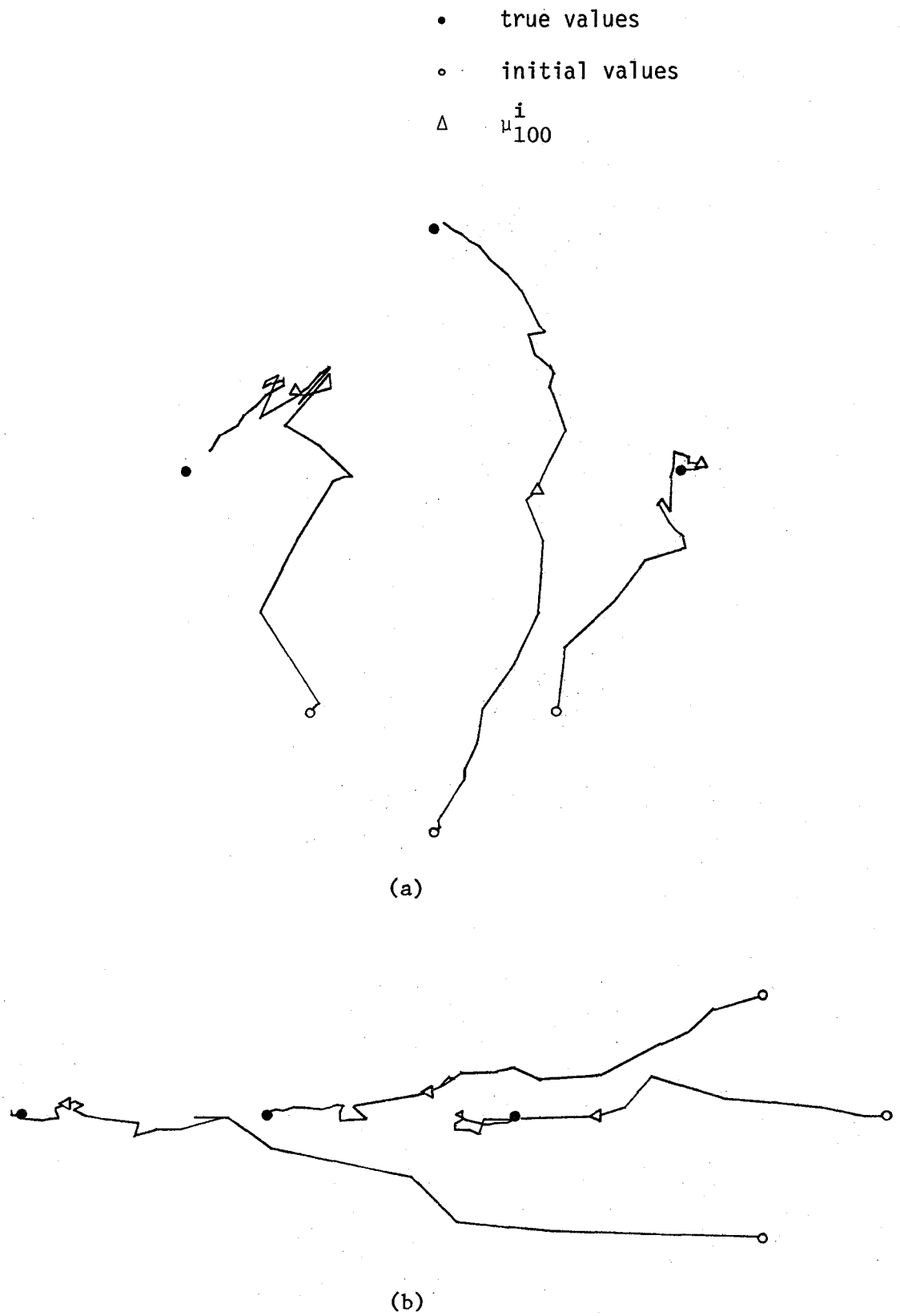


Fig. 4.3 Learning processes of mean values (loci of μ_n^i).

the learning processes of μ_n^i of the modified algorithm using the initial estimates far from the true values, where the *a priori* probabilities and the covariance matrices are unknown. In the figure, the marks •, ∘, and Δ designate the true values, the initial estimates, and the estimates at n = 100 of the mean vectors, respectively. All the results obtained above show that WDDM is an effective self-learning algorithm which is easy to execute independent of the dimensionality.

4.6 CONCLUSION

In this chapter, a nonsupervised parametric learning algorithm has been proposed which is called WDDM (weighted-decision-directed method). WDDM is an extended version of DDM, and it performs self-learning by regarding the k-th input pattern x_k as belonging to all categories with the weights $p(c^i | x_k, \chi_{k-1})$. Convergence of the algorithm was proved by employing convergence property of bounded martingales. In order to verify the efficiency of WDDM, computer simulation of the learning processes of the algorithm was made in decomposing multidimensional normal mixtures and in detecting a signal embedded in Gaussian noise, and satisfactory results were obtained.

We have discussed nonsupervised parametric learning in this chapter. The last three chapters will treat nonsupervised nonparametric learning.

CHAPTER 5

NONPARAMETRIC LEARNING WITHOUT A TEACHER

—— TWO-CATEGORY PROBLEM ——

5.1 INTRODUCTION

The problem of constructing nonsupervised nonparametric learning algorithms is rather difficult, since no *a priori* knowledge of pattern distribution is available. Most algorithms reported previously were based on analyses of stored sample patterns, so that they were rather complicated and comparatively difficult to execute. In this chapter we consider a nonsupervised nonparametric algorithm for designing a linear discriminant function (LDF) by limiting the discussion to 2-category problem, where no sample pattern is stored.

D. B. Cooper and P. W. Cooper [11] and Shimura and Imai [81] have discussed the first principal component in the problem of constructing an LDF $W^TX = \theta$ without knowing the *a priori* probability of each category. As pointed out in the above literatures, the first principal component of pattern distribution can be used as the weight vector W. In our system an algorithm due to Krasurina [42] for estimating the first principal component is employed and its simpler convergence proof is presented. In order to obtain a reasonable LDF, the threshold value θ must be determined appropriately. Now assume that we have a set of one-dimensional patterns with the probability density shown in Fig. 5.1. Considering that we intend to design a nonparametric algorithm, it is reasonable to set the threshold value θ at the minimum point of the

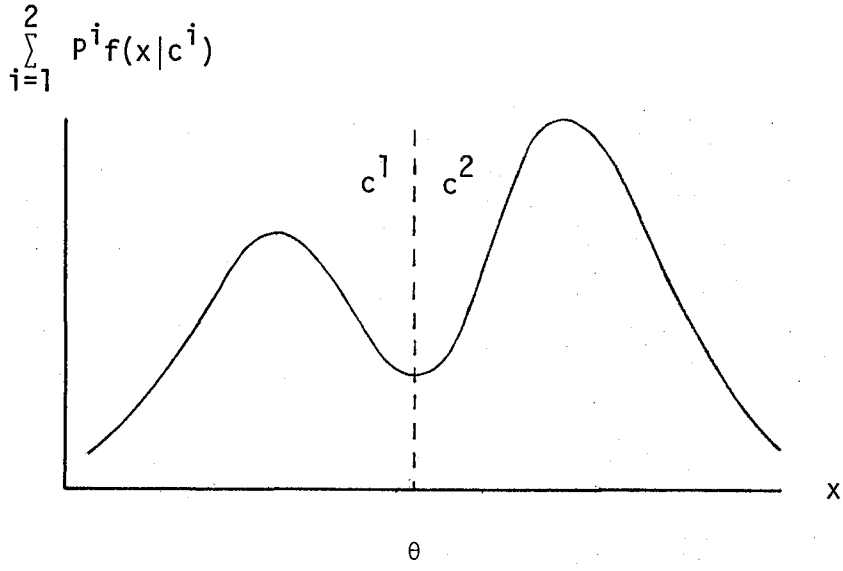


Fig. 5.1 Threshold value θ.

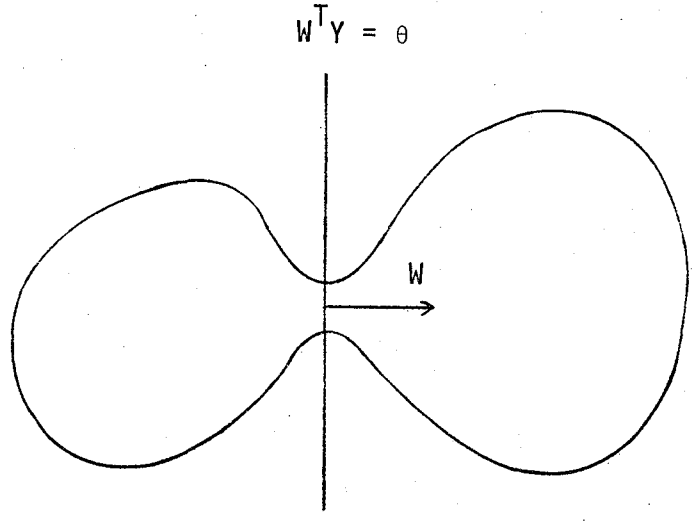


Fig. 5.2 A reasonable linear discriminant function for two-dimensional distribution.

mixture density. Therefore, the problem of determination of the threshold value is reduced to that of estimating a minimum point of a density function.

Generally speaking, stochastic approximation method is useful for the estimation problem of an extremum point of an unknown function. Unfortunately, it cannot be applied to the problem concerning probability density function (pdf). Recently, Wassel and Sklansky [91] have proposed Window Training Procedure (WTP), which makes it possible to treat the estimation problem as to pdf. However, WTP is an algorithm for estimating a unique intersection of two unknown pdf's with supervision. In this chapter, the author proposes a nonsupervised algorithm for estimating a unique maximum point of an unknown pdf by extending WTP.

In the case of multidimensional distribution, the mixture pdf does not have any minimum point but has a saddle point when the pdf of each category is unimodal. From Fig. 5.2 it is seen that the neck between the two clusters corresponds to a saddle point of the mixture pdf and that the threshold value θ should be so determined that the LDF $W^TX = \theta$ pass through the neck. In order to do this, the saddle point must be estimated, but it is difficult to detect directly. We here note that a minimum point of the one-dimensional pdf produced by projecting the pattern space on to the first principal component W corresponds to the neck of interest. Then our estimation algorithm can be used for obtaining the threshold value θ.

From the above discussion one can see that some assumptions are needed to obtain a reasonable LDF. That is, we assume that the patterns of each category are well clustered and with unimodal pdf. These

assumptions are usually met, though they impose some restrictions on pattern distribution. In this chapter, all considerations are made under the above assumptions.

5.2 LEARNING OF THE WEIGHT VECTOR W

Let Σ be the covariance matrix of the mixture distribution of input patterns, and let λ_M and W be the largest eigenvalue and the first principal component (the eigenvector corresponding to λ_M) of the distribution, respectively. Then, we have

$$\lambda_M = \max_{\|\eta\|=1} \eta^T \Sigma \eta = W^T \Sigma W. \tag{5.1}$$

By using this property the following theorem about estimation algorithm of W is obtained:

Theorem 5.1: Assume that the following conditions are satisfied:

$$\sum_{n=1}^{\infty} a_n = \infty \qquad \sum_{n=1}^{\infty} a_n^2 < \infty$$

$$E[\Sigma_n] = \Sigma \qquad E[\|\Sigma_n\|^2] < \infty.$$

Then, for an arbitrary vector $V_1 (\neq 0)$, W_n defined as

$$\begin{cases} V_{n+1} = V_n + a_n \left(\Sigma_n V_n - \dfrac{V_n^T \Sigma_n V_n}{V_n^T V_n} V_n \right) \\ W_{n+1} = \dfrac{V_{n+1}}{\|V_{n+1}\|} \end{cases} \tag{5.2}$$

converges to the eigenvector corresponding to λ_M (the first principal component) of Σ with probability one.

Proof: Considering that

$$\|V_{n+1}\|^2 = \|V_n\|^2 + a_n^2\|\Sigma_n V_n - \frac{V_n^T \Sigma_n V_n}{V_n^T V_n} V_n\|^2,$$

we have

$$E[\|V_{n+1}\|^2 | V_n] \leq \|V_n\|^2 (1 + a_n^2 E[\|\Sigma_n\|^2]).$$

Let define v_n as

$$v_n = \prod_{i=n}^{\infty} (1 + a_i^2 E[\|\Sigma_i\|^2])\|V_n\|^2. \tag{5.3}$$

Then, we have

$$\begin{aligned} E[v_{n+1} | V_n] &= E[\prod_{i=n+1}^{\infty} (1 + a_i^2 E[\|\Sigma_i\|^2])\|V_{n+1}\|^2 | V_n] \\ &\leq \prod_{i=n+1}^{\infty} (1 + a_i^2 E[\|\Sigma_i\|^2])(1 + a_n^2 E[\|\Sigma_n\|^2])\|V_n\|^2 \\ &= v_n. \end{aligned} \tag{5.4}$$

From the assumptions we also have

$$\sum_{i=1}^{\infty} \log(1 + a_i^2 E[\|\Sigma_i\|^2]) < \infty.$$

Therefore, we obtain

$$E[|v_1|] = \prod_{i=1}^{\infty} (1 + a_i^2 E[\|\Sigma_i\|^2])\|V_1\|^2 < \infty. \tag{5.5}$$

From (5.4) and (5.5) and $E[v_n] \geq 0$, the sequence $\{v_n\}$ turns out to be a semi-martingale. Hence, we have [15]

$$P[\lim_{n \to \infty} \|V_n\|^2 = \gamma] = 1, \qquad \text{for some } \gamma. \tag{5.6}$$

By a simple manipulation, we have

$$E[(W^T V_{n+1})^2 | V_n] = (W^T V_n)^2 + 2a_n (W^T V_n)^2 (\lambda_M - \frac{V_n^T \Sigma V_n}{V_n^T V_n}) + a_n^2 E[(W^T \Sigma_n V_n - \frac{V_n^T \Sigma_n V_n}{V_n^T V_n} W^T V_n)^2 | V_n]. \tag{5.7}$$

Considering that (5.1) is equivalent to

$$\lambda_M = \max_{\eta} \frac{\eta^T \Sigma \eta}{\eta^T \eta}, \tag{5.8}$$

we obtain

$$E[(W^T V_{n+1})^2 | V_n] \geq (W^T V_n)^2. \tag{5.9}$$

We also obtain from (5.6)

$$E[\|V_n\|^2] < \infty,$$

and hence,

$$E[(W^T V_n)^2] < \infty, \quad \text{with probability one.} \tag{5.10}$$

We here note that

$$E[\,(W^T V_{n+1})^2 \,|\, V_1, V_2, \ldots, V_n\,] \geq W^T V_1 + 2 \sum_{i=1}^{n} a_i (W^T V_i)^2 \left(\lambda_M - \frac{V_i^T \Sigma V_i}{V_i^T V_i}\right). \tag{5.11}$$

By using (5.8), (5.10), and (5.11) and $\sum a_i = \infty$, we have

$$\lambda_M - \frac{V_\infty^T \Sigma V_\infty}{V_\infty^T V_\infty} = 0 \tag{5.12}$$

or

$$(W^T V_\infty)^2 = 0 \tag{5.13}$$

with probability one. However, from (5.9) $E[(W^T V_n)^2]$ is monotone non-decreasing, so that the probability that (5.12) holds is one. Hence, from the definition we obtain

$$P[\lim_{n \to \infty} W_n = W\,] = 1. \tag{5.14}$$

This proves the theorem. (*Q.E.D.*)

Thus, a learning algorithm for estimating the weight vector W of the LDF $W^T X = \theta$ has been obtained. In the following sections, a learning algorithm for the threshold value θ is discussed.

5.3 ESTIMATION OF A MAXIMUM POINT OF PDF

As is discussed in Section 5.1, it is necessary to estimate a minimum point of the pdf of input patterns in order to obtain a reasonable threshold value. We now present an algorithm for estimating a maximum point of pdf.

Let us define T_{n+1} as follows:

$$T_{n+1} = T_n + a_n Z_n \tag{5.15}$$

where T_1 is arbitrarily determined and

$$Z_n = \sqrt{2/\pi}\ \frac{x_n - T_n}{b_n} \exp[\ -(\ x_n - T_n)^2/(2b_n^2\ )] \tag{5.16}$$

In (5.16), Z_n gives some information about the gradient of the pdf of the input patterns, and then hill-climbing method is used. As to the above algorithm the following theorem is obtained:

Theorem 5.2: Assume that the following restriction conditions are satisfied:

1) $a_n,\ b_n > 0$

2) $\lim_{n\to\infty} a_n = \lim_{n\to\infty} b_n = 0$

3) $\sum_{n=1}^{\infty} a_n b_n^2 = \infty$

4) $\sum_{n=1}^{\infty} a_n^2 b_n < \infty$

5) x_n is the n-th random sample vector from an unknown one-dimensional pdf p(x).

6) The derivative of p(x) is continuous and bounded at every point.

7) p(x) takes its unique maximum at $x = t^{\#}$ $(|t^{\#}| < \infty)$.

Then, T_n defined in (5.15) converges to $t^{\#}$ with probability one. That is,

$$P[\lim_{n \to \infty} T_n = t^{\#}] = 1. \tag{5.17}$$

Proof: See Appendix 5.1.

5.4 LEARNING OF THE THRESHOLD VALUE θ AND LDF

Let Y, Y_n, and W be a random vector with the pdf p(Y), the n-th random sample vector from p(Y), and the maximal eigenvector, respectively. Then, an LDF is wriiten as $W^TY = \theta$, where θ is the minimum point of the pdf $p(W^TY)$. In the rest of this chapter, it is assumed that the expected value of W^TY lies between the two maximum points of $p(W^TY)$. Fig. 5.3 shows an example of such pdf, where G and σ are the mean and the standard deviation of $p(W^TY)$, respectively, M^0 and M^2 are both maximum points of $p(W^TY)$, and M^1 is the minimum point of $p(W^TY)$. We have already obtained an algorithm for estimating a maximum point of one-dimensional pdf in the last section. By changing the sign of the gradient estimator Z_n, an algorithm for estimating a minimum point of an unknown pdf is obtained as follows:

$$T_{n+1} = T_n - a_n Z_n \tag{5.18}$$

It is seen from Fig. 5.3 that (5.18) needs a certain modification in order to estimate the minimum point M^1, since T_n diverges with positive probability when the initial estimate T_0 is not between M^0 and M^2. To avoid this difficulty the minimum point M^1 is sought in the interval (M^0, M^2) and the initial estimate T_0 is set at G.

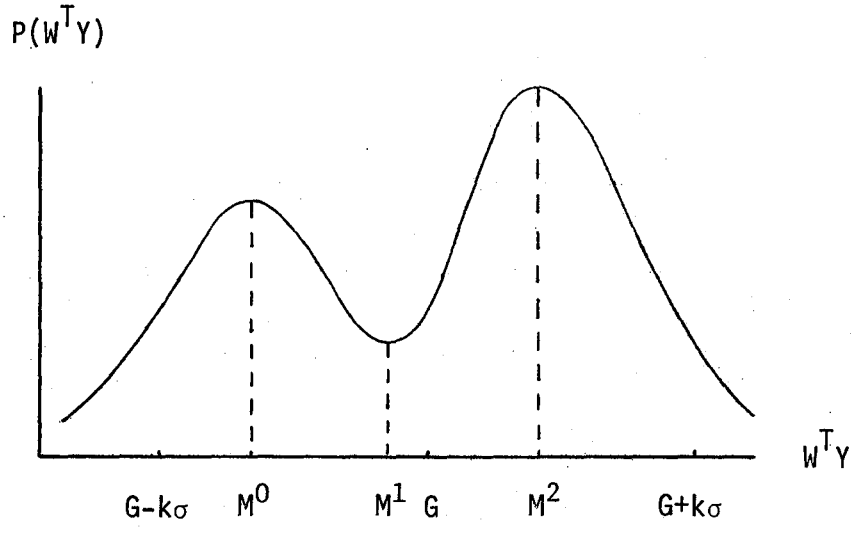


Fig. 5.3 Locations of several parameters.

Let T_n^i be the estimates of M^i (i=0,1,2). We now have the following design algorithm of an LDF:

$$\mu_n = \mu_{n-1} + \frac{1}{n}(Y_n - \mu_{n-1}) \tag{5.19}$$

$$\Sigma_n = \Sigma_{n-1} + \frac{1}{n-1}(\frac{n}{n-1}(Y_n - \mu_n)(Y_n - \mu_n)^T - \Sigma_{n-1}) \tag{5.20}$$

$$V_n = V_{n-1} + a_{n-1}(\Sigma_{n-1}V_{n-1} - \frac{V_{n-1}^T \Sigma_{n-1} V_{n-1}}{V_{n-1}^T V_{n-1}} V_{n-1}) \tag{5.21}$$

$$W_n = \frac{V_n}{\|V_n\|} \qquad G_n = W_n^T \mu_n \qquad \sigma_n = (W_n^T \Sigma_n)^{1/2}$$

$$T_0^i = G_{n\#} + (i - 1)k\sigma_{n\#} \tag{5.22}$$

$$T_m^i = T_{m-1}^i + (-1)^i a_m Z_m^i \tag{5.23}$$

$$T_m^0 = \begin{cases} T_m^0, & \text{if } T_m^0 \le G_{n\#+m} \\ G_{n\#+m} - k\sigma_{n\#+m}, & \text{otherwise} \end{cases} \tag{5.24}$$

$$T_m^1 = \begin{cases} T_m^1, & \text{if } T_m^0 < T_m^1 < T_m^2 \\ G_{n\#+m}, & \text{otherwise} \end{cases} \tag{5.25}$$

$$T_m^2 = \begin{cases} T_m^2, & \text{if } T_m^2 \ge G_{n\#+m} \\ G_{n\#+m} + k\sigma_{n\#+m}, & \text{otherwise} \end{cases} \tag{5.26}$$

$$Z_m^i = \sqrt{2/\pi}\,\frac{x_m - T_m^i}{b_m}\exp[\,-(x_m - T_m^i)^2/(2b_m^2)\,] \tag{5.27}$$

$$x_m = W_{n^\#+m}^T \cdot Y_{n^\#+m}. \tag{5.28}$$

Then, the LDF is defined as

$$\begin{cases} W_n^T Y = G_n & 1 \le n \le n^\# \\ W_{n^*+m}^T \cdot Y = T_m^1 \quad (m \ge 0), & n \ge n^\#. \end{cases} \tag{5.29}$$

Learning up to the $n^\#$-th step is for getting initial estimates of T_n^i. After the time $n^\#$, the estimation of the minimum point is performed by using these initial estimates. As is described above, T_m^1 is restricted in the interval (T_m^0, T_m^2), so that T_m^1 converges with probability one to M^1 which is the unique minimum point in (M^0, M^2). The LDF obtained in (5.29) works well even if the *a priori* probability of each category is unknown, since the threshold value θ is so determined that the hyperplane of the LDF pass through the minimum point of $p(W^TY)$ corresponding to the neck between the two pattern distributions.

5.5 COMPUTER EXPERIMENTS

In the computer study, 20-dimensional normal distribution $\sum_{i=1}^{2} P^i N(x|\,\mu^i, \Sigma^i)$ was used, where

$$P^1 = 0.7 \qquad P^2 = 0.3 \qquad \Sigma^1 = \Sigma^2 = s^2 I$$

$$\mu^1 = (2,2,\ldots,2)^T \qquad \mu^2 = (0,0,\ldots,0)^T.$$

Fig. 5.4 shows the learning processes of W_n, and Fig. 5.5 shows the probabilities of error of the LDF. Our algorithm was performed in the case

$$a = b = 2.5$$

$$\alpha = 0.5 \qquad \beta = 0.25$$

$$n^{\#} = 100 \qquad k = 1.5.$$

In Fig. 5.5, arrows indicate the probabilities of error of the optimal machine. From this figure one can see that the probabilities of error of our LDF converge to those of the optimal machine in all SN-ratios.

5.6 CONCLUSION

We have discussed a nonparametric learning algorithm for designing an LDF without supervision. An algorithm for estimating the eigenvector corresponding to the largest eigenvalue was presented along with its simple convergence prrof in Theorem 5.1. Also, we obtained an algorithm for estimating a maximum point of an unknown one-dimensional pdf and its convergence proof in Theorem 5.2. By using these algorithms, a design algorithm of an LDF which works well even if the *a priori* probabilities are unknown has been obtained. The effectiveness of our algorithms were verified by computer experiments.

This chapter has been concerned with the 2-category problem. Next chapter will deal with the multi-category problem by extending the mode estimation algorithm discussed here.

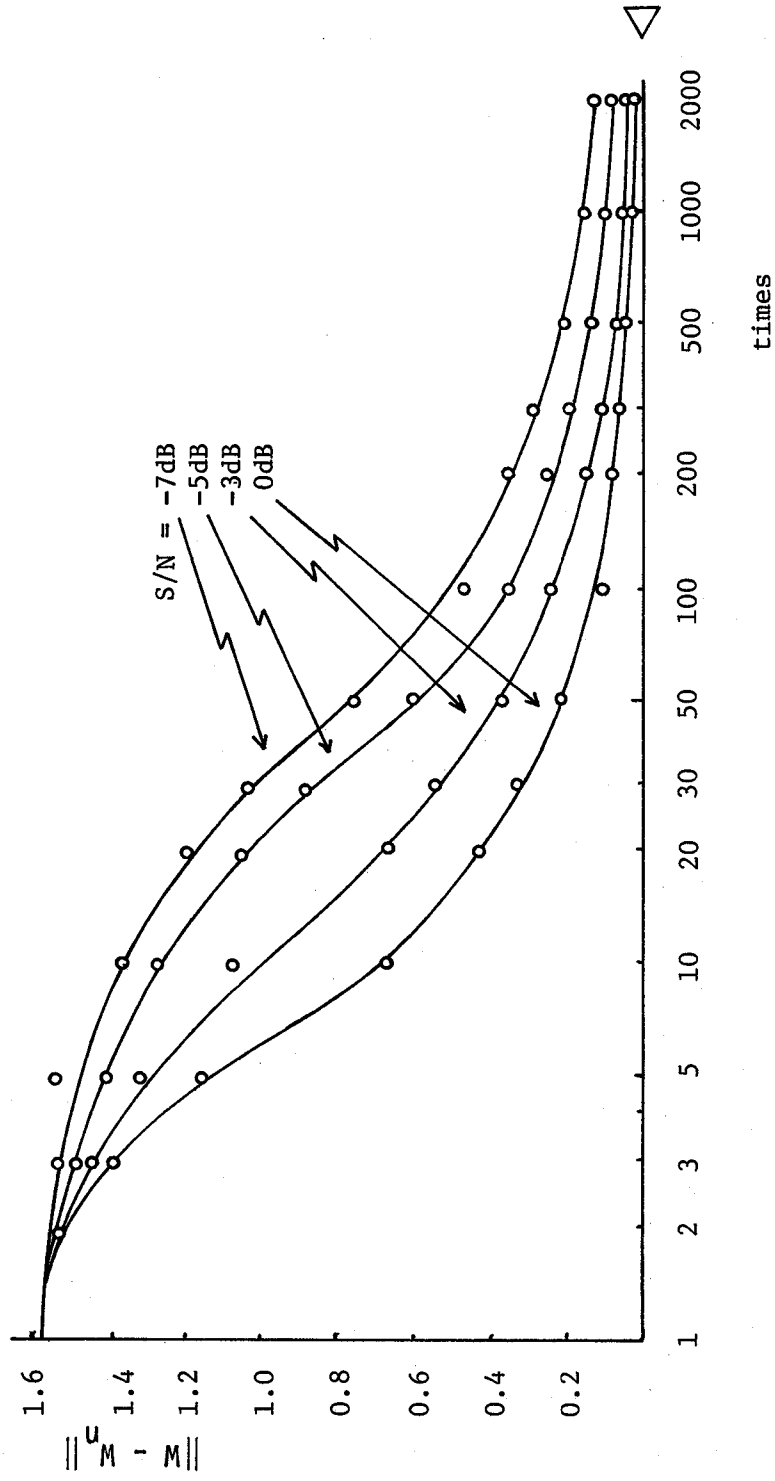


Fig. 5.4 Convergence process of W_n^u.

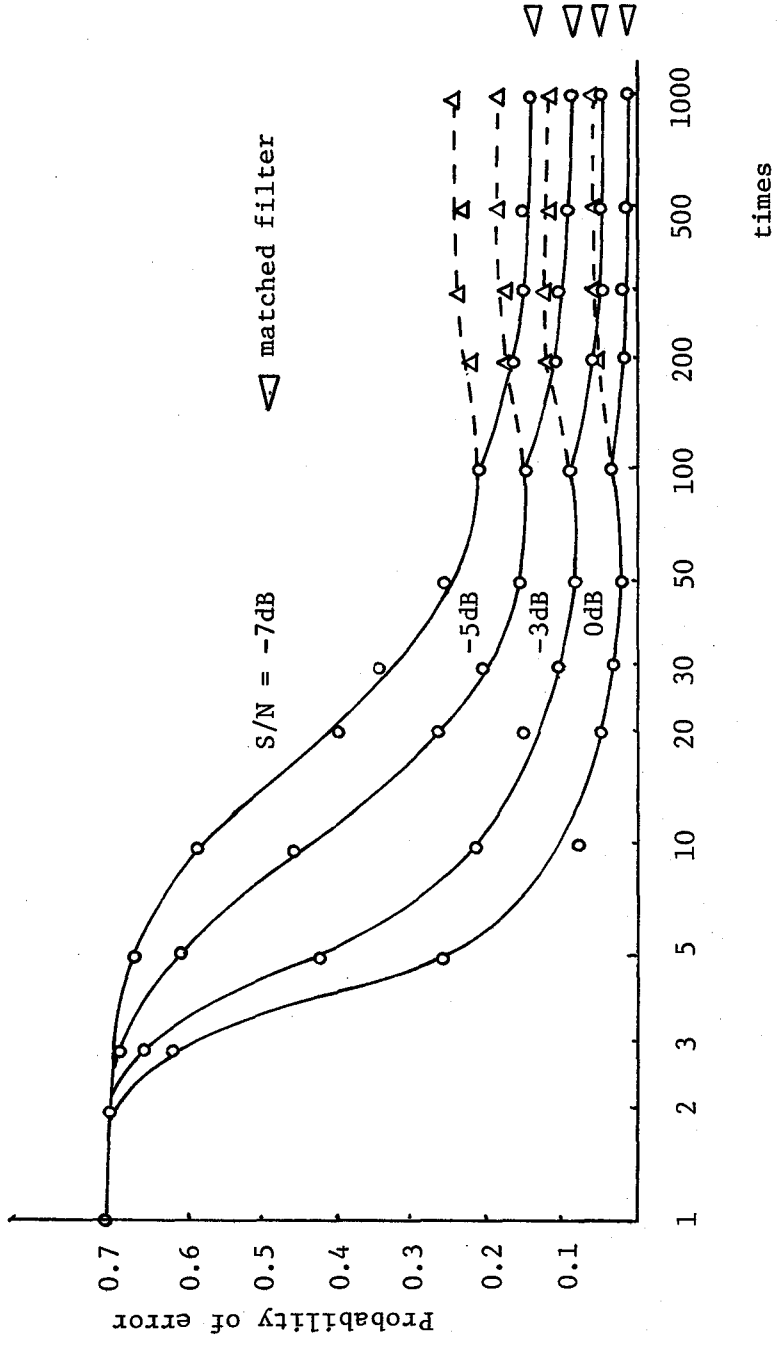


Fig. 5.5 Learning process of linear discriminant function.

APPENDIX 5.1 PROOF OF *Theorem 5.2*

Proof: First, we consider the gradient estimator Z_n. From

$$\frac{1}{2b_n^2} E[\, Z_n \mid T_n = t \,] = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \frac{x - t}{b_n^3} \exp[\, -\frac{(x-t)^2}{2b_n^2} \,]\, p(x)dx$$

$$= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\, b_n} \exp[\, -\frac{(x-t)^2}{2b_n^2} \,]\, p'(x)dx \qquad (5.30)$$

we have

$$p_n(t, b_n) \equiv \frac{1}{2b_n^2} E[\, Z_n \mid T_n = t \,] \xrightarrow[n \to \infty]{} p'(t) \qquad (5.31)$$

$$t_n^{\#} \xrightarrow[n \to \infty]{} t^{\#} \qquad (5.32)$$

where $t_n^{\#}$ is the zero point of $p_n(t, b_n)$, that is,

$$p_n(t_n^{\#}, b_n) = 0. \qquad (5.33)$$

Let $\sigma_{Z_n}^2$ be the variance of Z_n, then

$$\sigma_{Z_n}^2 \le E[\, Z_n^2 \,]$$

$$= E_{T_n}[\, \int_{-\infty}^{\infty} \frac{2}{\pi} \frac{(x - T_n)^2}{b_n^2} \exp[\, -\frac{(x-T_n)^2}{b_n^2} \,]\, p(x)dx \,]$$

$$= E_{T_n}\left[\frac{2}{\sqrt{\pi} b_n} \int_{-\infty}^{\infty} \frac{(x - T_n)^2}{\sqrt{\pi}\, b_n} \exp\left[- \frac{(x - T_n)^2}{b_n^2} \right] p(x)dx \right]$$

$$\leq \frac{2}{\sqrt{\pi}} M\, b_n$$

and

$$\sigma_{Z_n}^2 \;/\; b_n \;\leq\; \frac{2}{\sqrt{\pi}} M \tag{5.34}$$

where

$$\sup_x \left| p(x) \right| \;\leq\; M < \infty.$$

From (5.30)

$$\frac{1}{2b_n^2} \left| E[Z_n \,|\, T_n] \right| < M' \tag{5.35}$$

where

$$\sup_x \left| p'(x) \right| \;\leq\; M' < \infty.$$

Furthermore, from (5.31) and (5.32) and the constraints 6) and 7), we have

$$\exists N_1,\ \forall n > N_1 \Longrightarrow \begin{cases} \dfrac{1}{b_n^2} E[Z_n \,|\, T_n] > 0 & -\infty < T_n < t_n^{\#} \qquad (5.36) \\ \dfrac{1}{b_n^2} E[Z_n \,|\, T_n] < 0 & t_n^{\#} < T_n < \infty. \qquad (5.37) \end{cases}$$

Now, we rewrite (5.15) as

$$T_{n+1} = T_n + \xi_n + a_n E[Z_n \,|\, T_n] \tag{5.38}$$

where

$$\xi_n = a_n \left(Z_n - E[\, Z_n \,|\, T_n \,] \right).$$

Let define ζ_n as

$$\zeta_n = E[\, \xi_n \,|\, \xi_1, \xi_2, \ldots, \xi_{n-1} \,]$$

then we have

$$\begin{aligned} \zeta_n &= a_n \left(E[\, Z_n \,|\, \xi_1, \xi_2, \ldots, \xi_{n-1} \,] \right. \\ &\qquad \left. - E[\, Z_n \,|\, T_n, \xi_1, \xi_2, \ldots, \xi_{n-1} \,] \right) \\ &= a_n \left(E[\, Z_n \,|\, T_n \,] - E[\, Z_n \,|\, T_n \,] \right) \\ &= 0. \end{aligned}$$

Hence,

$$\sum_{n=1}^{\infty} \zeta_n = 0 \tag{5.39}$$

From (5.34) and the constraint 4), we also have

$$\begin{aligned} \sum_{n=1}^{\infty} Var[\, \xi_n] &= \sum_{n=1}^{\infty} a_n^2 \left(E[\, \left(Z_n - E[\, Z_n \,|\, T_n \,] \right)^2] \right. \\ &\qquad \left. - E[\, Z_n - E[\, Z_n \,|\, T_n \,] \,] \right) \\ &= \sum_{n=1}^{\infty} a_n^2 \left(E[\, Z_n^2 \,] - E^2[\, Z_n \,] \right) \end{aligned}$$

$$= \sum_{n=1}^{\infty} a_n^2 \, \sigma_{Z_n}^2$$

$$< \frac{2}{\sqrt{\pi}} M \sum_{n=1}^{\infty} a_n^2 \, b_n < \infty. \qquad (5.40)$$

It is seen from (5.39) and (5.40) that $\sum \xi_n$ converges with probability one [46]. Considering that (5.38) can be written as

$$T_{n+1} = T_1 + \sum_{j=1}^{n} \xi_j + \sum_{j=1}^{n} a_j \, E[\, Z_j \,|\, T_j \,],$$

one can see the existence of a random variable U such that

$$P[\, \lim_{n \to \infty} (T_{n+1} - \sum_{j=1}^{n} a_j \, E[\, Z_j \,|\, T_j \,]) = U \,] = 1. \qquad (5.41)$$

Next we shall show

$$P[\, \lim_{n \to \infty} T_n = \pm\infty \,] = 0.$$

Now, suppose that

$$P[\, \lim_{n \to \infty} T_n = \infty \,] > 0.$$

Then, from the constraint 7) and (5.32), we have

$$\exists N_2, \forall n > N_2 \Longrightarrow \quad T_n > t_n^{\#}.$$

Therefore, from (5.37)

$$\sum_{j=1}^{\infty} a_j \, E[\, Z_j \,|\, T_j \,] < \infty.$$

Hence, we have

$$P[\, \lim_{n \to \infty} (T_n - \sum_{j=1}^{n} a_j \, E[\, Z_j \,|\, T_j \,]) = \infty \,] > 0,$$

which contradicts (5.41). Similarly, in the case of $P[\, \lim T_n = -\infty] > 0$, a contradiction is also derived. Therefore, we have

$$P[\, \lim_{n \to \infty} T_n = \pm\infty] = 0. \tag{5.42}$$

Next, we shall show

$$P[\, \lim_{n \to \infty} T_n = T \,] = 1, \quad \text{for some } T.$$

Suppose that

$$P[\, \lim_{n \to \infty} T_n = T \,] < 1, \quad \text{for any } T.$$

Then, from (5.42) there exists such a sequence $\{T_n\}$ with positive probability that satisfies (5.41) and

$$\lim_{N \to \infty} \inf_{n \geq N} T_n \quad < \quad \lim_{N \to \infty} \sup_{n \geq N} T_n. \tag{5.43}$$

The following discussion is made in the case where

$$\lim_{N \to \infty} \sup_{n \geq N} T_n \quad > \quad t^{\#}.$$

In the other case where $\limsup T_n < t^{\#}$, a similar discussion can be made. Now, we can take two numbers γ and δ such that

$$\gamma > t^{\#} \tag{5.44}$$

$$\lim_{N\to\infty} \inf_{n \geq N} T_n < \gamma < \delta < \lim_{N\to\infty} \sup_{n \geq N} T_n \tag{5.45}$$

From (5.32) and (5.41) and the constraint 3), one can see that both γ and δ satisfy the following relations:

$$\exists N_3, \quad k > m > N_3 \Longrightarrow a_m b_m^2 \leq \frac{\delta - \gamma}{4M'} \tag{5.46}$$

$$\left| T_k - T_m - \sum_{j=m}^{k-1} a_j E[Z_j | T_j] \right| \leq \frac{\delta - \gamma}{2} \tag{5.47}$$

$$\gamma > t_m^{\#}. \tag{5.48}$$

Then, considering (5.43) and $|T_{n+1} - T_n| \longrightarrow 0$, we can take both k and m such that

$$T_m < \gamma \tag{5.49}$$

$$T_k > \delta \tag{5.50}$$

$$\gamma \leq T_j \leq \delta \qquad (m < j < k) \tag{5.51}$$

Therefore, from (5.37), (5.48), and (5.51) we have

$$T_j > t_m^{\#},$$

and hence,

$$S \equiv \sum_{j=m+1}^{k} a_j E[\, Z_j \,|\, T_j \,] < 0. \tag{5.52}$$

Furthermore, from (5.35) and (5.46) we obtain

$$a_m \,|\, E[\, Z_m \,|\, T_m \,] \,| \leq 2a_m b_m^2 M' \leq \frac{\delta - \gamma}{2}.$$

Therefore,

$$-S - \frac{\delta - \gamma}{2} \leq -\sum_{j=m}^{k-1} a_j E[\, Z_j \,|\, T_j \,] \leq -S + \frac{\delta - \gamma}{2}. \tag{5.53}$$

Now, considering that

$$T_k - T_m > \frac{\delta - \gamma}{2} > 0$$

and

$$-S > 0,$$

by substituting (5.53) into (5.47) we obtain

$$T_k - T_m - S - \frac{\delta - \gamma}{2} \leq \frac{\delta - \gamma}{2},$$

and hence,

$$T_k - T_m \leq \delta - \gamma, \tag{5.54}$$

which contradicts (5.49) and (5.50). Therefore, there exists a

random variable T such that

$$P[\lim_{n\to\infty} T_n = T] = 1. \tag{5.55}$$

Finally, we shall show that

$$P[T = t^{\#}] = 1.$$

The following discussion is made in the case where

$$P[T < t^{\#}] > 0.$$

In the other case where $P[T > t^{\#}] > 0$, a similar discussion can be made. Then, there exist two numbers r and s such that

$$-\infty < r < s < t^{\#} \tag{5.56}$$

$$P[r < T < s] > 0. \tag{5.57}$$

Therefore, from (5.32) we have

$$\exists N_4,\ \forall n > N_4 \implies r \le T_n \le s < t_n^{\#}.$$

Hence, from (5.33) and (5.36)

$$\forall n > N_4, \exists \varepsilon > 0 \implies \frac{1}{b_n^2} E[Z_n | T_n] = 2p_n(T_n, b_n) > \varepsilon.$$

Consequently, by using 3) we obtain

$$\forall n \ (< \infty), \quad \sum_{j=n}^{\infty} a_j \, E[\, Z_j \,|\, T_j \,] > \varepsilon \sum_{j=n}^{\infty} a_j \, b_j^2 = \infty.$$

Hence,

$$\sum_{j=1}^{\infty} a_j \, E[\, Z_j \,|\, T_j \,] = \infty, \tag{5.58}$$

which contradicts (5.41) and (5.55). Therefore,

$$P[\, T = t^{\#} \,] = 1. \tag{5.59}$$

Hence, from (5.55) and (5.59) we obtain

$$P[\, \lim_{n \to \infty} T_n = t^{\#} \,] = 1,$$

which proves the theorem. (*Q.E.D.*)

CHAPTER 6

NONPARAMETRIC LEARNING WITHOUT A TEACHER BASED ON MODE ESTIMATION —— MULTI-CATEGORY PROBLEM ——

6.1 INTRODUCTION

Parametric learning schemes, with or without a teacher, have been studied very extensively over the last ten years, whereas nonparametric learning schemes without a teacher do not seem to have been considered to the same extent in spite of their importances. Braverman [6], Shimura and Imai [81], and the author (Chapter 5) have discussed the nonsupervised algorithm for obtaining linear discriminant functions in two-category problems. Braverman has derived a learning algorithm based on potential functions. Shimura and Imai have discussed a learning algorithm for estimating the principal component of the mixture distribution of input patterns to construct a linear discriminant function. In chapter 5, the author has presented a learning algorithm for obtaining a linear discriminant function by estimating the unique minimum point of a univariate probability density function (pdf). Although some interesting ideas that are useful for the two-category problem are considered in the above literatures, multi-category problems are not dealt with.

Cluster detection techniques applicable to the multi-category problem have been studied by Gitman and Levine [22], Gitman [23], Koontz and Fukunaga [41], Zahn [98], and Jarvis and Patrick [34]. In their

algorithms, all sample patterns are stored in order to make up for the *a priori* information about the distribution of the patterns. Such algorithms are rather different from those discussed here.

This chapter discusses a nonsupervised multi-category problem in terms of nonparametric learning where no input pattern is memorized. In order to achieve our purpose a cluster detection algorithm is considered under the assumption that there exists a one-to-one correspondence between clusters and categories. We also assume that each mode of the mixture pdf of input patterns represents each cluster. As is discussed by Gitman and Levine [22] and Gitman[23], this assumption is rather reasonable, since unimodal pdf's can represent quite general distributions of patterns. Therefore, it can be an efficient way to estimate modes of the pdf of input patterns for constructing discriminant functions by nonparametric learning without a teacher. The problem of seeking modes of a pdf has been considered by Parzen [63] and Ryzin [72]. In their algorithms, modes are obtained by using the estimated pdf. However, it is comparatively difficult to estimate pdf's and besides, the whole schemes become rather complicated because they consist of two stages. For this reason, the author proposes a new mode estimation algorithm in which the pdf is not necessary to be estimated beforehand.

For seeking modes of a pdf, the hill-climbing method can be a useful technique provided that an efficient gradient search technique with respect to pdf's is available. In this chapter, therefore, a hypercubic window function is considered which gives some information about the gradient of a multivariate and multimodal pdf without memorizing input patterns. As is well-known, a mode estimator goes up the slope

of the pdf according to the estimated gradient. The hyper-cubic window function can be considered as an extended version of the window function introduced by Wassel and Sklansky [91]. Their window function operates with the aid of a teacher, while our window function operates without supervision. Furthermore, their window function is made smaller whenever an input pattern is given in the pattern space, while our window function is made smaller only when an input pattern is observed within the window. A detailed discussion about our window function will be made in the following sections.

As is discussed above, the estimate of each mode can be considered as a good approximation of the location of each cluster. For this reason, by using the estimates of the modes of the mixture pdf, a minimum-distance classifier [62] is constructed which assigns each input pattern to the category (cluster) corresponding to its nearest mode.

A nonparametric signal detection problem is also discussed as an example of the two-category problem to compare our algorithm with others. Under the assumption that the distribution of noise is symmetric with respect to its mean vector, both the input signal and the mean vector of noise are estimated by using our mode estimation algorithm. Moreover, it is shown that the probability of the signal occurrence and the covariance matrix of noise can be estimated. The threshold value of the linear discriminant function of the signal detector is determined for the discriminant hyperplane to pass through the valley lying between the two clusters. By using all the estimates, an adaptive signal detector converging nearly to the optimal machine is designed without supervision.

Some results of computer simulation of our learning algorithms are presented. The results show that the performance of our methods compares favorably with that of other methods.

6.2 ALGORITHM FOR ESTIMATING ONE OF THE MODES OF PDF

In this section we consider a nonsupervised multi-category problem in terms of nonparametric learning under the assumption that $N^{\#}$, the upper bound of the number of categories contained in the mixture distribution, is given. As is discussed in Section 6.1, we take the approach of estimating the modes of the mixture distribution of input patterns. In the algorithm the mode is estimated by using a hyper-cubic window function proposed here.

6.2.1 NOTATION

X_n	n-th input pattern.
Z_n	n-th mode estimator.
$\zeta_n(Z_{n-1})$	n-th window function.
$m(n)$	Total number of input patterns observed within the window up to the n-th step.
$b_{m(n)}$	Size of the n-th window.
$a_{m(n)}$	A positive coefficient.
W_n^i	i-th subregion of the n-th window.
d_n^i	i-th direction that the n-th mode estimator can move in.

6.2.2 BASIC MECHANISMS OF THE HYPER-CUBIC WINDOW FUNCTION

A maximum point of an unknown function is generally found by the

hill-climbing method according to the estimated gradient of the function. Our problem is to find the maximum point of a pdf of input patterns, so that it is necessary to obtain the estimate of the gradient of the pdf for using the hill-climbing method.

Wassel and Sklansky have proposed the window training procedure (WTP) which finds the unique intersection of two unknown univariate pdf's with the aid of a teacher. The author here introduces a new hyper-cubic window function and discusses an algorithm for estimating one of the modes of an unknown multimodal* and multivariate pdf without supervision. The hyper-cubic window is made smaller only when an input pattern is observed within the window. Therefore, our method makes it possible to decrease the influence of both the input pattern sequences and the initial estimate of a mode on the performance of the mode estimation algorithm. This is because if the window is made smaller whenever an input pattern is given as in the WTP, it sometimes happens that almost all the input patterns cannot be observed within the window. In such a case the window becomes so small that the performance of the algorithm decreases, particularly when the pdf of interest is a multimodal and multivariate one, or when the initial estimate is far from the mode.

Now, the author proposes the following mode estimation algorithm using the hyper-cubic window function mentioned above:

---

* In spite of the assumption that each cluster consists of a unimodal pdf, a mode estimation algorithm applicable to multimodal pdf needs to be developed because the mixture pdf of input patterns is multimodal.

$$Z_{n+1} = Z_n + a_{m(n+1)}\zeta_{n+1}(Z_n) \tag{6.1}$$

where

$$\xi_{n+1}(Z_n) = \begin{cases} 1, & \text{if } |X^d_{n+1} - Z^d_n| \le b_{m(n)} \quad \text{for any } d \\ 0, & \text{otherwise} \end{cases} \tag{6.2}$$

$$m(n+1) = m(n) + \xi_{n+1}(Z_n), \qquad m(0) = 1 \tag{6.3}$$

$$\zeta^d_{n+1}(Z_n) = \begin{cases} b^{-2}_{m(n)}, & \text{if } \xi_{n+1}(Z_n) = 1 \text{ and } Z^d_n \le X^d_{n+1} \\ 0, & \text{if } \xi_{n+1}(Z_n) = 0 \\ -b^{-2}_{m(n)} & \text{if } \xi_{n+1}(Z_n) = 1 \text{ and } X^d_{n+1} < Z^d_n \end{cases} \tag{6.4}$$

and the superscript d indicates the d-th component of each vector.

Here the author gives an intuitive interpretation of the above algorithm before proceeding to its convergence proof. An example of a two-dimensional hyper-cubic window function is shown in Fig. 6.1, which consists of four regions (W^1_n, W^2_n, W^3_n, and W^4_n). The n-th mode estimator Z_n is located at the center of the n-th $2b_{m(n)} \times 2b_{m(n)}$ window. The window function $\zeta_n(Z_{n-1})$ takes vector values $(-b^{-2}_{m(n)}, -b^{-2}_{m(n)})$, $(-b^{-2}_{m(n)}, b^{-2}_{m(n)})$, $(b^{-2}_{m(n)}, -b^{-2}_{m(n)})$, $(b^{-2}_{m(n)}, b^{-2}_{m(n)})$, or (0, 0), according as the n-th input pattern is observed within the region W^1_n, W^2_n, W^3_n, W^4_n, or outside the window. When an input pattern is observed within the window, $b_{m(n)}$ decreases and then the window is made smaller. Note that $b_{m(n)} \longrightarrow 0$ as $m(n) \rightarrow \infty$. When an input pattern is not observed within

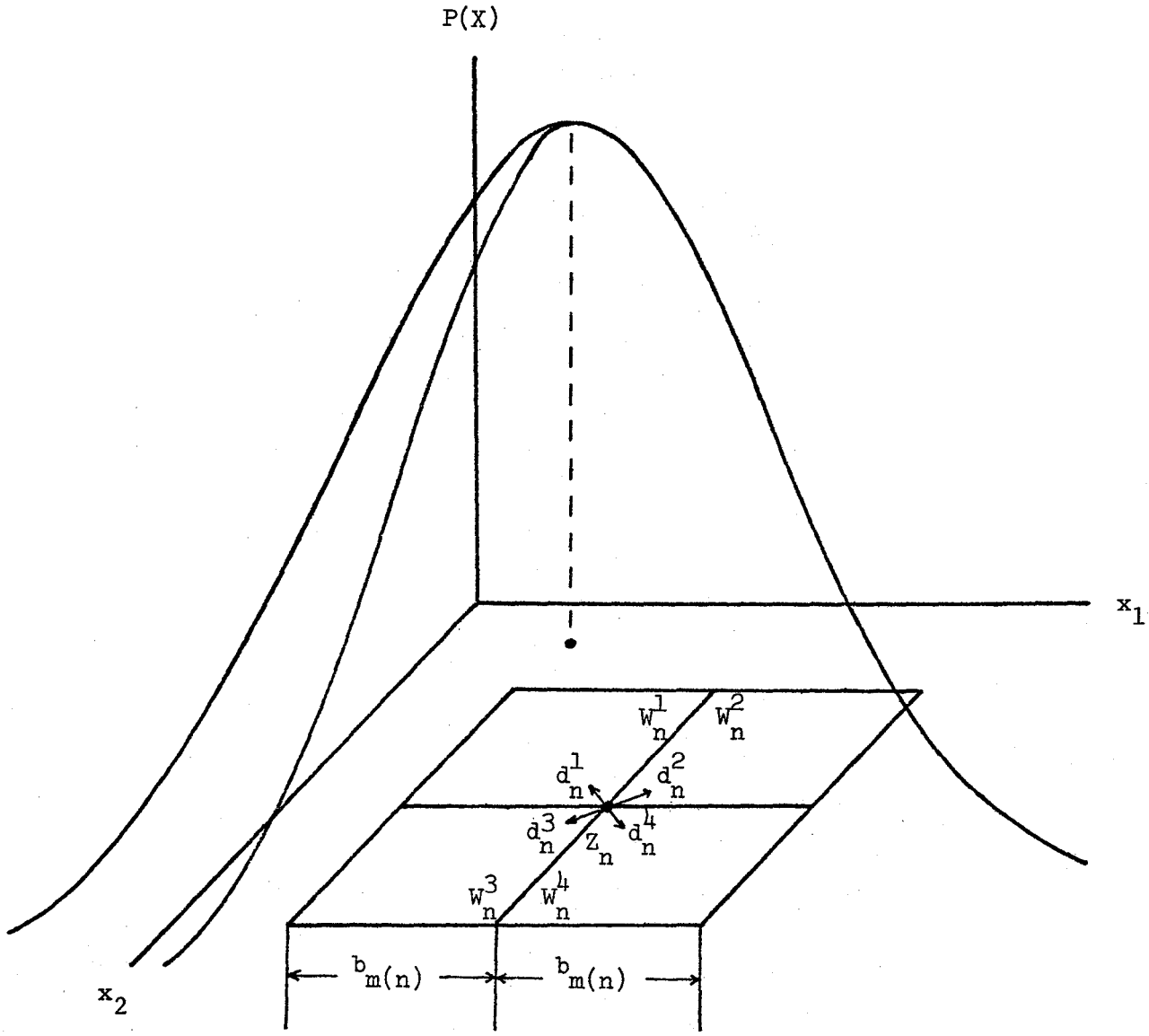


Fig. 6.1 Two-dimensional window function.

the window, no change of the window is made. At every learning step, $a_{m(n+1)}\zeta_{n+1}(Z_n)$ is added to Z_n, so that the estimator moves in the direction d_n^1, d_n^2, d_n^3, or d_n^4, according as the n+1th input pattern is observed within the region W_n^1, W_n^2, W_n^3, or W_n^4. Clearly, when no input pattern is observed within the window, Z_n does not move in any direction. In the above example, the region W_n^1 is nearer the mode than the other regions are, so that the probability of an input pattern being observed within W_n^1 is higher than within the other regions. Therefore, the mode estimator moves probabilistically in the direction d_n^1 and approaches one of the modes of the pdf with a reducing window. The convergence theorem is presented below.

6.2.3 CONVERGENCE THEOREM OF THE MODE ESTIMATION ALGORITHM

Theorem 6.1: Assume that the following conditions are satisfied:

1) X_n is the n-th random sample vector from an unknown L-dimensional continuous pdf p(X) which has N maxima at $X = {}^iZ$ ($\|{}^iZ\| < \infty$, i=1,2,...,N) and has a finite number of singular points.

2) $\lim_{k\to\infty} a_k = \lim_{k\to\infty} b_k = 0 \qquad (a_k,\ b_k > 0)$.

3) $\sum_{k=1}^{\infty} a_k b_k^{-1} = \infty$.

4) $\sum_{k=1}^{\infty} (a_k b_k^{-2})^2 < \infty$.

Then, Z_n defined in (6.1)-(6.4) converges to one of the maximum points of p(X) with probability one. That is,

$$P[\ \min_j \{\ \lim_{n\to\infty} \|Z_n - {}^jZ\|\ \} = 0\ ] = 1, \qquad \text{for some } j.$$

Proof: Proof of this theorem is divided into four parts. First, we begin the proof by showing

$$P[\ m(n) \xrightarrow[n \to \infty]{} \infty\ ] = 1.$$

Second, we show

$$\forall i\ (i=1,2,\ldots,N), \qquad P[\ \lim_{m(n) \to \infty} \| Z_{m(n)} - {}^{i}Z \| = \infty] = 0.$$

Third, we show

$$\exists\, {}^{i}\theta \geq 0\ (i=1,2,\ldots,N), \quad P[\ \lim_{m(n) \to \infty} \| Z_{m(n)} - {}^{i}Z \| = {}^{i}\theta] = 1.$$

And finally, by showing

$$P[\ \min_{j}\ {}^{j}\theta = 0\ ] = 1$$

the proof is completed. Convergence property of semi-martingales plays a prominent role in the detailed proof, which is presented in Appendix 6.1.

6.3 CONSTRUCTION OF A DISCRIMINANT FUNCTION

If our purpose were to estimate one of the modes of a pdf, then the proposed algorithm would perform the estimation successfully regardless of whether the initial estimate is far from the mode or not. As is discussed in Section 6.1, however, it is necessary for the design of a discriminant function to estimate all modes of the mixture pdf of input patterns. Unfortunately, the mode estimation algorithm that we

have discussed thus far is designed to seek one of the modes of a multimodal pdf, so that it does not assure us of success on the estimation of all the modes. We shall describe a procedure for estimating all the modes below. However, because of a difficulty in determining appropriate initial estimates, it does not guarantee that all the modes can be always detected, as will be discussed later.

Before executing the algorithm, a rough estimate of the range of the pattern distribution is obtained by observing input patterns. Then, dividing the range into $\eta N^{\#}$ parts, we set $\eta N^{\#}$ initial estimates at the center of each subrange which is the hyper-cubic window, where $N^{\#}$ is the upper bound of the number of categories.* (Usually, η is so chosen that $\eta N^{\#}$ is much larger than the number of the modes.) After determining the initial estimates the mode estimation algorithm is performed. In a sufficiently large number of learning iterations, the $\eta N^{\#}$ estimators are expected to form N (actual number of the modes) clusters, since each estimator converges to one of the modes. Therefore, the mean vector of each cluster formed by the estimators can be the estimates of the modes.

As described before, there is no guarantee that the above procedure always succeeds on estimating all the modes. That is, $\eta N^{\#}$ estimators may not reduce in number to approach the actual number of the modes N. For example, in the case of the initial estimates shown in

---

* Since we earlier assumed that there is a one-to-one correspondence between categories and modes, $N^{\#}$ can be the upper bound of the number of the modes.

Fig. 6.2, it may occur that ${}^1Z_n$, ${}^2Z_n$, ${}^4Z_n \longrightarrow {}^1Z$, ${}^3Z_n$, ${}^5Z_n$, ${}^6Z_n \longrightarrow {}^2Z$, ${}^7Z_n$, ${}^8Z_n$, ${}^9Z_n \longrightarrow {}^4Z$, and ${}^3Z$ may not be detected. In most cases, however, we can avoid such a difficulty by setting the range and η sufficiently large. In the above example, it is seen that all the modes can be detected with larger range and η. Generally, in nonsupervised algorithms, whether the global optimum is found or not depends on the initial estimates. Such a difficulty is not peculiar to our algorithm but common to other nonsupervised algorithms particularly based on minimization of criterion functions.

Thus, we have the following discriminant rule:

$$\text{decide: } X \in C^i \quad \text{if } \|{}^iZ_n - X\| = \min_j \|{}^jZ_n - X\| \tag{6.5}$$

where ${}^iZ_n$ is the n-th estimate of the mode ${}^iZ$. It is easily seen that this discriminant rule is constructed based on minimum-distance classifier [62] by using only the knowledge of the locations of the modes of the mixture pdf. Our discriminant rule works well especially when input patterns of each category cluster globularly. In order to obtain an efficient classifier of the performance free from the structure of the mixture distribution, the mixing coefficient, and the covariance matrix of each cluster need to be estimated in addition to the modes. The problem of getting such additional information is considered in the next section.

6.4 TWO-CATEGORY UNIMODAL CLASS DENSITY PROBLEM

A nonparametric signal detection problem is discussed here in order to demonstrate the efficiency of our mode estimation algorithm.

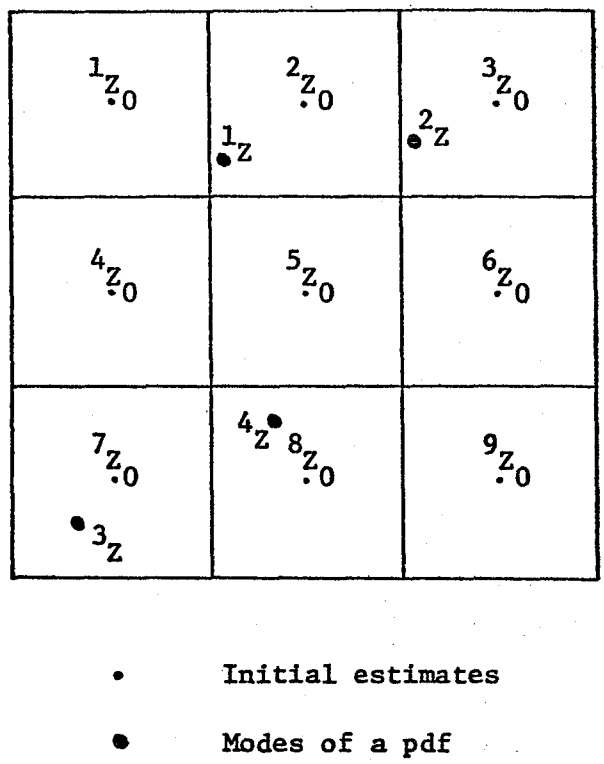


Fig. 6.2 An example of initial estimates where some mode may not be detected.

In the last section we have obtained a multicategory classifier based on the knowledge of the modes' locations, where the structure of each cluster and mixing coefficient have not been taken into account. We shall show that a signal detector with the ability to estimate such parameters as well as the input signal can be constructed without supervision.

Let P, μ, ν, and Σ be the probability of signal occurrence, the signal vector, the mean vector of noise, and the covariance matrix of noise, respectively. Assume that the pdf of noise is unimodal and symmetric with respect to its mean vector ν. Under this assumption, the mixture pdf of input patterns has two modes and each of them corresponds to either μ or ν. These two mean vectors can be estimated by using our mode estimation algorithm. Also, P and Σ can be obtained by using the estimates of μ and ν. Therefore, an adaptive signal detector which can estimate all the unknown parameters is constructed in the following manner.

Let M be the mean vector of the mixture distribution of input patterns, then we have

$$M = P\mu + (1 - P)\nu. \tag{6.6}$$

Let μ_n and ν_n be the n-th estimates of μ and ν, respectively, and X_i be the i-th input pattern. From (6.6) the n-th estimate of P, P_n is obtained as follows:

$$P_n = U[\ 1/L \sum_{d=1}^{L} (M_n^d - \nu_n^d) / (\mu_n^d - \nu_n^d)\ ]$$

where

$$U[\xi] = \begin{cases} 0.5, & \text{if } 1 < \xi \\ \xi, & \text{if } 0 \leq \xi \leq 1 \\ 0.5, & \text{if } \xi < 0 \end{cases}$$

$$M_n = 1/n \sum_{i=1}^{n} X_i$$

and the superscript d indicates the d-th component of each vector.

Now, we have the covariance matrix S of the mixture distribution of input patterns as follows:

$$S = E[(X - M)(X - M)^T]$$

$$= \Sigma + P(1 - P)(\mu - \nu)(\mu - \nu)^T \qquad (6.7)$$

From (6.7) the n-th estimate of Σ, Σ_n is obtained as follows:

$$\Sigma_n = S_n - P_n(1 - P_n)(\mu_n - \nu_n)(\mu_n - \nu_n)^T \qquad (6.8)$$

where

$$S_n = 1/(n-1) \sum_{i=1}^{n} (X_i - M_n)(X_i - M_n)^T.$$

It is easily seen that P_n and Σ_n converge to P and Σ, respectively, if and only if μ_n and ν_n converge to μ and ν, respectively. Hence, for a coefficient vector of the linear discriminant function $W^TX = \theta$ we can

use W_n defined in the following:

$$W_n \equiv \Sigma_n^{-1}(\mu_n - \nu_n)/\| \Sigma_n^{-1} (\mu_n - \nu_n) \| \tag{6.9}$$

where the structure of each cluster is taken into account.

Let us consider a two-dimensional case as an example (See Fig. 6.3). When the mean vectors are estimated, we usually construct a discriminant function of the form (d.f.1). However, when both the mean vectors and the covariance matrix are estimated, we can use (d.f.2) which is obtained by rotating (d.f.1) appropriately. It is obvious from Fig. 6.3 that (d.f.2) is superior to (d.f.1).

The remainder of this section deals with a decision algorithm of the threshold value θ. As is discussed in Chapter 5, the neck between the two clusters shown in Fig. 6.4 seems to be reasonable threshold, and it corresponds to the minimum point of the univariate pdf produced by projecting the original pattern space to the direction of W. Therefore, we use the minimum point of the pdf as the threshold value θ, which can be estimated by our mode estimation algorithm.

The n+1th estimate of the threshold value θ, θ_{n+1} is obtained in the following manner:

$$\theta_{n+1} = \theta_n - a_{m(n+1)} \zeta_{n+1}(\theta_n) \tag{6.10}$$

$$\theta_{n+1} = \begin{cases} \theta_{n+1}, & \text{if } W_{n+1}^T \nu_{n+1} \leq \theta_{n+1} \leq W_{n+1}^T \mu_{n+1} \\ W_{n+1}^T M_{n+1}, & \text{otherwise} \end{cases} \tag{6.11}$$

where

$$M_{n+1} = 1/(n+1) \sum_{i=1}^{n+1} X_i$$

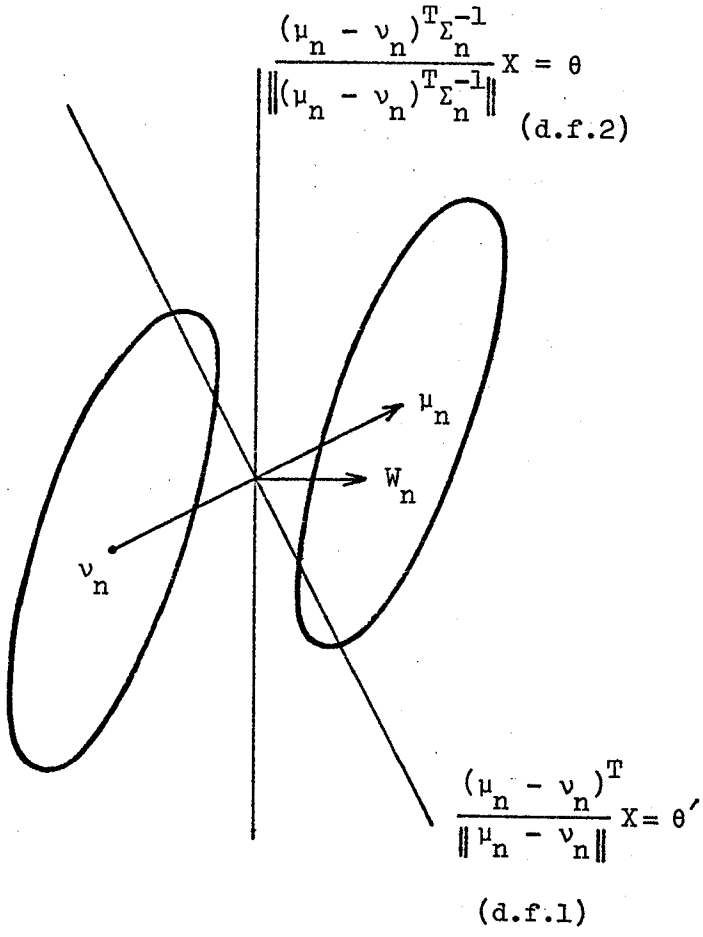


Fig. 6.3 Improvement of a linear discriminant function.

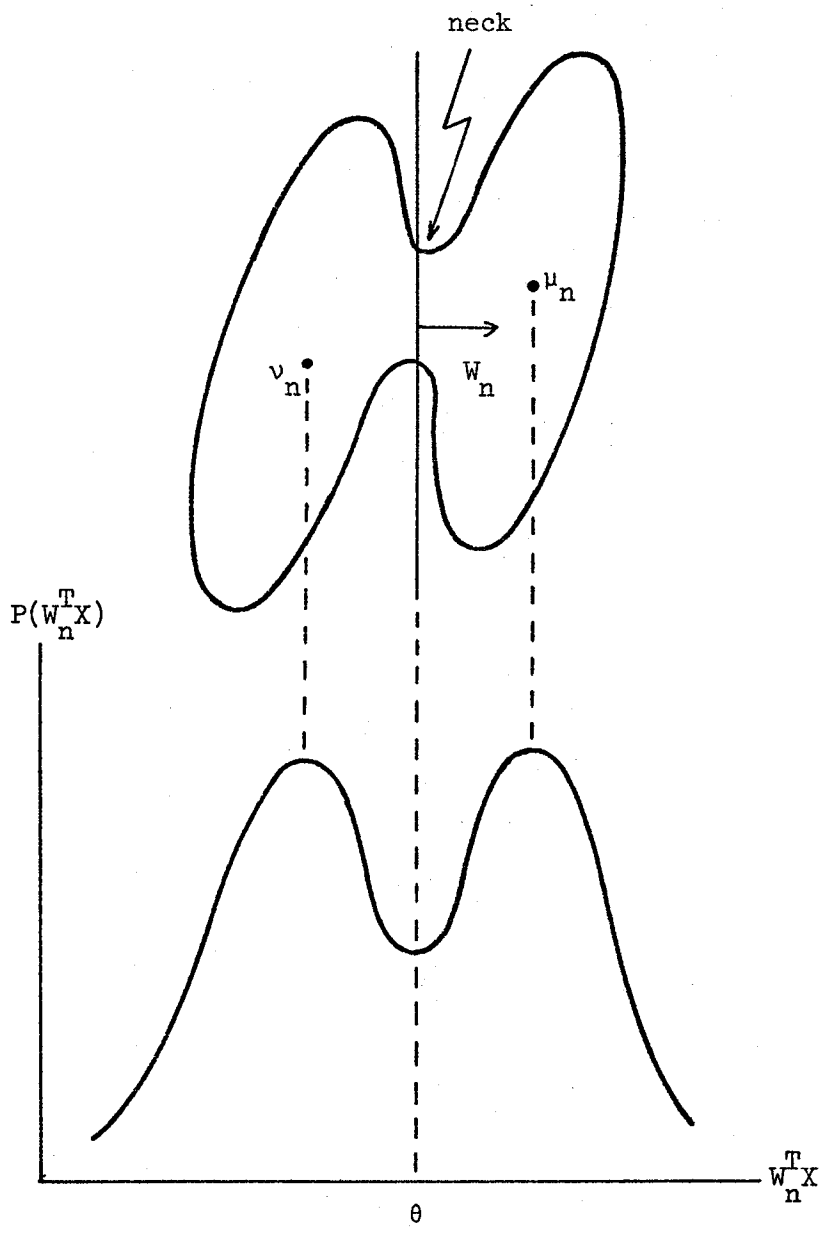


Fig. 6.4 Neck and threshold value.

$$\zeta_{n+1}(\theta_n) = \begin{cases} b_{m(n)}^{-2}, & \text{if } \xi_{n+1} = 1 \text{ and } \theta_n \le W_{n+1}^T X_{n+1} \\ 0, & \text{if } \xi_{n+1} = 0 \\ -b_{m(n)}^{-2}, & \text{if } \xi_{n+1} = 1 \text{ and } W_{n+1}^T X_{n+1} < \theta_n \end{cases}$$

$$\xi_{n+1} = \begin{cases} 1, & \text{if } |W_{n+1}^T X_{n+1} - \theta_n| \le b_{m(n)} \\ 0, & \text{otherwise} \end{cases}$$

and $a_{m(n)}$, $b_{m(n)}$, and $m(n)$ are defined in the same way as in Theorem 6.1.

Hence, from (6.9)-(6.11) we have the following discriminant rule:

$$\text{decide: } X \in \begin{cases} C_\mu, & \text{if } W_n^T X \ge \theta_n \\ C_\nu, & \text{otherwise} \end{cases}$$

where C_μ and C_ν indicate the categories corresponding to μ and ν, respectively. The above classifier shows almost optimal behavior because its decision is made by using the information about the probability of each category's occurrence, the mean vectors, and the covariance matrix. This fact is verified by computer simulation in the next section.

6.5 COMPUTER EXPERIMENTS

In this section, some results of a computer study of our learning algorithms are presented.

6.5.1 MODE ESTIMATION

In the computer study, modes of a two-dimensional normal mixture distribution $F(X) = \sum_{i=1}^{3} P \cdot N(\mu_i, \Sigma)$ are estimated, where $P = 1/3$, $\mu_1^T = (0, 2)$, $\mu_2^T = (2, -2)$, $\mu_3^T = (-2, -2)$, $\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. The mode estimation algorihtm is implemented in the case where $N^{\#} = 4$, $\eta = 9/4$, $a_k = 2/k$, and $b_k = 2/k^{0.2}$, so that nine initial estimates are determined as shown in Fig. 6.5. This figure also shows the loci of the mode estimators for 600 input patterns using the above parameters. It is seen that the mode estimators converge to the points corresponding to the modes in a large number of learning iterations and that in this case three clusters (^1Z_n & ^2Z_n & ^5Z_n, ^6Z_n & ^9Z_n, ^4Z_n & ^7Z_n & ^8Z_n) appear. Note that ^3Z_n cannot be a mode because it still stays at the initial point. We conclude, therefore, that the mixture pdf has three modes in all and each mode is located at the center of each cluster formed by the estimators.

6.5.2 SIGNAL DETECTION

Computer simulation of the signal detector discussed as an application of our mode estimation algorithm is made and some of the results are presented below. In the experiment, two-dimensional Gaussian noise is used, where $P = 0.7$, $\mu^T = (2, 2)$, $\nu^T = (0, 0)$, and $\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$. Convergence processes of P_n and W_n are shown in Fig. 6.6. Fig. 6.7 shows the learning processes of some signal detectors, where marks $\triangle$, $\square$, and $\circ$ indicate the probability of error of the discriminant functions

$$(\mu_n - \nu_n)^T X = (\mu_n - \nu_n)^T M_n \qquad \text{(d1)}$$

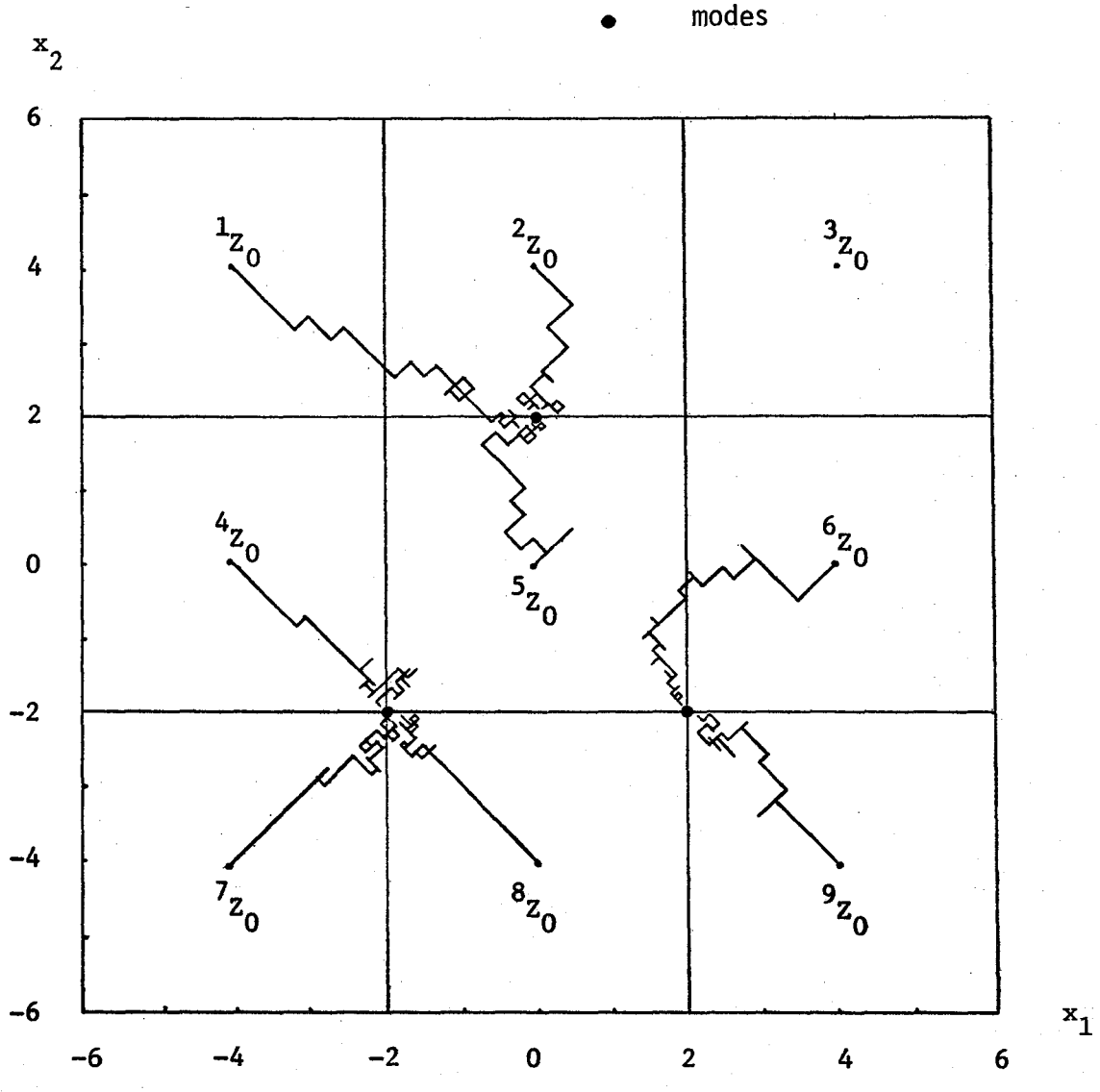


Fig. 6.5 Loci of the mode estimators.

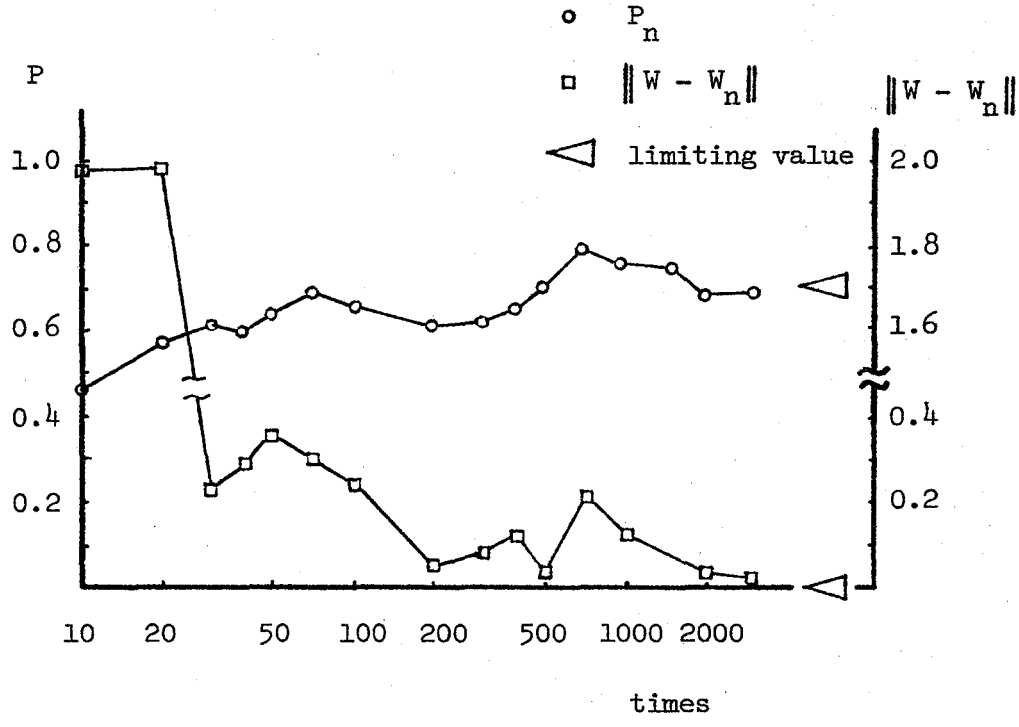


Fig. 6.6 Learning processes of P_n and W_n.

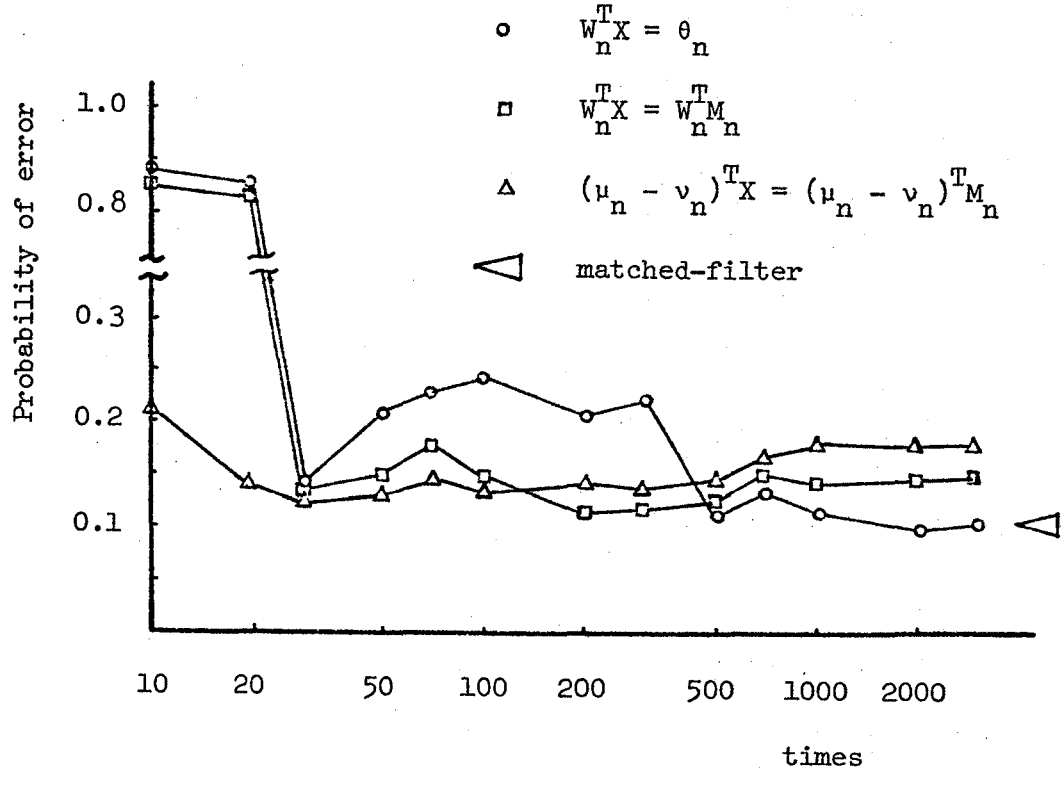


Fig. 6.7 Learning processes of some linear discriminant functions.

$$W_n^T X = W_n^T M_n \qquad \text{(d2)}$$

$$W_n^T X = \theta_n, \qquad \text{(d3)}$$

respectively. The weight coefficient of (d1) is obtained by using the knowledge of the modes' locations alone and the threshold value is determined by using the mean vector of the mixture distribution of input patterns. On the other hand, the weight vector of (d2) is obtained by using the knowledge of the covariance matrix in addition to that of the modes' locations. The discriminant function (d3) is that proposed in Section 6.4. From Fig. 6.7 it is seen that error rates of these discriminant functions become lower in order of (d1), (d2), and (d3), and that the probability of error of (d3) converges to that of the optimal machine indicated by arrows. These results compare favorably with others reported previously [12],[65],[75],[76],[83].

6.6 CONCLUSION

In this chapter we have discussed a nonparametric learning scheme, without a teacher, based on mode estimation. A new hyper-cubic window function has been introduced, which is useful for estimating the gradient of a pdf. By using the hyper-cubic window function, an algorithm for seeking one of the modes of a mixture pdf was proposed and its convergence proof was also presented. A minimum-distance classifier for the multi-category problem was constructed based on the estimated modes of the mixture pdf of input patterns. Furthermore, some discussions were made on a nonparametric signal detection problem as an application of our mode estimation algorithm. We also obtained an

adaptive signal detector which nearly converges to the optimal machine without supervision. In order to verify our algorithms some computer simulation of their learning processes was made, and satisfactory results were obtained.

This chapter has treated the problem of nonsupervised nonparametric learning without memorizing input patterns. We have designed a minimum-distance classifier for the multi-category problem. However, it does not have a satisfactory structure because of a complete lack of the information available during the learning period.

In the next chapter, the same problem as that discussed here is studied by memorizing all sample patterns. It will be revealed that memorizing patterns makes it possible to realize almost complete pattern classification.

APPENDIX 6.1 PROOF OF *Theorem 6.1*

Proof: First we shall show

$$P[\ m(n) \xrightarrow[n \to \infty]{} \infty\ ] = 1.$$

Assume that

$$\exists\, n_0,\ m_0, \qquad P[\ n > n_0 \Longrightarrow m(n) = m_0\ ] > 0,$$

then there exists a positive constant C such that

$$b_{m(n)} = C, \qquad \text{for any } n > n_0.$$

Therefore, the volume of the window converges to C^L. However, the probability that after the convergence, at least one input pattern is observed within the window of the positive volume C^L is one. From (6.2) and (6.3) this contradicts the assumption. Hence, we have

$$P[\ m(n) \xrightarrow[n \to \infty]{} \infty\ ] = 1. \tag{6.12}$$

Since a mode estimator and a window are not changed if input patterns are not observed within the window, we can neglect such input patterns in the following analyses. That is, we renumber the estimator Z_n and the window function ζ_n as follows:

$$Z_{m(n)} = Z_n$$

$$\zeta_{m(n)} = \zeta_n.$$

Next, we shall show

$$\forall i \ (i=1,2,\ldots,N), \qquad P[\lim_{m\to\infty} \|Z_m - {}^iZ\| = \infty] = 0. \qquad (6.13)^*$$

Suppose

$$\exists i, \qquad P[\lim_{m\to\infty} \|Z_m - {}^iZ\| = \infty] > 0.$$

From the constraint 4), there exists a random sequence $\{Z_m\}$ such that

$$\exists d, \qquad P[\lim_{m\to\infty} Z_m^d = \infty] > 0 \qquad (6.14)$$

or

$$\exists d, \qquad P[\lim_{m\to\infty} Z_m^d = -\infty] > 0. \qquad (6.15)$$

If $P[\lim_{m\to\infty} Z_m^d = \infty] > 0$, then from (6.1) and the constraint 1) there exists a random sequence $\{Z_m\}$ with positive probability such that

$$\forall C > 0, \ \exists M_1 > 0, \ m > M_1 \Longrightarrow Z_m^d > C \text{ and } E[Z_{m+1}^d | Z_m] < Z_m^d.$$

Therefore, Z_m^d is a semi-martingale [15]. From the convergence property of semi-martingales, the sequence $\{Z_m^d\}$ converges almost everywhere where (6.14) is true. This contradicts (6.14). Thus, $P[\lim_{m\to\infty} Z_m^d = \infty] = 0$.

---

* For simplicity, the notation m will be used instead of m(n).

Similarly, $P[\lim_{m\to\infty} Z_m^d = -\infty\ ] = 0$. Hence, we have

$$\forall i, \qquad P[\lim_{m\to\infty} \|Z_m - {}^iZ\| = \infty\ ] = 0. \tag{6.16}$$

Next, we shall show the existence of N nonnegative values ${}^i\theta$ $(i=1,2,\ldots,N)$ such that

$$P[\lim_{m\to\infty} \|Z_m - {}^iZ\| = {}^i\theta\ ] = 1. \tag{6.17}$$

Suppose that (6.17) is not true. Then, from (6.16) there exists a random sequence $\{Z_m\}$ with positive probability such that

$$\exists i,\ \beta > \alpha > 0,\ \lim_{M\to\infty} \inf_{m\geq M} \|Z_m - {}^iZ\| < \alpha < \beta$$

$$< \lim_{M\to\infty} \sup_{m\geq M} \|Z_m - {}^iZ\|. \tag{6.18}$$

From the constraints 1), 2), and 4), we can find a subsequence $\{Z_{m_j}\}$ of the sequence $\{Z_m\}$ such that

$$\left[\begin{array}{l} \sum_{j=1}^{\infty} a_{m_j}\, \zeta_{m_j}^d(Z_{m_j-1}) = \infty \qquad (6.19) \\ \\ E[\zeta_{m_j}^d(Z_{m_j-1}) \,|\, Z_{m_j-1}] < 0 \qquad (6.20) \end{array}\right.$$

or

$$\left[\begin{array}{l} \sum_{j=1}^{\infty} a_{m_j}\zeta_{m_j}^{d}(Z_{m_j-1}) = -\infty \qquad (6.21) \\ \\ E[\zeta_{m_j}^{d}(Z_{m_j-1})\,|\,Z_{m_j-1}] > 0. \qquad (6.22) \end{array}\right.$$

However, in both cases where (6.20) is true and where (6.22) is true, the sequence $\{\sum_{j=1}^{k} a_{m_j}\zeta_{m_j}^{d}(Z_{m_j-1})\}$ turns out to be a semi-martingale. Therefore, it converges almost everywhere where (6.18) is true. This contradicts (6.19) and (6.21). Hence, we obtain

$$\exists {}^{i}\theta \geq 0, \qquad P[\lim_{m\to\infty} \|Z_m - {}^{i}Z\| = {}^{i}\theta\ ] = 1. \qquad (6.23)$$

Finally, we shall show that

$$P[\min_{j} {}^{j}\theta = 0\ ] = 1.$$

Suppose

$$P[\min_{j} {}^{j}\theta > 0\ ] > 0.$$

From the proof of (6.23), it is obvious that Z_m converges with probability one. Accordingly, there exists a sequence $\{Z_m\}$ with positive probability such that

$$\exists Q\ (\neq {}^{i}Z\ (i=1,2,\ldots,N)),\quad Z_m \xrightarrow[m\to\infty]{} Q. \qquad (6.24)$$

Then, for arbitrary d (d=1,2,...,L) we have

$$\forall \varepsilon > 0, \quad \exists M_2, \quad p > q > M_2 \Longrightarrow \left| \sum_{m=q}^{p} a_m \zeta_m^d(Z_{m-1}) \right| < \varepsilon. \tag{6.25}$$

We here assume that Q is a regular point of p(X), that is, there exists at least one d such that

$$\exists \delta, \ \gamma > 0, \qquad \gamma = \inf_{\|Q-X\| < \delta} \frac{\partial p(X)}{\partial X^d}.$$

For such d, we have either of the following two cases:

$$\exists M_3 (> M_2), \quad m > M_3 \Longrightarrow E[\zeta_m^d(Z_{m-1}) | Z_{m-1}] > 0 \tag{6.26}$$

$$\exists M_4 (> M_2), \quad m > M_4 \Longrightarrow E[\zeta_m^d(Z_{m-1}) | Z_{m-1}] < 0. \tag{6.27}$$

In the case of (6.26), we have from the constraint 3)

$$E\left[\sum_{m=M_3+1}^{\infty} a_m \zeta_m^d(Z_{m-1}) \right] \geq \sum_{m=M_3+1}^{\infty} \gamma a_m b_m^{-1} = \infty \tag{6.28}$$

where the expectation is calculated over all the sequences of the observed input patterns which make $\{Z_m\}$ satisfy (6.26). Considering the constraint 4), (6.28) contradicts (6.25). Similarly, in the case of (6.27), a contradiction of (6.25) can be derived. Therefore, Q is not a regular point, and hence Q is either a saddle point or a minimum point of p(X) because Q is not a mode. Accordingly, there exists at least one component Q^d corresponding to a minimum point of a univariate pdf. It is obvious, however, that the probability of Z_m^d converging to

one of the minimum points of a pdf is zero. Thus, we have

$$P[\min_j {}^j\theta = 0] = 1. \tag{6.29}$$

Hence, from (6.12),(6.23), and (6.29) the proof is completed.

(Q.E.D.)

CHAPTER 7

A CLUSTER DETECTION ALGORITHM BASED ON HIERARCHICAL STRUCTURE

7.1 INTRODUCTION

The main purpose of the cluster detection problem is to develope efficient algorithms for partitioning a given data set into a finite number of subsets which can be considered as reasonable clusters, where no *a priori* knowledge as to the data is assumed. Cluster detection is essentially nonsupervised nonparametric learning in pattern recognition. However, it has been studied enthusiastically not only in pattern recognition but in biological and social scienses [3],[4],[14], [20]-[23], [27],[28],[34],[35],[41],[43]-[45],[67],[71],[73],[77],[78], [89],[98]. A cluster is loosely defined as a collection of data which are similar to each other, though its rigorous definition is not established yet. Therefore, various approaches to the problem have been discussed. The approaches can be divided into the following six major groups:

1) The approach using centers of clusters such as modes of the pattern distribution [4],[22],[23].
2) The approach based on minimization of appropriate criteria [21],[41].
3) The approach using graph-theoretical methods [3],[98].
4) The approach based on hierarchical ordering of data [27],[28],[35], [89].
5) The approach using nonlinear mapping [43],[44],[67],[73].
6) The approach based on appropriate similarity measures [34].

Some typical examples of 2-dimensional clusters that are easy for man to detect (Zahn [98]) are shown in Fig. 7.1. We now examine some of the above algorithms using the clusters in Fig. 7.1. Zahn's algorithm [98] has difficulties in detecting such clusters as shown in (c), (e), and (f), though it makes it possible to detect many types of clusters. Jarvis and Patrick's algorithm [34] seems to be unable to detect the clusters in (e). On the other hand, algorithms of Koontz and Fukunaga [41] and Gitman [23] are capable of detecting the clusters in (e), but they cannot detect ones in (b). Thus, such an algorithm that is able to detect all the clusters in Fig. 7.1 is not known yet.

It is difficult for man to consider multidimensional clusters directly, since they are invisible. Therefore, after investigating 2-dimensional clusters thoroughly multidimensional clusters are studied by analogy. We here note that the concept of clusters is vague and variable, that is, different users may require of the algorithm that it detect different types of clusters from the same data set. Considering this fact, we intend in this chapter to construct such a cluster detection algorithm that has a flexible structure so as to meet various requirements and that can detect all the clusters in Fig. 7.1 at the same time. In the following sections, analyses are made by using the relative values, e.g. the dissimilarities, between data instead of their absolute locations in order to broaden the scope of data to which the algorithm is applicable.

7.2 DEFINITIONS

Definition 7.1: Let $\Omega_k(i)$ be a set of k-nearest neighbors (k-NN's) of the point i, where k-NN's are the first k points of a sequence of points

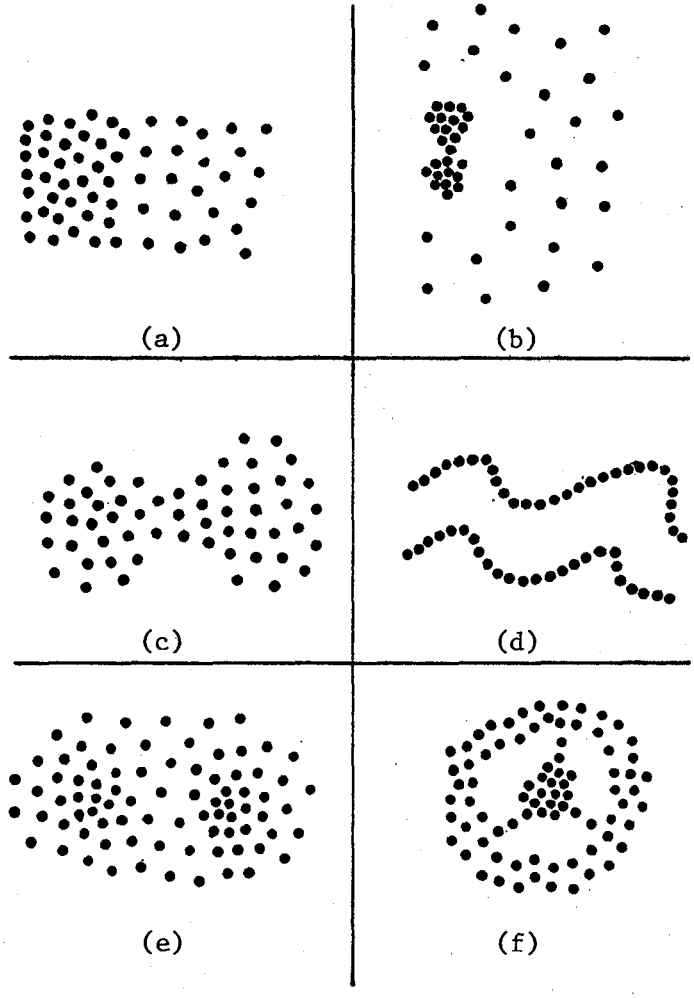


Fig. 7.1 Typical examples of two-dimensional clusters.

arranged in order of $1/d(i,j)$ $(j=1,2,\ldots,N)$, where N is the number of data and $d(i,j)$ is the dissimilarity between the points i and j $(d(i,i) = 0)$.

Definition 7.2: The potential $P_k(i)$ of the point i is defined as

$$P_k(i) \equiv 1/k \sum_{j \in \Omega_k(i)} d(i,j).$$

Definition 7.3: A weakly connected digraph is said to be a hierarchy if it has neither cycle nor loop.

Definition 7.4: A hierarchy is said to be a subcluster if each vertex is subordinate to at most one point.

Definition 7.5: A central point i_m of the subcluster m is the unique point subordinate to no point.

Definition 7.6: W_m is a set of points contained in the subcluster m.

Definition 7.7: Two points i and j are said to be k-adjacent to each other if $i \in \Omega_k(j)$ and $j \in \Omega_k(i)$. Let $S_k(i)$ be a set of points k-adjacent to the point i.

Definition 7.8: The potential $P_k^{sc}(m)$ of the subcluster m is defined as

$$P_k^{sc}(m) \equiv 1/\sigma[\Omega_k(i_m)] \sum_{i \in \Omega_k(i_m)} P_k(i)$$

where $\sigma(A)$ denotes the number of elements of the set A.

Definition 7.9: The k-boundary point set $Y_k^{m,n}$ of the subcluster m to the subcluster n is defined as

$$Y_k^{m,n} \equiv \{i \mid i \in W_m \text{ and } S_k(i) \cap W_n \neq \phi\}.$$

Definition 7.10: Two subclusters m and n are said to be $(\xi,\eta)_k$-adjacent to each other where

$$\xi \equiv \sigma[Y_k^{m,n}] \qquad \text{and} \qquad \eta \equiv \sigma[Y_k^{n,m}].$$

The subclusters m and n are said to be in k-touch with each other if $\xi \neq 0$ or $\eta \neq 0$. Otherwise, they are said not to be in k-touch with each other.

7.3 CLUSTER DETECTION ALGORITHM

In this section, detailed discussion is made concerning our cluster detection algorithm after presenting it. At a first glance the following algorithm may seem to be rather complicated because of the presence of four parameters at Step 1. However, the parameter δ is usually set at infinity and the parameters α, β, and γ are fixed when users define what a cluster is, so that k is the only parameter that varies with the runs, which will be considered later again.

7.3.1 ALGORITHM

Step 1: Set k, α, β, γ, and δ.

Step 2: Calculate $\Omega_k(i)$, $P_k(i)$, and $S_k(i)$ for every i.

Step 3: Subordinate every point i to the point j such that

$$P_k(j) = \min_{m \in S_k(i)} P_k(m).$$

If i = j, then the point i is subordinate to no point.

Step 4: Detect all central points.

Step 5: Assign every point i to the subcluster of the central point reachable from it.

Step 6: Take a new unordered pair (m, n) of subclusters.* If a new one can be taken successfully, then continue to Step 7. Otherwise, go to Step 12.

Step 7: If the pair of subclusters (m, n) is in 5-touch with each other, then continue to Step 8. Otherwise, go to Step 6.

Step 8: If

$$\max[\, P_k^{sc}(m),\ P_k^{sc}(n)\ ] > \alpha \cdot \min[\, P_k^{sc}(m),\ P_k^{sc}(n)\ ],$$

then continue to Step 9. Otherwise, go to Step 10.

Step 9: Reassign all the points of $Y_k^{m,n}$ to the subcluster n where $P_k^{sc}(m) < P_k^{sc}(n)$, and go to Step 6.

Step 10: If

$$\min[\, \sigma[W_m]/\sigma[Y_5^{m,n}],\ \sigma[W_n]/\sigma[Y_5^{n,m}]] > \beta \text{ and}$$

$$1/\sigma[X_5^{m,n}] \sum_{i \in X_5^{m,n}} P_k(i) < \delta \cdot \max[\, P_k^{sc}(m),\ P_k^{sc}(n)\ ],$$

then go to Step 6. Otherwise, continue to Step 11, where $X_5^{m,n} = Y_5^{m,n} \cup Y_5^{n,m}$.

Step 11: If

$$1/\sigma[X_5^{m,n}] \sum_{i \in X_5^{m,n}} P_k(i) > \gamma \cdot \max[\, P_k^{sc}(m),\ P_k^{sc}(n)\ ],$$

then go to Step 6. Otherwise, assign the two subclusters m and n to the same cluster, then go to Step 6.

---

* M(M-1)/2 subclusters are taken in all where M is total number of the subclusters. Different results may be obtained according to the order in which they are taken when some subclusters are changed in Step 9. However, this is neglected in our algorithm because the difference is very small.

Step 12: Construct clusters by collecting the subclusters assigned to the same cluster and terminate.

7.3.2 POTENTIAL AND HIERARCHICAL STRUCTURE

To begin with, the potential playing a prominent role in constructing subclusters is considered. In Definition 7.2 the potential of every point is defined as the mean value of the dissimilarities between the point and its k-NN's. One can see that the potential is a kind of measures of point density when Euclid distance is used as dissimilarity. Usually, local point density is measured by using the number of the points in a distance determined beforehand [23],[41]. However, no reasonable method of determining an appropriate distance is known. Moreover, in the case where two clusters having a great difference in density exist (See Fig. 7.2), every point density of the dense cluster is 20 and that of the sparse one is one. Thus, the usual measure cannot represent the difference in point density in the dence cluster. In our potential, on the other hand, the concept of k-NN is employed where the number of the neighbors k is fixed instead of the distance. The potential is obtained by making use of the characteristic of the k-NN that the radius of it varies automatically according to the point density (it is small when the point density is high and is large when it is low). Table 7.1 lists some examples of the potentials of some points in Fig. 7.2. One can see from the table that our potential is an efficient measure of local point density.

Next, let us examine the hierarchical structure introduced in Step 5. Its introduction is made by subordinating each point i to the point of the lowest potential in $S_k(i)$. Thus, the constructed

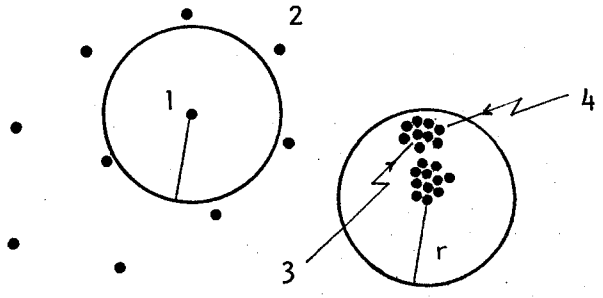


Fig. 7.2 Point density and its measure.

Table 7.1 Some examples of potentials.

$P_5(1)$	$P_5(2)$	$P_5(3)$	$P_5(4)$
1.35	1.81	0.16	0.21

hierarchies are subclusters. There is no point belonging to two subclusters or more, so that every point except the central point is subordinate to exactly one point. Therefore, all subclusters turn out to be a partition of the given data set. As is seen from Step 5, each subcluster is constructed by regarding the point of minimum potential (the point of maximum point density) as its center. It is also seen that every point density is detected quite easily by using the hierarchical structure.

We here note that our subclusters partition the given data set very sensitively to a change of point density. Let us take Fig. 7.3 as an example. Fig. 7.3 (b) depicts some subclusters constructed from the point set shown in Fig. 7.3 (a) for k=4, where seven subclusters appear corresponding to seven points of minimum potential. Four subclusters and two subclusters are also obtained according as k=6 and k=8, respectively (See Fig. 7.3 (c) and (d)). Moreover, the whole set becomes one subcluster for k as large as the number of the points N, since, for such k, every point is subordinate to the point of the global minimum potential instead of the point of the local minimum potential. Thus, the following relation holds for k not so large as N:

Total number of subclusters $\geq$ Total number of clusters.

It is seen from the above discussion that no subcluster contain such points that ought to be classified into two subclusters or more for appropriate k, though the point set may be partitioned into more subclusters than clusters. Therefore, it is sufficient for obtaining reasonable clusters to consider some merging operations of subclusters, which are discussed in the next section.

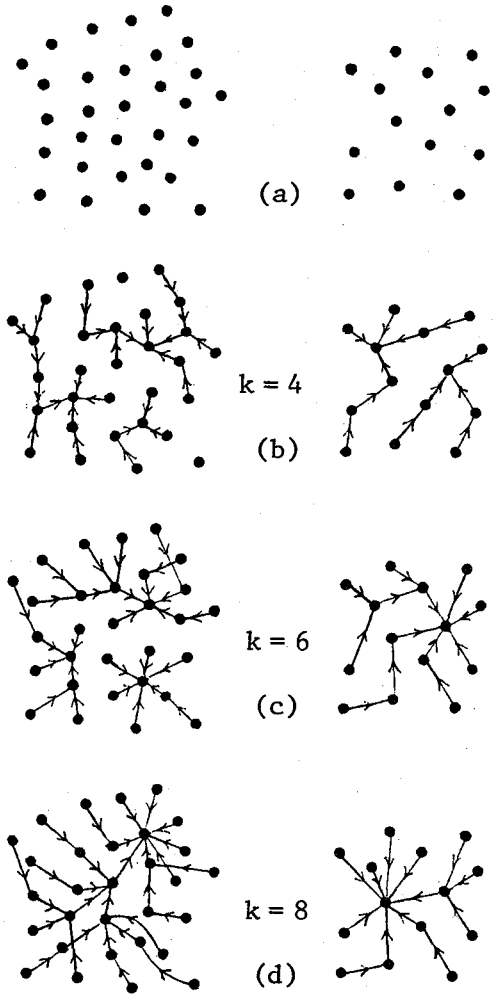


Fig. 7.3 Hierarchical structure and subclusters.

7.3.3 CONSTRUCTION OF CLUSTERS

In our algorithm some conditions under which subclusters can be regarded as not being included in the same cluster are employed instead of merging operations. All pairs of subclusters are examined whether they meet the conditions or not, and then the pair satisfying none of the conditions is regarded as being included in the same cluster. Four conditions are obtained by analyzing how man detects such two-dimensional clusters shown in Fig. 7.1.

Condition 7.1: The two subclusters are not in touch with each other (Step 7).

Condition 7.2: The difference in point density between the two subclusters is greater than a certain index (Step 8).

Condition 7.3: The size of the touching region of the two subclusters is smaller than a certain index (Step 10).

Condition 7.4: The difference in point density between the touching region and each subcluster is greater than a certain index (Step 11).

The pairs of subclusters that meet at least one of the above conditions are regarded as independent of each other.* The detailed discussion of each condition is presented below.

a) Condition 7.1

At a first glance, it may be obvious that two subclusters which are not in touch with each other are regarded as independent of each other. As a matter of fact, however, the touch between point sets is a rather vague concept. Let us take Fig. 7.4 as an example. In this

---

* Two subclusters are said to be independent of each other when they are not included in the same cluster.

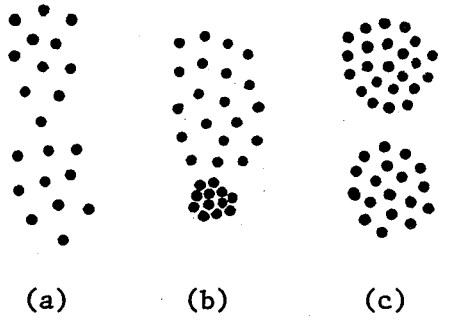
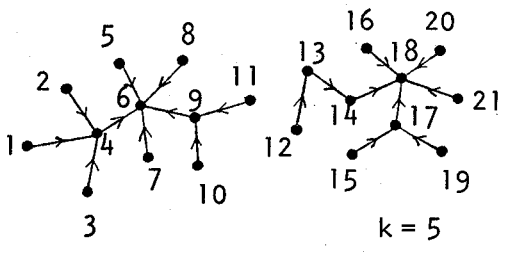


Fig. 7.4 Touching clusters.



subcluster 1 subcluster 2

Fig. 7.5 Touching subclusters.

figure the clusters in (c) are the only clusters that can be immediately said not to be in touch with each other. Whether or not the two clusters in (a), especially those in (b) are in touch with each other is rather vague. Thus, it is seen that the concept of touch between clusters is not clear even for man.

We have introduced the concept of k-touch between point sets in Section 7.2 in order to treat this problem quantitatively. An example is presented below. As to the point set in Fig. 7.5,

$$S_5(10) = \{10, 7, 9, 11\}$$
$$S_5(11) = \{11, 9, 12, 13, 8, 10\}$$
$$S_5(12) = \{12, 11, 13, 15\}$$
$$S_5(13) = \{13, 14, 12, 16, 11\}$$
$$Y_5^{1,2} = \{11\} \qquad Y_5^{2,1} = \{12, 13\}$$

are obtained. In this case, subclusters 1 and 2 are $(1,2)_5$-adjacent to and hence in 5-touch with each other. One also sees that two clusters in Fig. 7.4 (b) are $(1,1)_5$-adjacent to and hence in 5-touch with each other, and those in Fig. 7.4 (c) are not in 5-touch with each other. For $k \geq 8$, however, the clusters in Fig. 7.4 (c) are in k-touch with each other, so that k must be less than 8. In the algorithm 5-touch is employed, which will be discussed in Section 7.4 again.

b) Condition 7.2

Two point sets are usually regarded as independent of each other if there is a great difference in density between them. However, the index of the discrimination is not clear. In our algorithm, therefore, a threshold parameter α is introduced and the following decision rule is employed: Assign the two subclusters to different clusters when

the proportion of the higher potential to the lower potential exceeds α. For example, α=1.4 detects the two clusters in Fig. 7.6 (b) and (c), and regards the points in Fig. 7.6 (a) as one cluster (See Table 7.2). Note that α is not fixed at 1.4 but variable. Users can determine α according to their aims, since different values of α offer different indexes of detecting the differnces in point density. This flexibility is also given to the other parameters β, γ, and δ. Step 9 is for modifying the boundaries between subclusters (See Fig. 7.7 which shows the operation).

c) Condition 7.3

Three examples of touching point sets are depicted in Fig. 7.8. For detecting touching clusters it is an efficient way to make a decision by considering the size of the neck between them. Suppose that the two subclusters m and n are obtained for appropriate k. Then, we define the index of touch of the subcluster m to the subcluster n as $\sigma[W_m]/\sigma[Y_5^{m,n}]$ and that of the subcluster n to the subcluster m as $\sigma[W_n]/\sigma[Y_5^{n,m}]$. By comparing the smaller index of the two with an appropriate threshold value, a decision can be made on whehter or not these two subclusters are independent of each other. In the case of the subclusters in Fig. 7.5 which is constructed from the point set in Fig. 7.8 (a),

$$\sigma[W_1]/\sigma[Y_5^{1,2}] = 11/1 = 11$$

$$\sigma[W_2]/\sigma[Y_5^{2,1}] = 10/2 = 5$$

are obtained. It is sufficient to set β less than 5 in order to regard the two subclusters as independent of each other. The latter condition

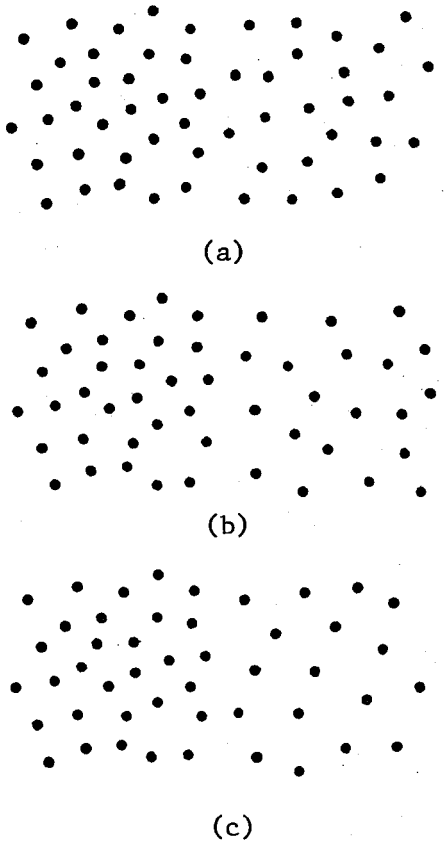


Fig. 7.6 Clusters with different point densities.

Table 7.2 Ratios with the potentials.

	(a)	(b)	(c)
ratio	1.22	1.42	1.54

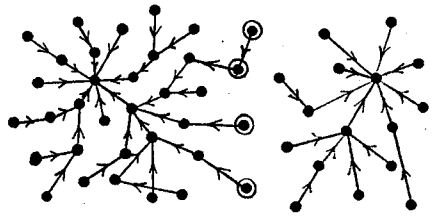


Fig. 7.7 Modification of the boundary between subclusters.

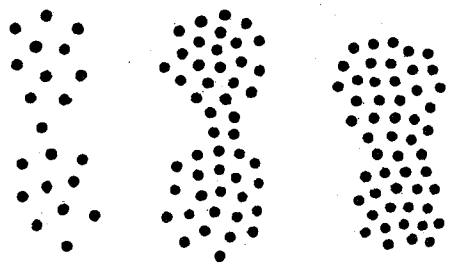


Fig. 7.8 Necks of point sets.

in Step 10 is for discriminating between the results by Condition 7.3 and those by Condition 7.4. The threshold parameter δ is infinity when there is no necessity of discriminating between them.

d) Condition 7.4

Let us consider Condition 7.4, taking the point sets in Fig. 7.9 as examples. The subclusters in (b) and (d) are constructed for k=6 by using the sets (a) and (c), respectively. The two subclusters in (b) are obviously in touch with each other. Moreover, there is neither a difference in density nor a neck between the two. Nevertheless, the two subclusters should be regarded as independent of each other. On the other hand, the point sets in (c) should be assigned to the same cluster. In order to distinguish between (b) and (d), a decision is made based on the potential of the boundary and that of each subcluster in Step 11. For example, from Table 7.3 listing the potentials of the subclusters in Fig. 7.9, one can see that it is sufficient to set γ at about 1.5.

From the discussion that we have made thus far, it is seen that the parameters α, β, and γ are threshold values which necessarily appear when the human operations of partitioning a two-dimensional point set into some clusters are formulated. Therefore, these parameters are fixed at appropriate values when users determine what kind of point set to regard as clusters.

7.4 COMPUTER SIMULATION AND DISCUSSION

In the computer study, simulation of our algorithm was made in detecting various two-dimensional clusters for α=1.4, β=4.8, γ=1.4,

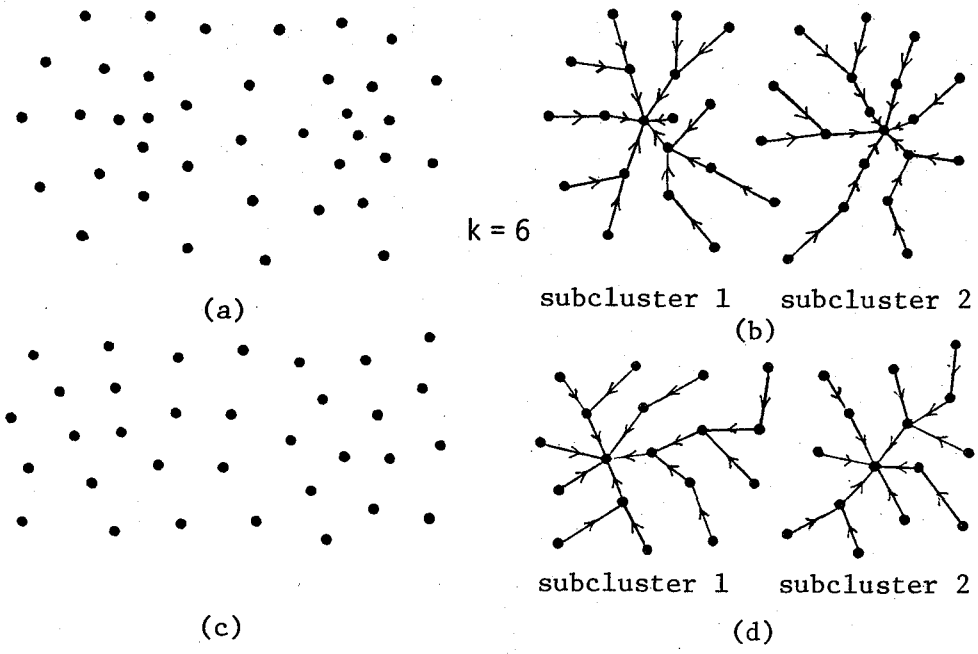


Fig. 7.9 Clusters with gradually varying point densities.

Table 7.3 Ratios of the potentials of the subclusters to those of the touching regions.

	(a)	(c)
$1/\sigma[X_5^{1,2}] \sum_{i \in X_5^{1,2}} P_6(i)$	1.70	1.34
$\max\{P_6^{SC}(1), P_6^{SC}(2)\}$	1.05	1.11
ratio	1.52	1.21

and $\delta=10^4$. All the clusters in Fig. 7.1 were detected successfully for k ($6 \leq k \leq 10$). These results show that our algorithm has the ability to detect all these clusters at once from the point set containing them all, which demonstrates a great advantage of our algorithm.

As is described in Section 7.1, our cluster detection algorithm has another advantage. It has a flexible structure, that is, by setting the parameters appropriately it can detect only the specific type of clusters required by users. Let us take the point sets in Fig. 7.1 as examples. Suppose that a user does not need to detect such clusters in (a) that are in touch with each other, though their point density is different. Then, the requirement is satisfied by setting $\alpha=\infty$. Suppose that another user intends to detect the difference in point density but does not need to detect such clusters in (c) that are in touch with each other having the same point density, though there is a neck. Then, by determining α appropriately and setting $\beta=\infty$, only the clusters specified by the user can be detected. This is the reason why we say the algorithm to have a flexible structure. Furthermore, roughly speaking, our algorithm coincides with Gitman's [23] and Jarvis-Patrick's [34] by setting $\alpha=\beta=\infty$ and by setting $\gamma=\infty$, respectively, so that it includes their algorithms as its special cases.

Our goal is to construct an algorithm for detecting such clusters that fit the concept of clusters based on visual intuition of man. However, we must note that such clusters are not always detected by our algorithm. Concerning the point sets in Fig. 7.10, for example, $(C_1, C_2, \ldots, C_6)$, $(C_1, C_2, C_3, C_4 \cup C_5, C_6)$, $(C_1 \cup C_2, C_3, C_4 \cup C_5, C_6)$, and $(C_1 \cup C_2, C_3 \cup \ldots \cup C_6)$ were detected as clusters for k=3, k=4,5, k=6,7, and $k \geq 8$, respectively. In this case, more reasonable clusters

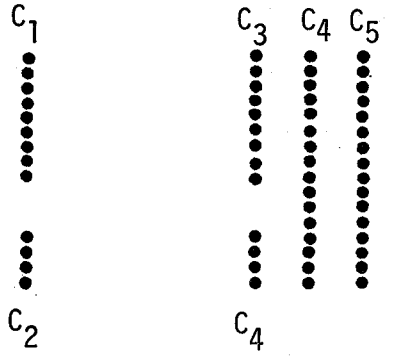


Fig. 7.10 An example of clusters which were not detected correctly.

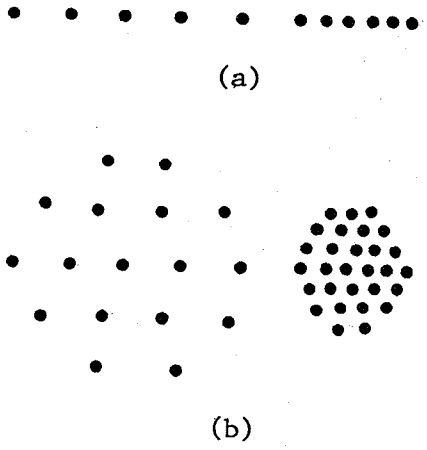


Fig. 7.11 Contacts between clusters.

such as $(C_1, C_2, C_3 \cup C_4, C_5, C_6)$ or $(C_1 \cup C_2, C_3 \cup C_4, C_5, C_6)$ were not detected. In spite of this failure, the above results reveal a remarkable characteristic of the parameter k. The result for k=3 corresponds to that of the precisest partition and the result for $k \geq 8$ corresponds to that of the roughest partition of the points. From this one can see that the parameter k indicating the size of neighborhood has a function as a measure representing the degree of roughness or preciseness of partitions.

We next consider the problem of multidimensional clusters. As is mentioned in Section 7.1, clusters of high dimensions are difficult to consider directly, since no rigorous definition of a cluster is available. In order to avoid this difficulty, we here assume that all clusters satisfy the four conditions proposed in Section 7.3.3 independently of the dimensionality. If the above assumption is accepted, our algorithm applies to every type of data, and reasonable results are expected to be obtained. It seems, however, that the touch between clusters needs to be considered again, since there is a little doubt on the performance of the 5-touch in the case of multidimensional clusters. At a first glance it may be seen that the higher the dimension of data becomes, the more neighbors need to be referred, so that the 5-touch defined by using a fixed number of neighbors does not work well. As a matter of fact, however, the 5-touch can be expected to show rather reasonable performance independent of the dimensionality, as is seen in the following. Let us take the point sets in Fig. 7.11 as examples. The point sets in (a) and (b) are similar to each other except for the dimensionality. According to our algorithm the two clusters in (a) are determined to be in 5-touch with each other, while

those in (b) are determined not to be in 5-touch with each other. However, it is natural for man to regard the clusters in (a) as being in touch with each other and those in (b) as not. Moreover, 3-dimensional globular clusters seem to be more likely to be regarded as independent of each other than 2-dimensional ones do. From the above discussion, it may be said that high dimensional clusters generally seem to be more compact than low dimensional ones do. Therefore, our k-touch can be a rather efficient measure representing the degree of touch between point sets by using a constant k independent of the dimensionality of data.

7.5 CONCLUSION

In this chapter, we have proposed a nonparametric algorithm for detecting clusters. The algorithm has been constructed by introducing hierarchical structure into data set based on the potential which is an efficient measure of point density. It has been shown that our algorithm is applicable to a wide range of data, and, though not complete, it can detect every type of clusters that man usually detects. Its flexibility has been also demonstrated.

CHAPTER 8

CONCLUDING REMARKS

In the present thesis, several estimation and learning algorithms have been studied, which are summarized as follows:

Chapter 2 has been concerned with the supervised nonparametric learning. By developing an algorithm for finding one of the optimal solutions of linear inequalities, a design algorithm of a piecewise linear discriminant function (PLDF) is obtained. A PLDF can approximate every kind of decision surfaces, so that our algorithm applies also to complicated pattern distributions.

Chapter 3 has treated nonsupervised signal detection. Two adaptive signal detectors converging to the optimal machine are constructed without knowing the probability of signal occurrence.

Chapter 4 has handled the problem of self-learning of a finite mixture. By extending the learning mechanism of DDM (decision-directed-machine), nonsupervised algorithm called WDDM (weighted-decision-directed-method) has been proposed. WDDM has a very simple structure and a great ability to decompose a finite mixture independent of the dimensionality of the mixture.

The last three chapters have dealt with nonsupervised nonparametric learning. In chapter 5, by restricting the discussion to the two-categry problem, a learning algorithm of a linear discriminant function (LDF) has been constructed. Our LDF works well even when the *a priori* probabilities are unknown, since the threshold value of the LDF

is so determined that the decision surface pass through the neck between the two pattern distributions of interest.

Chapter 6 has treated nonsupervised nonparametric learning in the multi-category problem. In order to design a discriminant function without memorizing patterns, an algorithm for estimating one of the modes of multimodal and multidimensional probability density function has been obtained. The efficiency of our mode estimation algorithm is demonstrated by applying it to nonparametric signal detection.

Chapter 7 has been concerned with cluster detection problem. Cluster detection is essentially a kind of nonsupervised nonparametric learning based on the stored sample patterns. In this chapter an efficient cluster detection algorithm has been obtained. It has been shown that the algorithm has a great ability to detect almost all types of clusters.

The author has been felt attracted to the problem of learning since he entered the graduate school.

Learning is an excellent function of human information processing and plays a prominent role in 'intelligence'. In spite of its importance the human learning mechanism is not known clearly. However, the purpose of learning can be defined as the extraction of some necessary information for a certain aim from a given stimulus. In pattern recognition, the stimulus is a set of sample patterns and the aim is to design a discriminant function. Then, the learning problem in pattern recognition is how to obtain a discriminant function having a low probability of misclassification from the given sample patterns.

Viewing 'learning' as described above, the author has made a

constant effort to study the learning problem in pattern recognition and obtained several results presented in this thesis. He believes that this thesis makes a steady step toward the completion of the theory of learning, particularly nonsupervised learning in pattern recognition.

REFERENCES

[1] Agrawala, A. K.: "Learning with a probabilistic teacher," IEEE Trans., IT-14, pp. 373-379 (July 1970).

[2] Albert, A.: "Regression and the Moore-Penrose pseudoinverse," Academic Press, New York (1972).

[3] Augustson, J. G. and J. Minker: "An analysis of some graph theoretical cluster techniques," J. ACM, 17, pp. 571-588 (Oct. 1970).

[4] Ball, G. H. and D. J. Hall: "ISODATA, An iterative method of multivariate analysis and pattern classification," presented at the Int. Commun. Conf., Philadelphia, Pa, (1966).

[5] Blum, J. R.: "Multidimensional stochastic approximation methods," Ann. Math. Statist., 25, pp. 737-744 (1954).

[6] Braverman, E. M.: "The method of potential functions in the problem of training machines to recognize patterns without a trainer," Automation and Remote Control, 27, pp. 1748-1770 (Oct. 1966).

[7] Cadzow, J. A.: "Synthesis of nonlinear decision boundaries by cascaded threshold gates," IEEE Trans., C-17, pp. 1165-1172 (Dec. 1968).

[8] Chang, C. L.: "Pattern recognition by piecewise linear discriminant functions," IEEE Trans., C-22, pp. 859-862 (Sept. 1973).

[9] Chen, C. H.: "A theory of Bayesian learning systems," IEEE Trans., SSC-5, pp. 30-37 (Jan. 1969).

[10] Chien, Y. T. and K. S. Fu: "On Bayesian learning and stochastic approximation," IEEE Trans., SSC-3 pp. 28-38 (June 1967).

[11] Cooper, D. B. and P. W. Cooper: "Adaptive pattern recognition and signal detection without supervision," IEEE Int'1. Conv. Rec. pt. 1, pp. 246-256 (1964).

[12] Davisson, L. D. and S. C. Schwartz: "Analysis of a decision-directed receiver with unknown priors," IEEE Trans., IT-16, pp. 270-276 (May 1970).

[13] Devyaterikov, I. P., A. I. Kaplinsky, and Y. Z. Tsypkin: "Convergence of learning algorithms," Automation and Remote Control, 30, pp. 1619-1626 (Oct. 1969).

[14] Diday, E.: "Optimization in non-hierarchical clustering," Pattern Recognition, 6, pp. 17-33 (Jan. 1974).

[15] Doob, J. L.: "Stochastic Processes," John Wiley (1953).

[16] Drake, K. W. and L. A. Gerhardt: "A class of pdf modeling algorithms," IEEE Trans., SMC-2, pp. 402-407 (July 1972).

[17] Dvoretzky, A.: "On stochastic approximation," Proc. Symp. Math. Statist. and Probability 3rd, Berkeley (1956).

[18] Fabian, V.: "Stochastic approximation of minima with improved asymptotic speed," Ann. Math. Statist., 38, pp. 191-200 (1967).

[19] Fralic, S. C.: "Learning to recognize patterns without a teacher," IEEE Trans., IT-13, pp. 57-64 (Jan. 1967).

[20] Friedman, H. P. and J. Rubin: "On some invariant criteria for grouping data," Journal of the American Statistical Association, 62, pp. 1159-1178 (Dec. 1967).

[21] Fukunaga, K. and W. L. G. Koontz: "A criterion and an algorithm for grouping data," IEEE Trans., C-19, pp. 917-923 (Oct. 1970).

[22] Gitman, I. and M. D. Levine: "An algorithm for detecting unimodal fuzzy sets and its application as a clustering technique," IEEE Trans., C-19, pp. 583-593 (July 1970).

[23] Gitman, I.: "An algorithm for nonsupervised pattern classification," IEEE Trans., SMC-3, pp. 66-74 (Jan. 1973).

[24] Gladyshev, E. G.: "On stochastic approximation," Theory of probability and its applications, 10, pp. 275-278 (1965).

[25] Henrichon, E. G. and K. S. Fu: "A nonparametric partitioning procedure for pattern classification," IEEE Trans., C-18, pp. 614-624 (July 1969).

[26] Ho, Y. C. and R. L. Kashyap: "An algorithm for linear inequalities and its applications," IEEE Trans., EC-14, pp. 683-688 (Oct. 1965).

[27] Hubert, L.: "Monotone invariant clustering procedures," Psychometrika, 38, pp. 47-62 (March 1973).

[28] Hubert, L.: "Min and max hierarchical clustering using asymmetric similarity measures," Psychometrika, 38, pp. 63-75 (March 1973).

[29] Ibaraki, T. and S. Muroga: "Adaptive linear classifier by linear programming," IEEE Trans., SSC-6, pp. 53-62 (Jan. 1970).

[30] Isomichi, Y.: "Floating disk method — A model of self-learning," Trans. of IECEJ, 55-D, pp. 435-441 (July 1972) (in Japanese).

[31] Isomichi, Y.: "Nonparametric learning of distribution function using stochastic approximation," Trans. of IECEJ, 56-D, pp. 291-297 (May 1973) (in Japanese).

[32] Isomichi, Y.: "Self-learning of finite-mixture distributions by using stochastic approximation method," Trans. of IECEJ, 56-D, pp. 696-701 (Dec. 1973) (in Japanese).

[33] Jakowatz, C. V., R. L. Shuey, and G. M. White: "Adaptive waveform recognition," 4th Int'l. Symp. on Information Theory, London (1960).

[34] Jarvis, R. A. and E. A. Patrick: "Clustering using a similarity measure based on shared near neighbors," IEEE Trans., C-22, pp. 1025-1034 (Nov. 1973).

[35] Johnson, S. C.: "Hierarchical clustering schemes," Psychometrika, 32, pp. 241-254 (Sept. 1967).

[36] Kabasawa, Y., S. Noguchi, and J. Oizumi: "Non-supervised learning by using stochastic approximation method," Trans. of IECEJ, 57-D, pp. 629-636 (Nov. 1974) (in Japanese).

[37] Kashyap, R. L. and C. C. Blaydon: "Estimation of probability density and distribution functions," IEEE Trans., IT-14, pp. 549-556 (July 1968).

[38] Keehn, D. G.: "A note on learning for Gaussian properties," IEEE Trans., IT-11, pp. 126-132 (Jan. 1965).

[39] Kesten, H.: "Accelerated stochastic approximation," Ann. Math. Statist., 29, pp. 41-59 (1958).

[40] Kiefer, J. and J. Wolfowitz: "Stochastic estimation of the maximum of a regression function," Ann. Math. Statist., 23, pp. 462-466 (1952).

[41] Koontz, W. L. G. and K. Fukunaga: "A nonparametric valley-seeking technique for cluster analysis," IEEE Trans., C-21, pp. 171-178 (Feb. 1972).

[42] Krasulina, T. P.: "Method of stochastic approximation in the determination of the largest eigenvalue of the mathematical expectation of random matrices," Automation and Remote Control, 31, pp. 215-221 (Feb. 1970).

[43] Kruskal, J. B.: "Multidimensional scaling by optimizing goodness of fit to a nonparametric hypothesis," Psychometrika, 29, pp. 1-27 (June 1964).

[44] Kruskal, J. B.: "Nonmetric multidimensional scaling: a numerical method," Psychometrika, 29, pp. 115-129 (June 1964).

[45] Ling, R. F.: "A probability theory of cluster analysis," Journal of the American Statistical Association, 68, pp. 159-164 (March 1973).

[46] Loeve, M.: "Probability theory," 3rd ed. Van Nostrand, New York (1963).

[47] Mangasarian, O. L.: "Multisurface method of pattern separation," IEEE Trans., IT-14, pp. 801-807 (Nov. 1968).

[48] Mangasarian, O. L.: "Nonlinear programming," McGraw-Hill, New York (1969).

[49] Mizoguchi, R. and M. Shimura: "Some considerations on an unsupervised learning scheme using the maximal eigenvector," Tech. Rep. of IECEJ, PRL73-80, pp. 41-49 (Jan. 1974) (in Japanese).

[50] Mizoguchi, R. and M. Shimura: "Nonparametric learning without a teacher based on mode estimation," Tech. Rep. of IECEJ, PRL74-46, pp. 13-23 (Jan. 1975) (in Japanese).

[51] Mizoguchi, R. and M. Shimura: "On a clustering algorithm using hierarchical structure," Record of the National Convention of IECEJ, 1196, p. 1103 (March 1975) (in Japanese).

[52] Mizoguchi, R. and M. Shimura: "A cluster detection algorithm based on hierarchical structure," Tech. Rep. of IECEJ, PRL75-2, pp. 13-24 (Apr. 1975) (in Japanese).

[53] Mizoguchi, R. and M. Shimura: "An approach to unsupervised learning classification," IEEE Trans., C-24, pp. 979-983 (Oct. 1975).

[54] Mizoguchi, R. and M. Shimura: "Parametric learning without a teacher — Self-learning of a finite mixture by WDDM," Tech. Rep. of IECEJ, PRL75-72, pp. 31-36 (Jan. 1976) (in Japanese).

[55] Mizoguchi, R. and M. Shimura: "Nonparametric learning without a teacher based on mode estimation," Trans. of IECEJ, 59-D, pp. 196-203 (March 1976) (in Japanese).

[56] Mizoguchi, R., M. Shimura, and M. Kizawa: "Piecewise linear discriminant functions in statistical pattern recognition," Tech. Rep. of IECEJ, PRL76-5, pp. 37-45 (Apr. 1976) (in Japanese).

[57] Mizoguchi, R. and M. Shimura: "A cluster detection algorithm based on hierarchical structure," Trans. of IECEJ, 59-D, pp. 451-458 (July 1976) (in Japanese).

[58] Mizoguchi, R. and M. Shimura: "A parametric learning method without a teacher — WDDM," Trans., of IECEJ, 59-D, pp. 719-724 (Oct. 1976) (in Japanese).

[59] Mizoguchi, R. and M. Shimura: "Nonsupervised learning without a teacher based on mode estimation," IEEE Trans., C-25, pp. 1109-1117 (Nov. 1976).

[60] Mizoguchi, R., M. Shimura, and M. Kizawa: "Piecewise linear discriminant functions in pattern recognition," Trans. of IECEJ, 60-D (1977) (to be published in Japanese).

[61] Nagaraja, G. and G. Krishna: "An algorithm for the solution of linear inequalities," IEEE Trans., C-23, pp. 421-427 (Apr. 1974).

[62] Nilson, N. L.: "Learning machines," McGraw-Hill, New York (1965).

[63] Parzen, E. "On estimation of a probability density function and mode," Ann. Math. Statist., 33, pp. 1065-1076 (1962).

[64] Patrick, E. A. and J. C. Hancock: "Nonsupervised sequential classification and recognition of patterns," IEEE Trans., IT-12, pp. 362-372 (July 1966).

[65] Patrick, E. A. and J. P. Costello: "Asymptotic probability of error using two decision-directed estimators for two unknown mean vectors," IEEE Trans., IT-14, pp. 160-162 (Jan. 1968).

[66] Patrick, E. A.: "On a class of unsupervised estimation problems," IEEE Trans., IT-14, pp. 407-415 (May 1968).

[67] Patrick, E. A. and F. P. Fischer II: "Cluster mapping with experimental computer graphics," IEEE Trans., C-18, pp. 987-991 (Nov. 1969).

[68] Patrick, E. A. and J. P. Costello: "On unsupervised estimation algorithms," IEEE Trans., IT-16, pp. 556-569 (Sept. 1970).

[69] Robbins, H. and S. Monro: "A stochastic approximation method," Ann. Math. Statist., 22, pp. 400-407 (1951).

[70] Rosenblatt, F.: "Principles of neurodynamics: Perceptrons and the theory of brain mechanizms," Spartan Books, Washington, D. C. (1961).

[71] Ruspini, E. H.: "A new approach to clustering," Information and Control, 15, pp. 22-32 (1969).

[72] Ryzin, J. V.: "On strong consistency of density estimates," Ann. Math. Statist., 40, pp. 1765-1772 (1969).

[73] Sammon, J. W., Jr.: "A nonlinear mapping for data structure analysis," IEEE Trans., C-18, pp. 401-409 (May 1969).

[74] Saridis, G. N., Z. J. Nikolic, and K. S. Fu: "Stochastic approximation algorithms for system identification, estimation, and decomposition of mixtures," IEEE Trans., SSC-5, pp. 8-15 (Jan. 1969).

[75] Scudder, H. J.: "Adaptive communication receivers," IEEE Trans., IT-11, pp. 167-174 (Apr. 1965).

[76] Scudder, H. J.: "Probability of error of some adaptive pattern recognition machines," IEEE Trans., IT-11, pp. 363-371 (July 1965).

[77] Shepard, R. N.: "The analysis of proximities: multidimensional scaling with an unknown distance function — I," Psychometrika, 27, pp. 125-139 (1962).

[78] Shepard, R. N.: "The analysis of proximities: multidimensional scaling with an unknown distance function — II," Psychometrika, 27, pp. 219-246 (1962).

[79] Shimauchi, T., S. Noguchi, and J. Oizumi: "The empirical Bayes approach to unsupervised parameter estimation and classification problems for mixture distributions," Trans. of IECEJ, 58-D, pp. 473-480 (Aug. 1975) (in Japanese).

[80] Shimura, M., T. Imai, and R. Mizoguchi: "Nonsupervised learning classifiers with optimal performance," Proc. of the First International Joint Conference on Pattern Recognition, Washington, D. C., pp. 255-262 (1973).

[81] Shimura, M. and T. Imai: "Nonsupervised classification using the principal component," Pattern Recognition, 5, pp. 353-363 (Apr. 1973).

[82] Smith, F. W.: "Pattern classifier design by linear programming," IEEE Trans., C-17, pp. 367-372 (Apr. 1968).

[83] Tanaka, K. and S. Tamura: "Some considerations on a type of pattern recognition using nonsupervised learning procedure," IFAC Int'l. Symp. on Technical and Biological problems of Control, Yerevan, Armenia (Sept. 1968).

[84] Teicher, H.: "Identifiability of mixtures," Ann. Math. Statist., 32, pp. 244-248 (1961).

[85] Teicher, H.: "Identifiability of finite mixtures," Ann. Math. Statist., 34, pp. 1265-1269 (1963).

[86] Teicher, H.: "Identifiability of mixtures of product measures," Ann. Math. Statist., 38, pp. 1300-1302 (1967).

[87] Tsypkin, Y. Z.: "Self-learning — What is it?" IEEE Trans., AC-13, pp. 608-612 (Dec. 1968).

[88] Venter, J. H.: "An extension of the Robbins-Monro procedure," Ann. Math. Statist., 38, pp. 181-190 (1967).

[89] Ward, J. H., Jr.: "Hierarchical grouping to optimize an objective function," Journal of the American Statistical Association, 58, pp. 236-244 (March 1963).

[90] Warmack, R. E. and R. C. Gonzalez: "An algorithm for the optimal solution of linear inequalities and its application to pattern recognition," IEEE Trans., C-22, pp. 1065-175 (Dec. 1973).

[91] Wassel, G. N. and J. Sklansky: "Training a one-dimensional classifier to minimize the probability of error," IEEE Trans., SMC-2, pp. 533-541 (Sept. 1972).

[92] Widrow, B. and M. E. Hoff: "Adaptive switching circuits," 1960 IRE WESCON Conv. Rec., Part 4, pp. 96-104 (Aug. 1960).

[93] Yakowitz, S. J. and J. Spragins: "On the identifiability of finite mixtures," Ann. Math. Statist., 39, pp. 209-214 (1968).

[94] Yakowitz, S. J.: "Unsupervised learning and identifiability of finite mixtures," IEEE Trans., IT-16, pp. 330-338 (May 1970).

[95] Yau, S. S. and J. M. Schumpert: "Design of pattern classifiers with the updating property using stochastic approximation techniques," IEEE Trans., C-17, pp. 861-872 (Sept 1968).

[96] Young, T. Y. and G. Coraluppi: "Stochastic estimation of a mixture of normal density functions using an information criterion," IEEE Trans., IT-16, pp. 258-263 (May 1970.

[97] Young, T. Y. and A. A. Farjo: "On decision-directed estimation and stochastic approximation," IEEE Trans., IT-18 pp. 671-673 (Sept. 1972).

[98] Zahn, C. T.: "Graph-theoretical methods for detecting and describing gestalt clusters," IEEE Trans., C-20, pp. 68-86 (Jan. 1971).