A Rate-Distortion Theoretic Approach to Pattern Recognition

Mohammad Reza Soleymani

A Thesis

in

The Department

of

Electrical and Computer Engineering

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

December, 1987

ISBN 0-315-44240-9

ABSTRACT

A Rate-Distortion Theoretic Approach to Pattern Recognition

Mohammad Reza Soleymani, Ph.D.
Concordia University, 1987.

In this work, a fundamental model for the pattern recognition process based on a generalized communication system model is presented. Applying this model to several examples, interesting conclusions including the relationship between the performance of the classifier and the number and the quality of the features is drawn. Based on these conclusions, the idea of vector recognition, i.e., concurrent classification of several patterns is suggested and substantiated. To remedy the problem of computational complexity which may result as a consequence of considering several patterns at one time, a new sample set condensing method and several new fast nearest neighbor search algorithms are presented and other possibilities are discussed.

To memory of my father Abbas Soleymani

and

To my mother Seddigheh Fallah

ACKNOWLEDGEMENTS

I would like to express my most sincere gratitude towards my friend and teacher Prof. S.D. Morgera for his valuable guidance, encouragement and constructive criticism without which this work would have been impossible.

More than any other person, I am indebted to my wife Fariba. I take this opportunity to express my appreciation for her love, understanding and patience throughout our marriage and during the course of my studies.

I would like also to thank all my fellow graduate students for their encouragement and help from time to time. I am very grateful to Mr. Y.R. Shayan for proof reading the text of this work and for his help in its final compilation.

This research work was supported by Canada NSERC Grant A0912 and A6824, and Québec FCAR Grant EQ-0350.

Table of Contents

List of Figures

LIST OF TABLES

CHAPTER 1

Introduction

The fields of information theory, in particular, the area of rate-distortion theory, and pattern recognition stand as well developed disciplines. While the areas of interest to researchers in the two fields have overlapped in the past [1]-[6], up to now no comprehensive effort has been made to relate the philosophy, goals, and analytical techniques of these two disciplines. This work is motivated by the belief that such an examination of the two fields would uncover a number of interesting new research questions, would add to the understanding of the fields, both separately and together, and would provide a basis for increased collaboration among researchers.

Rate-distortion theory is the branch of information theory dealing with data compression. A rate-distortion function R(D) may be associated with each given stochastically modeled information source and a distortion measure. This function gives the minimum number of bits per sample, i.e., the rate $R(D)$, required to represent the source so that it can be reproduced with an average distortion not exceeding a certain value D [7].

Pattern recognition is defined as the categorization of the input data (patterns) into identifiable classes via the extraction of significant features or attributes of the data from a background of irrelevant detail [8]. In this sense any pattern recognition problem can be viewed as a data compression problem. On the other hand, multivariate distributions, defined over the pattern space, provide a suitable model for the variability of pattern representation, i.e., the pattern generating mechanism can be modeled using statistical distributions, in a manner similar to the modeling of information sources and channels in information theory. Statistical modeling of the pattern generating mechanism together with the application of the concepts from statistical decision theory has resulted in a branch of pattern recognition, called statistical or decision theoretic pattern recognition, which has dominated the field for the last two decades [9]-[12].

The above considerations point to a close relationship between pattern recognition and rate-distortion theory. By treating the patterns to be categorized as the outputs of a noisy channel and assigning a numerical penalty to each erroneous classification, it is possible to formulate a communication channel model for the pattern recognition process, similar to the model used for deriving the rate-distortion function. In this sense, the rate-distortion function can provide a relationship between the average probability of classification error and the number and the quality of the observations.

The first push in the direction of applying rate-distortion theory to pattern recognition was made by Pearl [2] and Crolotte and Pearl [3, 4]. Pearl [2] treated the pattern recognition process as a question answering system. In his formulation, the rate-distortion function provides a lower bound on the number of classification rules that the system should memorize versus the accuracy of its answers. His results are limited to statistically independent patterns and show that, in this case, the reduction in the memory space induced by the error-tolerance is not drastic. More recently, Chou and Gray [6] have used rate-distortion theory in order to demonstrate the substantial sub-optimality of pattern recognition systems employing a decision tree structure.

In this work, we present a fundamental model for the pattern recognition process based on the generalization of rate-distortion proposed by Dobrushin and Tsybakov [13]. Using this model, we find the explicit expression for the rate-distortion function for the case of independent, equiprobable classes and also derive a simple recursive method for the numerical evaluation of the rate-distortion function when the a priori probabilities are not equal. We also compute tight bounds to the rate-distortion function in the case of correlated patterns. The result of such computation indicates that a reduction in the probability of error for a given rate is possible by increasing the number of patterns considered at one time. In our formulation, the rate then corresponds to the average number of dichotomies answered by the pattern recognizer in order to classify the patterns. Treating the patterns in batch also enables us to achieve fractional rates, i.e., less than one binary question answered per pattern. These conclusions parallel those which support the use of vector quantization instead of scalar quantization in source coding. Based on these results, we propose the idea of vector recognition, i.e., classifying several patterns concurrently, and using a typical example, demonstrate the superiority of this approach over conventional pattern-by-pattern classification. By computing the rate-distortion function for several dimensions, i.e., the number of patterns considered at a time, and for different degrees of degradation, it is shown that the improvement in performance resulting from the increase in dimension depends on the quality of the observation.

1.1 Major Contributions of the Work

The main objective of this thesis is to present a rate-distortion theoretic formulation for the basic problem of pattern recognition and apply a number of concepts from rate-distortion theory to pattern recognition. The major contributions of this work can be summarized as follows:

1- A fundamental model for the pattern recognition process based on the notion of a noisy source is presented.

2- Applying the model to two-class pattern recognition, the explicit expression for the rate-distortion function for the case of independent patterns and equal a priori probabilities is found and a simple recursive method for numerical evaluation of the rate-distortion function when the a priori probabilities are not equal is derived.

3- For correlated patterns, tight bounds to the rate-distortion function are computed.

4- Based on the numerical computation of the rate-distortion function for a different number of patterns and different degrees of degradation, several interesting conclusions, including the advantage of delayed decision or vector recognition are drawn.

5- Justification for a feature selection methodology based on the direct use of the expression for the probability of error is presented.

6- A new method for condensing the sample set in nearest neighbor pattern classification is derived and discussed.

7- Several new fast nearest neighbor search algorithms are derived and discussed.

1.2 Structure of the Thesis

The thesis consists of two parts. The first part, Chapters 2 and 3, is devoted to pattern recognition. These two chapters, while containing several new results in parametric and nonparametric statistical pattern classification, outline the basic concepts of statistical pattern recognition and serve as an introduction for the remainder of the thesis. The second part of the thesis, Chapters 4-7, is devoted to an exploration of the relationship between rate-distortion theory and pattern recognition. A brief summary of the chapters follows.

In Chapter 2, we discuss the parametric (Bayes) pattern recognition. In this chapter, beside presenting the necessary concepts of decision theoretic pattern recognition, essential to the understanding of the relationship established later between pattern recognition and data compression, a feature selection strategy based on the direct utilization of the expression for the Bayes probability of error is formulated and justified. Examples of application of this method to two-class Gaussian pattern classification are also included.

Chapter 3 is devoted to a class of nonparametric pattern classification methods, called nearest neighbor pattern classification. In this chapter, first, different nearest neighbor rules are discussed briefly and their performance is compared with that of optimal (Bayes) classification. Next, in order to overcome one of the major difficulties involved in nearest neighbor rules, arising from the infinite sample size assumption, a new algorithm for sample set condensing based on vector quantization and editing is proposed. Furthermore, in this chapter, several new fast algorithms for nearest neighbor search are presented. These algorithms, while very useful in conventional nearest neighbor pattern classification, are mainly intended to make the idea of nonparametric vector recognition more practical.

In Chapter 4, after presenting the necessary details from rate-distortion theory, the generalized communication model of Dobrushin and Tsybakov is discussed and used to model the pattern recognition process. In addition, the use of Blahut's computational algorithm is suggested for numerical calculation of the rate-distortion function and its applicability is demonstrated by computing the rate-distortion function for a specific example.

In Chapter 5, the general model of Chapter 4 is applied to the two-class pattern recognition problem. It is shown that, in the case of independent, equiprobable classes, due to the existing symmetry, the exact expression for the rate-distortion function can be found. In this chapter, a simple method is also presented for the numerical computation of the rate-distortion function in the case of two classes with different a priori probabilities.

In Chapter 6, tight bounds to the rate-distortion function for several examples with correlated patterns are computed. Based on these examples, several intersting conclusions are reached. One important conclusion points to the efficiency of delayed decision or vector recognition. Examples to support this point and other conclusions are also included.

Chapter 7 summarizes the contents of the thesis, outlines the important implications of the rate-distortion theoretic approach to pattern recognition, and offers suggestions in regard to future research directions.

CHAPTER 2

Parametric (Bayes) Pattern Recognition

As was stated in Chapter 1, in many situations the pattern generating mechanism can be modeled using well-known multivariate distribution (density) functions and, therefore, decision theoretic methods can be used to derive optimal classifiers. This results in parametric or Bayesian pattern recognition. Here, it is assumed that the pattern classification problem can be posed in a probabilistic way and that all probabilities involved are either known or can be derived from the data. In this chapter we discuss the parametric approach. We first present the general formulation of the problem and then specialize to the Gaussian case. Finally, a feature extraction method based on the direct use of the expression for the Bayes probability of error is presented. This method is based on the argument formulated by Kovalevsky [14]. Kovalevsky's argument is based on the fact that the most economical and the most informative feature is the decision function itself, and, therefore, dividing the classification task into two separate phases, i.e., first searching for the "good" features and then deriving the decision function based on these "good" features, inevitably results in a loss of information, and, therefore, is ineffective. This argument can be viewed as a direct result of the data processing theorem of information theory [15]. In our view, in parametric pattern recognition, this argument supports the direct use of the expression for the probability of error, instead of interclass distance measures which are only indirectly related to the probability of classification error, for the purpose of feature selection.

2.1 Formulation of the Pattern Classification Problem.

First, we model the pattern generating mechanism and then discuss the model for the decision-making process. Assume that there are M pattern classes $x_1, x_2, \ldots, x_M$ and an arbitrary pattern belongs to class x_i with a priori probability $P(x_i)$, $P(x_i) \geq 0$ and $\sum_{i=1}^{M} P(x_i) = 1$. Patterns are represented by k-dimensional feature vectors $\underline{z} = (z_1, \ldots, z_k)$. The feature vector $\underline{z}$ can be considered as a random vector taking values in some k-dimensional feature space Z. For example, if feature vectors are real-valued, i.e., $z_j \in \mathrm{R}$, $j = 1, \ldots, k$, then $Z \equiv \mathrm{R}^k$. The generation of patterns is assumed to be governed by the multivariate conditional probability density functions $p(\underline{z} \mid x_i)$, $i = 1, \ldots, M$. That is, when nature is in state x_i it generates patterns distributed according to $p(\underline{z} \mid x_i)$. These probability density functions model the inherent variability of the patterns as well as the approximation incurred due to our observation and measurement.

The a priori probabilities and the class conditional densities discussed above enable us to model a wide range of pattern generating mechanisms. Now, we model the decision-making or classification process. The function of the classifier can be specified by a function $f(\underline{z})$. This function specifies to which class a given pattern $\underline{z}$ should be assigned, e.g., $f(\underline{z}) = x_i$ means that the pattern $\underline{z}$ should be classified as a member of pattern class x_i. Usually, $f(\underline{z})$ takes values $x_1, \ldots, x_M$, however, in some cases when there is a great deal of ambiguity involved it is helpful to defer the decision about a given pattern by rejecting it for special handling. In such cases, an extra decision option is allowed, e.g., $f\,(\mathbf{z}) = x_0$, where x_0 denotes the reject option. Here, for an M-class problem we have $M + 1$ possible decisions. Another element which should be defined in order to complete the model is the loss function. A loss function $\lambda(x_i \mid x_j)$ is a non-negative real number specifying the penalty for deciding in favor of class x_i when the actual class is in fact x_j. The loss function determines the seriousness of each classification error. In the important special case, in which all wrong decisions are equally harmful and right decisions result in no penalty, we can use the probability of error criterion, i.e.,

$$\lambda(x_i \mid x_j) = \begin{cases} 0\,, & i = j\,, \\ 1\,, & i \neq j\,. \end{cases} \tag{2.1}$$

In the case with rejection option, assuming that a constant loss λ_r is incurred for each rejection, we have,

$$\lambda(x_i \mid x_j) = \begin{cases} 0\,, & i = j\,, \\ 1\,, & 0 \neq i \neq j\,, \\ \lambda_r\,, & i = 0\,, \end{cases} \tag{2.2}$$

where λ_r is called the rejection threshold.

2.2 Bayes Decision Rule

In terms of the definitions given in the previous section, the goal of statistical decision theory is to devise a decision rule $f\,(\mathbf{z})$ in such a way that the average loss per decision is as small as possible. In this section, we first examine the general loss function case and then specialize to the case for which we employ the probability of error criterion.

2.2.1 Bayes Decision Rule for Minimum Risk

Assume that we want to classify an arbitrary pattern $\underline{z}$ of unknown class x. Observation of the pattern $\underline{z}$ changes our information about the state of nature from a priori probabilities $P(x_i)$ to a posteriori probabilities $P(x_i \mid \underline{z})$. Where $P(x_i \mid \underline{z})$ is the probability that nature is in state x_i, given that pattern $\underline{z}$ is observed. The a posteriori probability $P(x_i \mid \underline{z})$ can be computed from $p(\underline{z} \mid x_i)$ by Bayes rule:

$$P(x_i \mid \underline{z}) = \frac{p(\underline{z} \mid x_i) P(x_i)}{p(\underline{z})}, \tag{2.3}$$

where

$$p(\underline{z}) = \sum_{j=1}^{M} p(\underline{z} \mid x_j) P(x_j). \tag{2,4}$$

Assume that upon observing a particular pattern $\underline{z}$, we decide that it belongs to class x_i while it actually belongs to x_j. By making this decision, we will incur the loss $\lambda(x_i \mid x_j)$. Since the probability that $\underline{z}$ belongs to x_j is $P(x_j \mid \underline{z})$, the expected loss associated with deciding x_i is,

$$R(x_i \mid \underline{z}) = \sum_{j=1}^{M} \lambda(x_i \mid x_j) P(x_j \mid \underline{z}). \tag{2.5}$$

In decision theoretic terminology, an expected loss is called risk , and $R(x_i \mid \underline{z})$ is known as the conditional risk. For any observation $\underline{z}$, the expected loss is minimized by selecting the class which guarantees the minimum conditional risk.

Since the decision rule is a function $f(\underline{z})$ that tells us which decision to make for every possible pattern $\underline{z}$, we can see from (2.5) that a particular decision function has the conditional risk,

$$R(f(\underline{z}) \mid \underline{z}) = \sum_{j=1}^{M} \lambda(f(\underline{z}) \mid x_j) P(x_j \mid \underline{z}) \,, \tag{2.6}$$

and the average risk is given by,

$$R = \int R(f(\underline{z}) \mid \underline{z}) p(\underline{z}) d\underline{z} \,, \tag{2.7}$$

where $d\underline{z}$ denotes a volume element in k-dimensional feature space and the integral extends over the entire feature space. Since $p(\underline{z}) \geq 0$ for all $\underline{z}$, it is clear that the integral in (2.7) can be minimized by minimizing the conditional risk for each $\underline{z}$, i.e., to minimize the overall risk, we should compute the conditional risk,

$$R(x_i \mid \underline{z}) = \sum_{j=1}^{M} \lambda(x_i \mid x_j) P(x_j \mid \underline{z}) \,, \tag{2.8}$$

for all i and select the class x_i for which $R(x_i \mid \underline{z})$ is minimum. In other words, the Bayes decision rule can be stated as follows,

$$\text{Decide } x_i \quad \text{if} \quad R(x_i \mid \underline{z}) \leq R(x_j \mid \underline{z}) \quad \text{for all } j \,. \tag{2.9}$$

Note that ties may possibly occur, i.e., it is possible that more than one decision minimizes the conditional risk. In such a case, the conditional risk is not affected by the way of breaking the tie and, therefore, any tie-breaking rule can be used.

From (2.9) we can see that the Bayes decision rule has the minimum conditional risk,

$$R^*(\underline{z}) = \min_{\text{all } i} R(x_i \mid \underline{z})$$

$$= \min_{all\ i} \sum_{j=1}^{M} \lambda(x_i \mid x_j) P(x_j \mid \underline{\mathbf{z}}) , \tag{2.10}$$

and the minimum average risk,

$$R^* = \int R^*(\underline{\mathbf{z}}) p(\underline{\mathbf{z}}) d\underline{\mathbf{z}} , \tag{2.11}$$

which is called the Bayes risk and is the limit of excellence beyond which it is impossible to go [12].

2.2.2 Error-Reject Tradeoff

We now consider the case of M class pattern recognition with $M+1$ decisions. The extra decision (reject option) is reserved for the case where there is not enough evidence for classifying a particular pattern into one of M classes. Assuming that correct decisions incur no penalty, all erroneous decisions are equally serious, and associating a constant loss with each rejection, we have the loss function of (2.2). With this choice of loss function, the conditional risk will be,

$$R(x_0 \mid \underline{\mathbf{z}}) = \sum_{j=1}^{M} \lambda_r P(x_j \mid \underline{\mathbf{z}}) = \lambda_r , \tag{2.12a}$$

and

$$R(x_i \mid \underline{\mathbf{z}}) = \sum_{\substack{j=1 \\ j \neq i}}^{M} P(x_j \mid \underline{\mathbf{z}}) = 1 - P(x_i \mid \underline{\mathbf{z}}) , \quad i=1, \ldots, M . \tag{2.12b}$$

We note that the conditional risk $R(x_i \mid \underline{\mathbf{z}})$, $i \neq 0$, in this case, is the conditional probability of classification error associated with the decision $f(\underline{\mathbf{z}}) = x_i$. Now, according to (2.9), the decision rule can be written as,

$$f(\underline{z}) = \begin{cases} x_i \,, & \text{if } R(x_i \mid \underline{z}) = \min\limits_{j \neq 0} R(x_j \mid \underline{z}) \leq \lambda_r \,, \\ & \\ x_0 \,, & \text{if } \lambda_r < \min\limits_{j \neq 0} R(x_j \mid \underline{z}) \,, \end{cases} \tag{2.13}$$

or using (2.12b),

$$f(\underline{z}) = \begin{cases} x_i \,, & \text{if } P(x_i \mid \underline{z}) = \max\limits_{j=1,\dots,M} P(x_j \mid \underline{z}) \geq 1 - \lambda_r \,, \\ & \\ x_0 \,, & \text{if } \max\limits_{j=1,\dots,M} P(x_j \mid \underline{z}) < 1 - \lambda_r \,. \end{cases} \tag{2.14}$$

It is clear that for the M-class problem, $\frac{1}{M} \leq \max\limits_{j=1,\dots,M} P(x_j \mid \underline{z}) \leq 1$, so for the reject option to be activated we should have $0 \leq \lambda_r < \frac{M-1}{M}$.

The decision function of (2.14) partitions the feature space into M acceptance regions $Z_1, \dots, Z_M$ and one rejection region Z_0. The acceptance region Z_i contains all of the patterns which are classified as x_i and is described by

$$Z_i \equiv \{\underline{z} \mid P(x_i \mid \underline{z}) = \max_{j=1,\dots,M} P(x_j \mid \underline{z}) \geq 1-\lambda_r\} \,. \tag{2.15}$$

The overall acceptance region is $Z_A = Z_1 \cup Z_2 \cup \dots \cup Z_M$. The rejection region contains all of the patterns which are rejected, i.e.,

$$Z_0 \equiv \{\underline{z} \mid 1 - \lambda_r > \max_{j=1,\dots,M} P(x_j \mid \underline{z})\} \,. \tag{2.16}$$

It is clear that $Z = Z_A \cup Z_0$, where Z denotes the entire feature space.

For a given pattern $\underline{z}$, the acceptance probability for the decision $f(\underline{z}) = x_i$ is,

$$P_{A,i} = \int_{Z_i} p(\underline{\mathbf{z}})d\underline{\mathbf{z}} \quad i=1, \ldots, M \; , \tag{2.17}$$

and the average acceptance probability or acceptance ratio is,

$$P_A = \sum_{i=1}^{M} P_{A,i} = \int_{Z_A} p(\underline{\mathbf{z}})d\underline{\mathbf{z}} \, . \tag{2.18}$$

Similarly. the probability of rejection or rejection rate is,

$$P_R = \int_{Z_0} p(\underline{\mathbf{z}})d\underline{\mathbf{z}} \, . \tag{2.19}$$

Acceptance of the decision $f(\underline{\mathbf{z}}) = x_i$ gives rise to either a classification error or correct decision with probabilities,

$$P_{E,i} = \int_{Z_i} [1 - P(x_i \mid \underline{\mathbf{z}})] p(\underline{\mathbf{z}})d\underline{\mathbf{z}} \, , \tag{2.20}$$

and

$$P_{C,i} = \int_{Z_i} P(x_i \mid \underline{\mathbf{z}}) p(\underline{\mathbf{z}})d\underline{\mathbf{z}} \, , \tag{2.21}$$

respectively. As for the acceptance rate, the average probability of error, or error rate is,

$$P_E = \sum_{i=1}^{M} P_{E,i} = \int_{Z_A} E^*(\underline{\mathbf{z}}) p(\underline{\mathbf{z}})d\underline{\mathbf{z}} \, , \tag{2.22}$$

and the average probability of correct decision is,

$$P_C = \sum_{i=1}^{M} P_{C,i} = \int_{Z_A} [1 - E^*(\underline{\mathbf{z}})] p(\underline{\mathbf{z}})d\underline{\mathbf{z}} \, , \tag{2.23}$$

where,

$$E^*(\underline{\mathbf{z}}) = \min_{i=1,\dots,M} R(x_i \mid \underline{\mathbf{z}}) = 1 - \max_{i=1,\dots,M} P(x_i \mid \underline{\mathbf{z}}) . \tag{2.24}$$

Given the underlying distributions, the acceptance rate P_A of the Bayes rule is a function of the rejection threshold only. The reason for this is that the boundary of the acceptance region Z_i of (2.15) over which the integral of (2.17) is calculated depends on λ_r only. Therefore, we can write $P_A = P_A(\lambda_r)$. The same argument applies to the probabilities of rejection, classification error, and correct decision; thus, we can denote them by $P_R(\lambda_r)$, $P_E(\lambda_r)$, and $P_C(\lambda_r)$. It is easy to see that these probabilities are related by,

$$P_A(\lambda_r) = 1 - P_R(\lambda_r) , \tag{2.25}$$

and

$$P_A(\lambda_r) = P_E(\lambda_r) + P_C(\lambda_r) . \tag{2.26}$$

The overall risk associated with the Bayes rule can also be written as $R^* = R^*(\lambda_r)$ and is related to the classification error probability and the rejection rate as,

$$\begin{aligned} R^*(\lambda_r) &= \int_Z R^*(\underline{\mathbf{z}}) p(\underline{\mathbf{z}}) d\underline{\mathbf{z}} = \int_Z \min[E^*(\underline{\mathbf{z}}), \lambda_r] p(\underline{\mathbf{z}}) d\underline{\mathbf{z}} \\ &= \int_{Z_A} E^*(\underline{\mathbf{z}}) p(\underline{\mathbf{z}}) d\underline{\mathbf{z}} + \int_{Z_0} \lambda_r p(\underline{\mathbf{z}}) d\underline{\mathbf{z}} \\ &= P_E(\lambda_r) + \lambda_r P_R(\lambda_r) . \end{aligned} \tag{2.27}$$

From (2.27), the optimality of the Bayes decision rule can be formulated as follows: For a given pattern classification problem, among all the classifiers with rejection rate equal to $P_R(\lambda_r)$, the Bayes classifier has the lowest average probability of classification error. Or, alternatively, for a given probability of classification error, the Bayes rule rejects less patterns than any other classification scheme.

From (2.25) and (2.26), we observe that the probabilities $P_R(\lambda_r)$, $P_A(\lambda_r)$, $P_E(\lambda_r)$, and $P_C(\lambda_r)$ are not independent and knowing any two of them is sufficient to completely characterize the performance of the Bayes decision rule. The following theorem takes us one further step and shows that knowing just one of these probabilities over the full range of λ_r enables us to compute the other three.

Theorem 2.1 : Let $P_E(\lambda_r)$ and $P_R(\lambda_r)$ denote the probabilities of classification error and rejection of the Bayes rule. Then, we have,

$$P_E(\lambda_r) = -\int_0^{\lambda_r} \lambda \, dP_R(\lambda) \,. \tag{2.28}$$

Proof:

Denote the rejection region for a certain value λ of the rejection threshold by Z_0. Now, assume that we decrease the rejection threshold by $\Delta\lambda$, i.e., $\lambda' = \lambda - \Delta\lambda$. This results in an expansion of the rejection region from Z_0 to $Z'_0 = Z_0 \cup \Delta Z_0$. Any pattern $\underline{z}$ in the incremental region ΔZ_0 would have been accepted if the rejection threshold had been λ, i.e.,

$$E^*(\underline{z}) = \min_{j\neq 0} R(x_j \mid \underline{z}) \leq \lambda \,, \tag{2.29}$$

whereas the same pattern will be rejected now that the rejection threshold is reduced to $\lambda - \Delta\lambda$, i.e.,

$$E^*(\mathbf{z}) > \lambda - \Delta\lambda\,. \tag{2.30}$$

So, for any $\mathbf{z} \in \Delta Z_0$, we have,

$$\lambda - \Delta\lambda < E^*(\mathbf{z}) \leq \lambda\,, \tag{2.31}$$

or,

$$(\lambda - \Delta\lambda)p(\mathbf{z}) < E^*(\mathbf{z})p(\mathbf{z}) \leq \lambda p(\mathbf{z})\,. \tag{2.32}$$

Integrating (2.32) with respect to $\mathbf{z}$ over the incremental region ΔZ_0, we have,

$$(\lambda - \Delta\lambda)\Delta P_R(\lambda) < -\Delta P_E(\lambda) \leq \lambda\Delta P_R(\lambda)\,. \tag{2.33}$$

Now, going from the incremental change $\Delta\lambda$ to the differential $d\lambda$, $\Delta P_E(\lambda) \rightarrow dP_E(\lambda)$, $\Delta P_R(\lambda) \rightarrow dP_R(\lambda)$, and $\Delta\lambda\Delta P_R(\lambda) \rightarrow 0$. Therefore, the upper and lower limits in (2.33) coincide and we have,

$$dP_E(\lambda) = -\lambda dP_R(\lambda) \tag{2.34}$$

Integrating with respect to λ, we get the desired result, i.e.,

$$P_E(\lambda_r) = -\int_0^{\lambda_r} \lambda\, dP_R(\lambda) \qquad \blacksquare \tag{2.35}$$

Relation (2.28) describes the tradeoff between the probability of classification error and the rejection rate. It indicates that the knowledge of the rejection rate versus λ is the only thing needed for calculating the average probability of classification error. This relation proves useful in classifier error rate estimation[16].

2.2.3 Minimum Error Rate Classification

Here, we consider the case of the zero-one loss function of (2.1). The difference with the case studied above is that we do not have the reject option and upon observing a pattern, the classifier decides in favor of one of M pattern classes. Note that the fact that we take zero and one as the cost of correct and erroneous decisions does not involve any loss of generality. We show in Appendix 2.A that for a loss function which assigns constant costs to correct and wrong decisions, e.g., λ_c and λ_e, respectively, the Bayes decision rule will be the same as that of the zero-one loss function.

The case considered here can be looked at yet in another way. We showed that in the previous case for the reject option to be activated we should have $\lambda_r < \frac{M-1}{M}$. Therefore, the present case can be considered as a special case of the previous one, i.e., with $\lambda_r \geq \frac{M-1}{M}$.

The importance of the case considered here is that the risk associated with this loss function is precisely the average probability of error, since the conditional risk in this case is,

$$R\,(x_i \mid \underline{z}) = \sum_{j=1}^{M} \lambda(x_i \mid x_j) P\,(x_j \mid \underline{z})$$

$$= \sum_{j \neq i} P\,(x_j \mid \underline{z})$$

$$= 1 - P\,(x_i \mid \underline{z})\,, \tag{2.36}$$

and $P(x_i \mid \underline{z})$ is the conditional probability that x_i is the correct decision. In order to minimize the overall risk, the classifier should make decisions in a way which minimizes the conditional risk for any particular pattern. Thus, to minimize the average probability of error, we should select i such that the <u>a posteriori</u> probability $P(x_i \mid \underline{\mathbf{z}})$ is maximized. Therefore, the minimum error rate decision rule is,

$$\text{Decide } x_i \text{ if } P(x_i \mid \underline{\mathbf{z}}) > P(x_j \mid \underline{\mathbf{z}}) \text{ for all } j \neq i . \tag{2.37}$$

Then, the average probability of error associated with this decision rule is,

$$\begin{aligned} R^* = P_E &= \int_Z [1 - \max_i P(x_i \mid \underline{\mathbf{z}})] p(\underline{\mathbf{z}}) d\underline{\mathbf{z}} \\ &= \sum_{i=1}^{M} \int_{Z_i} [1 - P(x_i \mid \underline{\mathbf{z}})] p(\underline{\mathbf{z}}) d\underline{\mathbf{z}} \\ &= 1 - \sum_{i=1}^{M} \int_{Z_i} P(x_i \mid \underline{\mathbf{z}}) p(\underline{\mathbf{z}}) d\underline{\mathbf{z}} = 1 - P_C , \end{aligned} \tag{2.38}$$

where P_C is the average probability of correct decision.

2.3 Two-Class Pattern Recognition

In this section, we consider the two-class pattern recognition problem. We specialize the results for the general case considered in the previous section in order to derive the decision rule and the expression for the probability of error. In the next section, we discuss the example of Gaussian or normal patterns.

In this case the Bayes decision rule can be simply stated as: decide x_1 if $P(x_1 \mid \underline{\mathbf{z}}) > P(x_2 \mid \underline{\mathbf{z}})$ and decide x_2 otherwise. Alternatively, we can define the discriminant function [10],

$$g(\underline{\mathbf{z}}) = P(x_1 | \underline{\mathbf{z}}) - P(x_2 | \underline{\mathbf{z}}) \,, \tag{2.39}$$

and then state the decision rule as: decide x_1 if $g(\underline{\mathbf{z}}) > 0$ and decide x_2 otherwise. It is necessary to point out that the choice of the discriminant function is not unique. First, notice that from (2.3), the condition $P(x_1 | \underline{\mathbf{z}}) > P(x_2 | \underline{\mathbf{z}})$ is equivalent to,

$$P(x_1)p(\underline{\mathbf{z}} | x_1) > P(x_2)p(\underline{\mathbf{z}} | x_2) \,, \tag{2.40}$$

and, therefore, choosing

$$g(\underline{\mathbf{z}}) = P(x_1)p(\underline{\mathbf{z}} | x_1) - P(x_2)p(\underline{\mathbf{z}} | x_2) \,, \tag{2.41}$$

as the discriminant function has the same effect as that of (2.39). On the other hand, for any monotonically increasing function $f(.)$,

$$f\,[P(x_1)p(\underline{\mathbf{z}} | x_1)] > f\,[P(x_2)p(\underline{\mathbf{z}} | x_2)] \,, \tag{2.42}$$

is equivalent to (2.40) and hence choosing

$$g(\underline{\mathbf{z}}) = f\,[P(x_1)p(\underline{\mathbf{z}} | x_1)] - f\,[P(x_2)p(\underline{\mathbf{z}} | x_2)] \,, \tag{2.43}$$

results in the same decision rule. A common choice for the function $f(.)$ is the logarithmic function. Using the logarithmic function, we get the following discriminant function,

$$g(\underline{\mathbf{z}}) = \log \frac{p(\underline{\mathbf{z}} | x_1)}{p(\underline{\mathbf{z}} | x_2)} + \log \frac{P(x_1)}{P(x_2)} \,. \tag{2.44}$$

Using (2.38), the probability of error for the two-class case is found to be,

$$P_E = \int_{Z_1} [1 - P(x_1 \mid \mathbf{z})] p(\mathbf{z}) d\mathbf{z} + \int_{Z_2} [1 - P(x_2 \mid \mathbf{z})] p(\mathbf{z}) d\mathbf{z} , \qquad (2.45)$$

where

$$Z_1 \equiv \{ \mathbf{z} \mid P(x_1 \mid \mathbf{z}) > P(x_2 \mid \mathbf{z}) \} , \qquad (2.46)$$

and $Z_2 \equiv \overline{Z}_1$. Noting that $P(x_1 \mid \mathbf{z}) + P(x_2 \mid \mathbf{z}) = 1$ and using (2.3), we obtain,

$$P_E = P(x_1) \int_{Z_2} p(\mathbf{z} \mid x_1) d\mathbf{z} + P(x_2) \int_{Z_1} p(\mathbf{z} \mid x_2) d\mathbf{z} . \qquad (2.47)$$

2.4 Two-Class Gaussian Problem

The general multivariate normal density is written as,

$$p(\mathbf{z}) = (2\pi)^{-\frac{k}{2}} \mid \Sigma \mid^{-\frac{1}{2}} \exp \left[-\frac{1}{2} (\mathbf{z} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{z} - \boldsymbol{\mu}) \right] , \qquad (2.48)$$

where $\mathbf{z} = (z_1, \ldots, z_k)^T$ is a k-dimensional column vector, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_k)^T$; where $\mu_i = E[z_i]$, and Σ is a $k \times k$ covariance matrix with (i,j)th element

$$\sigma_{ij} = E[(z_i - \mu_i)(z_j - \mu_j)] . \qquad (2.49)$$

The notation $\mid \Sigma \mid$ denotes the determinant of Σ.

The covariance matrix Σ is symmetric, positive semidefinite. However, we are mostly interested in the case in which Σ is positive definite. Since, if the rank of Σ is $k' < k$, we can reduce the problem to a k'-dimensional problem by selecting a subset of the original features consisting of k' linearly independent variables for which the resulting covariance matrix Σ' is then positive definite.

It is common to write (2.48) as $p(\underline{z}) = \mathrm{N}(\underline{\mu}\ ,\ \Sigma)$. It can be easily shown that the distribution of any linear combination of normally distributed random variables is again normal[17]. In particular, if A is a $k \times n$ matrix and $\underline{y} = A^T \underline{z}$ is an n-dimensional vector, then $p(\underline{y}) = \mathrm{N}(A^T \underline{\mu}\ ,\ A^T \Sigma A\ )$.

Now, assume that we have two classes with a priori probabilities $P(x_1)$ and $P(x_2)$, and also assume that the class conditional densities are Gaussian, i.e.,

$$p(\underline{z} \mid x_i) = (2\pi)^{-\frac{k}{2}} \mid \Sigma_i \mid^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\underline{z} - \underline{\mu}_i)^T \Sigma_i^{-1}(\underline{z} - \underline{\mu}_i)\right] ,\quad i=1,2\ . \tag{2.50}$$

The discriminant function of the type (2.44) for this case can be easily found to be,

$$g(\underline{z}) = -\frac{1}{2}\log\frac{\mid \Sigma_1 \mid}{\mid \Sigma_2 \mid} - \frac{1}{2}(\underline{z} - \underline{\mu}_1)^T \Sigma_1^{-1}(\underline{z} - \underline{\mu}_1)$$

$$+ \frac{1}{2}(\underline{z} - \underline{\mu}_2)^T \Sigma_2^{-1}(\underline{z} - \underline{\mu}_2) + \log\frac{P(x_1)}{P(x_2)}\ . \tag{2.51}$$

The decision boundaries defined by $g(\underline{z}) \equiv 0$ are, in the general case, hyperquadrics and can assume any of the forms - pairs of hyperplanes, hyperspheres, hyperellipsoids, hyperparaboloids, and hyperhyperboloids. From (2.47), the probability of error is,

$$P_E = P(x_1) \int_{g(\underline{z})<0} p(\underline{z} \mid x_1) d\underline{z} + P(x_2) \int_{g(\underline{z})>0} p(\underline{z} \mid x_2) d\underline{z}\ . \tag{2.52}$$

2.4.1 Case of Equal Covariance Matrices

Now, we consider an important special case where patterns belonging to two classes have the same covariance matrices but different mean vectors, i.e., $\Sigma_1 = \Sigma_2 = \Sigma$. For this case, the first term in (2.51) is equal to zero and expanding the second and third terms, the quadratic form $\underline{z}^T \Sigma^{-1} \underline{z}$ is canceled and we have,

$$g(\underline{z}) = [\underline{z} - \frac{1}{2}(\underline{\mu}_1+\underline{\mu}_2)]^T \Sigma^{-1}(\underline{\mu}_1-\underline{\mu}_2) + \log \frac{P(x_1)}{P(x_2)} . \tag{2.53}$$

The discriminant function of (2.53) is linear and, therefore, the decision boundary in this case is a hyperplane.

Define,

$$L(\underline{z}) = [\underline{z} - \frac{1}{2}(\underline{\mu}_1+\underline{\mu}_2)]^T \Sigma^{-1}(\underline{\mu}_1-\underline{\mu}_2) \tag{2.54}$$

and,

$$\alpha = \log \frac{P(x_1)}{P(x_2)} . \tag{2.55}$$

Then the Bayes decision rule may be expressed as: decide x_1 if $L(\underline{z}) > \alpha$, and decide x_2 otherwise. The average probability of error for this decision rule can be written as,

$$P_E = P(x_1) Pr [L(\underline{z}) \leq \alpha \mid x_1] + P(x_2) Pr [L(\underline{z}) > \alpha \mid x_2] . \tag{2.56}$$

When $\underline{z}$ belongs to the class x_1 it is a Gaussian random vector distributed as, $p(\underline{z}) = \mathrm{N}(\underline{\mu}_1 , \Sigma)$. Since $L(\underline{z})$ is a linear function of $\underline{z}$, it is also Gaussian with mean,

$$E[L(\underline{z})] = \overline{L(\underline{z})} = [\underline{\mu}_1 - \frac{1}{2}(\underline{\mu}_1 + \underline{\mu}_2)]^T \Sigma^{-1}(\underline{\mu}_1 - \underline{\mu}_2)$$

$$= \frac{1}{2}(\underline{\mu}_1 - \underline{\mu}_2)^T \Sigma^{-1}(\underline{\mu}_1 - \underline{\mu}_2) \, , \tag{2.57}$$

and variance,

$$E\left[(L(\underline{z}) - \overline{L(\underline{z})})^2\right]$$

$$= (\underline{\mu}_1 - \underline{\mu}_2)^T \Sigma^{-1}(\underline{\mu}_1 - \underline{\mu}_2) \, . \tag{2.58}$$

Thus, $L(\underline{z})$ has the density $N(\frac{1}{2}\Delta \, , \, \Delta)$, where Δ is the Mahalanobis distance between $\underline{\mu}_1$ and $\underline{\mu}_2$, i.e.,

$$\Delta = (\underline{\mu}_1 - \underline{\mu}_2)^T \Sigma^{-1}(\underline{\mu}_1 - \underline{\mu}_2) \, . \tag{2.59}$$

Similarly, it can be shown that for $\underline{z}$ belonging to x_2, the probability density function for $L(\underline{z})$ is $N(-\frac{1}{2}\Delta \, , \, \Delta)$. Therefore, (2.56) can be written as,

$$P_E = \frac{P(x_1)}{\sqrt{2\pi\Delta}} \int_{-\infty}^{\alpha} \exp[-\frac{1}{2\Delta}(w - \frac{\Delta}{2})^2]dw$$

$$+ \frac{P(x_2)}{\sqrt{2\pi\Delta}} \int_{\alpha}^{\infty} \exp[-\frac{1}{2\Delta}(w + \frac{\Delta}{2})^2]dw \, . \tag{2.60}$$

This expression can also be written as,

$$P_E = P(x_1)Erfc(-\frac{\alpha}{\sqrt{\Delta}} + \frac{\sqrt{\Delta}}{2}) + P(x_2)Erfc(\frac{\alpha}{\sqrt{\Delta}} + \frac{\sqrt{\Delta}}{2}) \, , \tag{2.61}$$

where,

$$Erfc(x) \equiv \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{y^2}{2}} dy \, . \tag{2.62}$$

Now, we show that the probability of error is a monotonically decreasing function of Δ and, subsequently suggest a feature selection criterion for this case. Differentiating (2.60) with respect to Δ, we have,

$$\frac{dP_E}{d\,\Delta} = -(\frac{\alpha}{2\Delta\sqrt{\Delta}} + \frac{1}{4\sqrt{\Delta}})P(x_1)\exp\left[-\frac{1}{2}(-\frac{\alpha}{\sqrt{\Delta}} + \frac{\sqrt{\Delta}}{2})^2\right]$$

$$-(-\frac{\alpha}{2\Delta\sqrt{\Delta}} + \frac{1}{4\sqrt{\Delta}})P(x_2)\exp\left[-\frac{1}{2}(\frac{\alpha}{\sqrt{\Delta}} + \frac{\sqrt{\Delta}}{2})^2\right]. \qquad (2.63)$$

Noting that $\alpha = \log\frac{P(x_1)}{P(x_2)}$, and performing some algebraic manipulation, (2.63) becomes,

$$\frac{dP_E}{d\,\Delta} = -\frac{\sqrt{P(x_1)P(x_2)}}{2\sqrt{\Delta}}\exp\left[-\frac{1}{2}(\frac{\alpha^2}{\Delta} + \frac{\Delta}{4})\right] \qquad (2,64)$$

It is clear that $\frac{dP_E}{d\,\Delta} < 0$ and, therefore, P_E is a monotonically decreasing function of Δ. This fact can be used in selecting the best features in this case. In particular, assume that Σ is a diagonal matrix with diagonal elements $\sigma_1^2, \ldots, \sigma_k^2$. This assumption does not entail any loss of generality, since, in general, the matrix Σ can be written as $\Sigma = P^{-1}\Lambda P$, where Λ is a diagonal matrix, and performing the change of variable $\underline{y} = P\underline{z}$, we have $p(\underline{y} \mid x_i) = \mathrm{N}(\underline{\mu}_i' , \Lambda)$, $i = 1, 2$, where, $\underline{\mu}_i' = P\underline{\mu}_i$, $i = 1, 2$. Therefore, the case of an arbitrary covariance matrix can be reduced to the case of a diagonal matrix with a rotation of coordinates.

In the case of $\Sigma = diag\,[\sigma_1^2, \ldots, \sigma_k^2]$, Δ can be written as ,

$$\Delta = \sum_{j=1}^{k}\frac{(\mu_{1j} - \mu_{2j})^2}{\sigma_j^2}. \qquad (2.65)$$

Now, assume that we want to select $k' < k$ features from the k original features. It is natural to choose the k' features associated with the largest values of $\frac{|\mu_{1j} - \mu_{2j}|}{\sigma_j}$.

2.4.2 Case of Equal Mean Vectors

Here, we consider the case where $p(\underline{z} \mid x_i) = N(\underline{\mu}, \Sigma_i)$, $i=1,2$. Since, with a change of variable $\underline{y} = \underline{z} - \underline{\mu}$ we have $p(\underline{y} \mid x_i) = N(0, \Sigma_i)$, $i=1,2$, without loss of generality, we assume $\underline{\mu} = 0$. For this case, the discriminant function $g(\underline{z})$ can be written as,

$$g(\underline{z}) = -\frac{1}{2}\log\frac{|\Sigma_1|}{|\Sigma_2|} - \frac{1}{2}\underline{z}^T(\Sigma_1^{-1} - \Sigma_2^{-1})\underline{z} + \log\frac{P(x_1)}{P(x_2)}, \qquad (2.66)$$

and the probability of error is,

$$P_E = P(x_1)(2\pi)^{-\frac{k}{2}}|\Sigma_1|^{-\frac{1}{2}}\int_{Z_2}\exp\left[-\frac{1}{2}\underline{z}^T\Sigma_1^{-1}\underline{z}\right]d\underline{z}$$

$$+ P(x_2)(2\pi)^{-\frac{k}{2}}|\Sigma_2|^{-\frac{1}{2}}\int_{Z_1}\exp\left[-\frac{1}{2}\underline{z}^T\Sigma_2^{-1}\underline{z}\right]d\underline{z}, \qquad (2.67)$$

where,

$$Z_1 \equiv \{\underline{z} \mid \underline{z}^T(\Sigma_1^{-1} - \Sigma_2^{-1})\underline{z} + \log\frac{|\Sigma_1|}{|\Sigma_2|} < 2\log\frac{P(x_1)}{P(x_{2)}}\}, \qquad (2.68)$$

and $Z_2 = \overline{Z}_1$. Since Σ_1 and Σ_2 are real symmetric and positive definite, we can find a matrix R such that,

$$R^T\Sigma_1 R = I \quad \text{and} \quad R^T\Sigma_2 R = \Lambda,$$

where Λ is a diagonal matrix with diagonal elements $\lambda_1 \geq \ldots \geq \lambda_k \geq 0$ which are roots of the equation $| \Sigma_2 - \lambda \Sigma_1 | = 0$ [18]. Let $\underline{y} = R^T \underline{z}$ in (2.67) to obtain,

$$P_E = P(x_1)(2\pi)^{-\frac{k}{2}} \int_{Y(\lambda)} \exp[-\frac{1}{2}\underline{y}^T \underline{y}] d\underline{y}$$

$$+ P(x_2)(2\pi)^{-\frac{k}{2}} |\Lambda|^{-\frac{1}{2}} \int_{Y(\lambda)} \exp[-\frac{1}{2}\underline{y}^T \Lambda^{-1} \underline{y}] d\underline{y} , \qquad (2.69)$$

where,

$$Y(\underline{\lambda}) \equiv \{\underline{y} \mid \sum_{i=1}^{k} (1-\frac{1}{\lambda_i}) y_i^2 > \sum_{i=1}^{k} \log\lambda_i + 2\log\frac{P(x_1)}{P(x_2)}\} . \qquad (2.70)$$

The probability of error can be equivalently written as,

$$P_E = P(x_1) Pr \left[\sum_{i=1}^{k} (1-\frac{1}{\lambda_i}) Y_i^2 > \sum_{i=1}^{k} \log\lambda_i + 2\log\frac{P(x_1)}{P(x_2)} \right]$$

$$+ P(x_2) Pr \left[\sum_{i=1}^{k} (\lambda_i - 1) Y_i^2 < \sum_{i=1}^{k} \log\lambda_i + 2\log\frac{P(x_1)}{P(x_2)} \right] , \qquad (2.71)$$

where the Y_i's are independent $N(0\ ,\ 1)$ random variables.

2.4.3 A Feature Selection Algorithm

Here, we show that the expression for the probability of error given by (2.73) can be used to evaluate the features. The following theorem shows that those features for which λ_i is farther from 1 are more suitable in the sense that they decrease the probability of error more. We will use this result for deriving a feature selection algorithm.

Theorem 2.2 : The probability of error defined by (2.73) is monotonically decreasing with λ_i for each $\lambda_i > 1$ and monotonically increasing with λ_i for each $\lambda_i < 1$.

The proof of the theorem may be found in the Appendix 2.B. ■

The above theorem can be used in order to extract a set of $l < k$ features from the original k features. In particular, when all λ_i's are greater (smaller) than one, the problem is straightforward. In such a case, we pick $l < k$ largest (smallest) λ_i's and form the feature vector $\underline{\mathbf{y}} = L^T \underline{\mathbf{z}}$, where the ith column of the matrix L is the eigenvector of $\Sigma_1^{-1}\Sigma_2$ corresponding to the eigenvalue λ_i. When some of the λ_i's are smaller than one and some are greater than one the problem is more difficult. However, the following observations make the search for good features easier.

Define $l+1$ subsets $S_l(p)$, $p=0, \ldots, l$ of the set $\{\lambda_1, \ldots, \lambda_k\}$ as follows

$$S_l(p) \equiv \{\lambda_1, \ldots, \lambda_p, \lambda_{n-(l-p)}, \ldots, \lambda_k\}$$

The quantities $\lambda_1, \ldots, \lambda_p$ are the p smallest, and $\lambda_{n-(l-p)}, \ldots, \lambda_k$ are the $l-p$ largest, eigenvalues. $S_l(0)$ contains the l largest eigenvalues and $S_l(l)$ contains the l smallest eigenvalues. A natural consequence of Theorem 2.2 is that the eigenvalues corresponding to the best l features are given by one of the subsets $S_l(p)$, $p=0, \ldots, l$. It is also clear that if $S_l(m)$ outperforms $S_l(m+1)$ it also outperforms $S_l(m')$, $m' > m+1$. Similarly, if $S_l(m)$ outperforms $S_l(m-1)$, it also outperforms $S_l(m'')$, $m'' < m-1$.

Based on the above observations, we propose the following algorithm for the feature selection:

Step 1 : Find $\lambda_1, \ldots, \lambda_k$, the eigenvalues of $\Sigma_1^{-1}\Sigma_2$.

Step 2 : Find $P_l(0)$ and $P_l(l)$ the probability of errors corresponding to $S_l(0)$ and $S_l(l)$, respectively. If $P_l(l) < P_l(0)$, then set $j = l-1$ and go to 5. Else, set $j = 1$ and go to 3.

Step 3 : If $j < l$, find $P_l(j)$ the probability of error corresponding to $S_l(j)$. Else, set $S_l^* = S_l(j-1)$ and stop.

Step 4 : If $P_l(j) < P_l(j-1)$, set $j = j+1$ and go to 3. Else, set $S_l^* = S_l(j-1)$ and stop.

Step 5 : If $j > 0$ then, find $P_l(j)$. Else $S_l^* = S_l(j+1)$ and stop.

Step 6 : If $P_l(j) < P_l(j+1)$, set $j = j+1$ and go to 5. Else $S_l^* = S_l(j+1)$ and stop.

In the above algorithm $P_l(j)$ is the probability of error when the feature set with eigenvalues represented by $S_l(j)$ is used, i.e.,

$$P_l(j) = P(x_1)Pr\left[\sum_{i \in I_l(j)} (1-\frac{1}{\lambda_i})y_i^2 > \sum_{i \in I_l(j)} \log\lambda_i + 2\log\frac{P(x_1)}{P(x_2)}\right]$$

$$+ P(x_2)Pr\left[\sum_{i \in I_l(j)} (\lambda_i - 1)y_i^2 < \sum_{i \in I_l(j)} \log\lambda_i + 2\log\frac{P(x_1)}{P(x_2)}\right],$$

where $I_l(j)$ is the set of indices of eigenvalues belonging to $S_l(j)$, i.e., $I_l(j) \equiv \{1, \ldots, j, k-(l-j), \ldots, k\}$, and y_i's are i.i.d. $N(0, 1)$ random variables. $P_l(j)$ can be found using Monte-Carlo simulation. For large values of k and l, the asymptotic approximate formula of Morgera-Datta[19] can be used.

We have applied the above algorithm to four examples which are of practical interest. In the first and second example, the feature vectors are distributed according to a first-order Gauss-Markov density. The covariance matrices in this case are of Toeplitz form, with an element σ_{ij} of the covariance matrix given as,

$$\sigma_{ij} = \sigma_{|i-j|} = e^{-\alpha |i-j|} .$$

In the first example, we use $\alpha_1=1$ for the patterns belonging to the first class and $\alpha_2=0.5$ for those belonging to the second class. The second example uses $\alpha_1=1.0$ and $\alpha_2=0.25$. In the third example the patterns are generated according to a second-order Gauss-Markov density, with an element of the covariance matrix given as,

$$\sigma_{|i-j|} = e^{-\beta}\sigma_{|i-j|-1} + e^{-\gamma}\sigma_{|i-j|-2} .$$

We have used $\beta_1=1.0$ and $\gamma_1=1.4$ for the first class and $\beta_2=0.2$ and $\gamma_2=2.0$ for the second class. In the fourth example, the first covariance matrix is first-order Markov with $\alpha_1=2.0$ and the second one is second order Markov with $\beta_2=0.5$ and $\gamma_2=2.0$.

In each case, we have assumed equal a priori probabilities. Each feature vector consists of 40 features. Table 2.1 shows the best 10 features extracted from the 40 original features in each case. These examples illustrate a compression of the original data space by 75%.

Example	Σ_1	Σ_2	Eigenvalues Selected	
			Largest	Smallest
1	First-order Markov $\alpha_1=1.0$	First-order Markov $\alpha_2=0.5$	10	0
2	First-order Markov $\alpha_1=1.0$	First-order Markov $\alpha_2=0.25$	10	0
3	Second-order Markov $\beta_1=1.0$, $\gamma_1=1.4$	Second-order Markov $\beta_2=0.2$, $\gamma_2=2$	10	0
4	First-order Markov $\alpha_1=2.0$	Second-order Markov $\beta_2=0.5$, $\gamma_2=2.0$	8	2

Table 2.1 : Eigenvalues selected using the new method.

2.5 Discussion

In this chapter, we discussed the parametric approach to pattern recognition in general, and its application to the case of probability of error loss function and Gaussian feature vectors. Also, a feature selection methodology based on the direct use of the expression for the probability of error was introduced and its application in two special cases of interest was discussed.

In the parametric pattern classification, the pattern generating mechanism is modeled in a manner similar to the modeling of the source and channel in information theory. This not only enables us to view the pattern classification problem as a hypothesis testing problem, and, therefore apply the decision theoretic concepts in order to design and evaluate the performance of the optimum classifiers, but also, as we will show in Chapter 4, makes it possible to apply a communication system model to the pattern recognition problem in order to obtain lower bounds on the performance of the classifier versus its complexity.

While the modeling of pattern generating mechanism using multivariate distributions is valid in general, the assumption made in parametric pattern classification regarding the availability of these distributions is not always true. This observation calls for a different approach, called the nonparametric pattern classification method, which does not require advance knowledge of the pattern statistics. In the next chapter, we will discuss a category of nonparametric methods called the nearest neighbor rules.

Appendices

2.A Classification Rule for the General Loss Function

In this appendix, we show that the Bayes decision rule for the loss function,

$$\lambda(x_i \mid x_j) = \begin{cases} \lambda_c & i=j \\ \lambda_e & i\neq j \end{cases} \qquad (2.A.1)$$

is the same as the one derived for the zero-one loss function of (2.1). The conditional risk for the loss function of (2.A.1) is,

$$R(x_i \mid \underline{z}) = \sum_{j=1}^{M} \lambda(x_i \mid x_j) P(x_j \mid \underline{z})$$

$$= \lambda_c P(x_i \mid \underline{z}) + \lambda_e \sum_{j\neq i} P(x_j \mid \underline{z})$$

$$= \lambda_e - (\lambda_e - \lambda_c) P(x_i \mid \underline{z}) \qquad (2.A.2)$$

Since $\lambda_e > \lambda_c$, in order to minimize the conditional risk we need to maximize the a posteriori probability $P(x_i \mid \underline{z})$, i.e., the decision rule is,

$$\text{Decide } x_i \quad \text{if} \quad P(x_i \mid \underline{z}) > P(x_j \mid \underline{z}) \quad \text{for all } j\neq i$$

which is the same as the decision rule given by (2.37).

2.B Proof of Theorem 2.2

In this appendix, we provide a proof for Theorem 2.2. Assume that we have $\underline{\lambda}=(\lambda_1, \dots, \lambda_k)$. Take $\underline{\lambda}' =(\lambda_1' , \dots, \lambda_k')$ which differs from $\underline{\lambda}$ only in one component, e.g., $\lambda_j' > \lambda_j > 1$. Denote the probability of error corresponding to $\underline{\lambda}$ and $\underline{\lambda}'$ by $P_E(\underline{\lambda})$ and $P_E(\underline{\lambda}')$, respectively, then from (2.69), we have,

$$P_E(\underline{\lambda}') = P(x_1)(2\pi)^{-\frac{k}{2}} \int_{Y(\underline{\lambda}')} \exp\left[-\frac{1}{2}\sum_{i=1}^{k} y_i^2\right] d\underline{y}$$

$$+ P(x_2)(2\pi)^{-\frac{k}{2}} (\prod_{i=1}^{k} \lambda_i')^{-\frac{1}{2}} \int_{\bar{Y}(\underline{\lambda}')} \exp\left[-\frac{1}{2}\sum_{i=1}^{k} \frac{y_i^2}{\lambda_i'}\right] d\underline{y}$$

$$< P(x_1)(2\pi)^{-\frac{k}{2}} \int_{Y(\underline{\lambda})} \exp\left[-\frac{1}{2}\sum_{i=1}^{k} y_i^2\right] d\underline{y}$$

$$+ P(x_2)(2\pi)^{-\frac{k}{2}} (\prod_{i=1}^{k} \lambda_i')^{-\frac{1}{2}} \int_{\bar{Y}(\underline{\lambda})} \exp\left[-\frac{1}{2}\sum_{i=1}^{k} \frac{y_i^2}{\lambda_i'}\right] d\underline{y}$$

$$= P(x_1)Pr\left[\sum_{i=1}^{k}(1-\frac{1}{\lambda_i})y_i^2 > \sum_{i=1}^{k}\log\lambda_i + 2\log\frac{P(x_1)}{P(x_2)}\right]$$

$$+ P(x_2)Pr\left[\sum_{i=1}^{k}(1-\frac{1}{\lambda_i})\lambda_i' \, y_i^2 < \sum_{i=1}^{k}\log\lambda_i + 2\log\frac{P(x_1)}{P(x_2)}\right]$$

$$\leq P(x_1)Pr\left[\sum_{i=1}^{k}(1-\frac{1}{\lambda_i})y_i^2 > \sum_{i=1}^{k}\log\lambda_i + 2\log\frac{P(x_1)}{P(x_2)}\right]$$

$$+ P(x_2)Pr\left[\sum_{i=1}^{k}(\lambda_i-1)y_i^2 < \sum_{i=1}^{k}\log\lambda_i + 2\log\frac{P(x_1)}{P(x_2)}\right] = P_E(\underline{\lambda})$$

The first inequality is due to optimality of the Bayes decision rule and the second equality results from the fact that $(1-\frac{1}{\lambda_i})\lambda_i' \geq (1-\frac{1}{\lambda_i})\lambda_i = \lambda_i-1$ for all i. This proves the first part of Theorem 2.2, i.e., it shows that the probability of error is monotonically decreasing with λ_j for each $\lambda_j>1$.

To prove the second part of the theorem, we take $\underline{\lambda}''$ which is different from $\underline{\lambda}$ only in one component $\lambda_l'' < \lambda_l < 1$ and in a similar way, show that $P_E(\underline{\lambda}'') < P_E(\underline{\lambda})$.

CHAPTER 3

Nearest Neighbor Pattern Classification

The Bayes classification rule discussed in Chapter 2 relies on the assumption that the parametric form of the pattern generating mechanism is known in advance. However, in many practical situations such an assumption is not valid. This gives rise to the use of nonparametric methods that can be used without assuming that the form of the underlying densities is known.

There are several different nonparametric methods. These can be divided into two main categories. The first category consists of those techniques which try to estimate the underlying densities, e.g., using Parzen windows [20], [21], and then design the optimal classifier based on the estimated densities. The second category attempts to totally bypass the probability estimation by deriving the decision rule directly from the available data samples. Different nearest neighbor rules are examples of this approach. This chapter is devoted to this category of non-parametric techniques. First, different nearest neighbor rules are discussed and some practical problems involved in using these methods are addressed, e.g., the computational complexity and storage requirement resulting from a large sample size assumption. Next, in order to remedy these problems, a new method for sample set condensing based on a combination of vector quantization and editing techniques, and several new fast nearest neighbor search algorithms is presented.

3.1 Nearest Neighbor Rules

The nearest neighbor rules exchange the need for the knowledge of the underlying distributions for that of knowing a large number of correctly classified sample patterns. By doing so, the nearest neighbor rules bring us a lot closer to the reality of the practical problems. The basic idea behind the nearest neighbor rules is that the patterns which are closer in the feature space are likely either to belong to the same class or to have about the same a posteriori probability distribution. The first case gives rise to the single nearest neighbor rule, 1-NN, while the second prompts the use of a k-nearest neighbor rule, k-NN. The first formulation of NN rules was given by Fix and Hodges [22], [23]. Cover and Hart[24] have demonstrated the admissibility of the 1-NN rule, in the sense that it has strictly lower probability of error than any other k-NN rule for certain classes of distributions. They have also established the relationship between the probability of error for the 1-NN rule and the minimum (Bayes) probability of error in terms of a lower and an upper bound.

3.1.1 1-NN Classification Rule

Assume that we have a set S_n of n pairs of statistically independent, identically distributed random variables $(\underline{\mathbf{z}}_1 , x_1) , \ldots , (\underline{\mathbf{z}}_n , x_n)$, where $\underline{\mathbf{z}}_i$'s take values in a space Z upon which a metric d is defined, and the x_i's take value in the set $\{1 , \ldots , M\}$. Each x_i represents the actual (true) class of $\underline{\mathbf{z}}_i$. Now, assume that a new pair $(\underline{\mathbf{z}} , x)$ is drawn according to the same distribution, but we are only given $\underline{\mathbf{z}}$ and asked to estimate its class x. The single nearest neighbor rule decides that $\underline{\mathbf{z}}$ belongs to the class x' if $\underline{\mathbf{z}}' \in \{\underline{\mathbf{z}}_1 , \ldots , \underline{\mathbf{z}}_n\}$ is the closest point to $\underline{\mathbf{z}}$ according to the metric d, where x' is the category of $\underline{\mathbf{z}}'$. In other words,

$$\hat{x} = x' \quad \text{if} \quad d(\underline{z}, \underline{z}') = \min_{i=1,\ldots,n} d(\underline{z}, \underline{z}_i) . \tag{3.1}$$

It is easy to show that if the metric space Z is separable then the nearest neighbor $\underline{z}'$ converges in probability to $\underline{z}$ with n, i.e.,

$$\underline{z}' \overset{n}{\rightarrow} \underline{z} \quad \text{in probability} .$$

Furthermore, the relationship between the probability of error for the 1-NN rule and the Bayes probability of error can be given as [24],

$$P_E^* \le P_E' \le P_E^* (2 - \frac{M}{M-1} P_E^*) , \tag{3.2}$$

for an M-class problem, where P_E^* is the Bayes probability of error and P_E' is the average probability of error for the 1-NN rule. For the two-class pattern recognition situation, $M=2$, we have,

$$P_E^* \le P_E' \le 2 P_E^* (1 - P_E^*) . \tag{3.3}$$

From (3.3), we notice that the 1-NN probability of error is bounded from above by twice the Bayes error rate. This, in a sense, shows that half of the classification information available in an infinite collection of classified samples is contained in the first nearest neighbor. This important result provided much of the incentive for the widespread interest in the nearest neighbor rules over the last decade.

3.1.2 Other Nearest Neighbor Rules

A natural extension of the single nearest neighbor rule is the k-NN rule. The k-NN rule consists of collecting k nearest neighbors of $\underline{z}$ in S_n and assigning $\underline{z}$ to the class to which the majority of its k-NN belong. As with the Bayes rule, we can safeguard ourselves against excessive classification error by resorting to the reject option. Under this mode of operation, a classification decision is made only if one of the classes receives a number of votes at least equal to a qualifying majority level l, otherwise the pattern is rejected. This rule is the $(k-l)$-NN rule. This last class of rules can still be generalized by letting the acceptance level depend on the decision to be made. In other words, we decide $\hat{x} = x_i$ if at least l_i-NN among k-NN to $\underline{z}$ are from class x_i. This rule is the $(k-l_i)$-NN rule.

It can be shown that the probability of error for the k-NN rule is a decreasing function of k and is upper bounded by twice the Bayes error rate and converges to the Bayes error rate as k grows arbitrarily large [25].

3.2 Condensing the Sample Set

The bounds given in the previous section for the probability of error of the nearest neighbor rules relies on the assumption of infinite sample size. However, in practice a large sample size makes the rule very demanding in terms of the needed storage space and computational complexity. To remedy the latter situation, one approach is to devise fast algorithms for nearest neighbor search [26]-[28]. While these algorithms reduce the computational complexity compared with the brute force search, the complexity still remains considerably high and the memory size is not altered or even increases. Another approach to the computational complexity and memory size reduction is the condensing of the sample set, i.e., only keeping a moderate number of points and discarding the rest [29], [30]. In this section, a new method for reducing the size of the sample set using a combination of vector quantization and an editing technique based on the evaluation of the performance of the sample points is presented. Later, we present several fast algorithms which can reduce the computational complexity of nearest neighbor search.

3.2.1 Vector Quantization

Vector Quantization (VQ) is a data compression method used in image and speech coding. It is a generalization of scalar quantization schemes such as PCM. While in scalar quantization each sample is treated independently, in vector quantization several consecutive samples of the source are encoded together as a vector. The reason for the increasing interest in VQ is the fact that vector encoding the source can result in lower distortion for a given rate. Vector quantization is not unknown to the pattern recognition community. In fact, it appears in another guise, namely, the K-means clustering algorithm in the pattern recognition literature [31] and is used mainly in unsupervised learning [32].

A k-dimensional vector quantizer Q of size N can be considered as a mapping from the k-dimensional vector space $\mathbf{R}^k$ into a finite subset $Y = \{\underline{y}_i \; ; \; i = 1 ,\ldots ,N\}$ [33]. The subset Y is called a codebook and its elements are called codewords or reproducing vectors. The codewords $\underline{y}_i$, $i = 1$, ... , N partition the space $\mathbf{R}^k$ into N regions S_i , $i = 1$, ... ,N, such that:

$$S_i \equiv \{\underline{\mathbf{z}} \mid Q(\underline{\mathbf{z}}) = \underline{\mathbf{y}}_i\} . \tag{3.4}$$

Therefore, a vector quantizer can be completely specified in terms of the codebook Y and the partition $S = \{S_i \ , i = 1 , \ldots ,N\}$.

3.2.2 Design Procedure for Vector Quantizer

The performance of the VQ is judged by the average distortion between its input and output. The distortion caused by quantizing $\underline{\mathbf{z}}$ into $Q(\underline{\mathbf{z}})$ is given by a non-negative function $d(\underline{\mathbf{z}} \ , Q(\underline{\mathbf{z}}))$,

$$d(\ \cdot\ ,\ \cdot\ ) : \mathbf{R}^k \times Y \to \mathbf{R} .$$

The optimal vector quantizer should minimize the average distortion $E[d(\underline{\mathbf{Z}} \ , Q(\underline{\mathbf{Z}}))]$. The most general formulation of the design procedure for a vector quantizer is given by Linde, Buzo, and Gray [34] and, therefore, is usually referred to as the LBG algorithm. The LBG algorithm is based on the following two necessary optimality properties and results in a locally optimal quantizer:

Property 1: For a given codebook, the best partition is the one based on the nearest neighbor rule, i.e.,

$$S_i \equiv \{\underline{\mathbf{z}} \mid d(\underline{\mathbf{z}} \ , \underline{\mathbf{y}}_i) \leq d(\underline{\mathbf{z}} \ , \underline{\mathbf{y}}_j) \quad \text{for all } j\} . \tag{3.5}$$

For the Euclidean or squared-error distortion measure, i.e.,

$$d(\underline{\mathbf{z}} \ , \underline{\mathbf{y}}_i) = \sum_{j=1}^{k} (y_{ij} - z_j)^2 \ , \tag{3.6}$$

the regions S_i described by (3.5) are called the Voronoi regions.

Property 2: For any partition $S = \{S_i\ ;\ i = 1\ , \ldots\ , N\}$ of $\mathbf{R}^k$, the average distortion will be minimized if the reproduction vector $\underline{\mathbf{y}}_i$ corresponding to S_i is taken to be $\hat{\underline{\mathbf{z}}}(S_i)$, defined as,

$$\hat{\underline{\mathbf{z}}}(S_i) = \min_{\underline{\mathbf{u}}}{}^{-1} E\,[d\,(\underline{\mathbf{z}}\ ,\ \underline{\mathbf{u}}\ |\ \underline{\mathbf{z}} \in S_i]\ . \tag{3.7}$$

Here $\hat{\underline{\mathbf{z}}}(S_i)$ is called the centroid of S_i. It can be shown that for the Euclidean distortion measure the centroid is given by,

$$\hat{\underline{\mathbf{z}}}(S_i) = E\,[\underline{\mathbf{z}}\ \mid\ \underline{\mathbf{z}} \in S_i]\ . \tag{3.8}$$

The LBG algorithm iteratively uses the above two properties in order to optimize the quantizer. Given a long sequence of training vectors and an initial codebook, it first encodes each vector into one of the codewords based on the nearest neighbor rule and then replaces each codeword with the centroid of all input vectors mapped into it. These two steps are repeated until the relative change in distortion falls below some prescribed, small threshold level.

3.2.3 Sample Set Condensing Using VQ

Vector quantization can be used for selecting a small number of prototypes or templates from a large set of feature vectors. To do so, we first design a codebook of size $N < n$ using the sample set S_n as training set. In the next stage, we label the codewords or templates found in the first stage by encoding the feature vectors and labeling each template with the label of the majority of feature vectors mapped into it.

It is obvious that not all of the templates have the same importance from a classification standpoint. Therefore, it is natural to evaluate the performance of the templates formed using vector quantization based on their contribution to the probability of error and discard those templates which are not essential for the classification purpose. The following editing algorithm is based on the assessment of the effect of presence or absence of each of the templates.

Denote the first and second nearest neighbors to a given feature vector $\underline{\mathbf{z}}$ among the codewords generated using VQ by $\underline{\mathbf{z}}'$ and $\underline{\mathbf{z}}''$. Let, x, x' and x'' denote the labels of $\underline{\mathbf{z}}$, $\underline{\mathbf{z}}'$ and $\underline{\mathbf{z}}''$, respectively. If $x' = x''$, then $\underline{\mathbf{z}}'$ does not have any effect on the classification of the particular feature vector $\underline{\mathbf{z}}$, because in this case $\underline{\mathbf{z}}''$ can classify $\underline{\mathbf{z}}$ as good (bad) as $\underline{\mathbf{z}}'$. Similarly, when $x' \neq x$ and $x'' \neq x$, the presence of $\underline{\mathbf{z}}'$ does not have any effect on the classification of $\underline{\mathbf{z}}$. But, if $x' = x \neq x''$, then the presence of $\underline{\mathbf{z}}'$ prevents an error. On the other hand, when $x' \neq x = x''$, then the presence of $\underline{\mathbf{z}}'$ results in an error which had otherwise been avoided. Based on these observations, we propose the following editing algorithm for selecting $N_c < N$ templates from N templates generated using vector quantization. This algorithm, in each iteration, calculates a function $f(\underline{\hat{\mathbf{z}}}_j)$ for each template $\underline{\hat{\mathbf{z}}}_j$, and rejects the $\underline{\hat{\mathbf{z}}}_j$ with the smallest value of $f(\underline{\hat{\mathbf{z}}}_j)$. The function $f(.)$ measures the role of each template in reducing the probability of classification error.

Step 0 : $N' \leftarrow N$

Step 1 : $i \leftarrow 1$

Step 2 : Encode $\underline{\mathbf{z}}_i$ using codewords $\underline{\hat{\mathbf{z}}}_1, \ldots, \underline{\hat{\mathbf{z}}}_{N'}$ to find the first and second nearest neighbors $\underline{\mathbf{z}}'$ and $\underline{\mathbf{z}}''$.

Step 3 : Let

$$f(\underline{\mathbf{z}}') = \begin{cases} f(\underline{\mathbf{z}}') + 1, & \text{if} \quad x' = x \neq x'', \\ f(\underline{\mathbf{z}}') - 1, & \text{if} \quad x' \neq x = x'', \\ f(\underline{\mathbf{z}}'); & \text{otherwise} \end{cases} \tag{3.9}$$

```
Step 4 : $i \leftarrow i+1$ if $i \leq n$ go to 2.

Step 5 : Edit (discard) the codeword $\hat{\underline{z}}_j$ with the smallest $f(\hat{\underline{z}}_j)$,

        $N' \leftarrow N' - 1$.

Step 6 : If $N' > N_c$ go to 1. Else, stop.
```

3.2.4 An Example of the Application of the Condensing Algorithm.

As an example, consider the two class pattern recognition problem with equal a priori probabilities, i.e., $P(x_1) = P(x_2) = \frac{1}{2}$. Also, assume that the feature vectors are four-dimensional real vectors with class conditional probability densities $p(\underline{z} \mid x_i) = \mathrm{N}(\underline{\mu}_i\ , \Sigma_i)$, $i=1, 2$, where, $\underline{\mu}_1 = (\ 1\ , 1\ , 1\ , 1\ )^T$ and $\underline{\mu}_2 = (\ -1\ , -1\ , -1\ , -1\ )^T$ and,

$$\Sigma_i = \begin{bmatrix} 1 & \rho_i & \rho_i^{\,2} & \rho_i^{\,3} \\ \rho_i & 1 & \rho_i & \rho_i^{\,2} \\ \rho_i^{\,2} & \rho_i & 1 & \rho_i \\ \rho_i^{\,3} & \rho_i^{\,2} & \rho_i & 1 \end{bmatrix}, \qquad i=1, 2. \tag{3.10}$$

That is, the feature vectors are first order Gauss-Markov random variables. Let $\rho_1 = 0.3$ and $\rho_2 = 0.7$. The Bayes probability of error for this example is around 0.0718. [See Appendix 3.A.]

To apply the procedure discussed above to this example, first, using the LBG algorithm, an $N=16$ point vector quantizer was designed for a sample set of size $n=10000$ containing an equal number of feature vectors from each class. The probability of error for the 16 templates was found to be 0.0786 which is close to the Bayes probability of error. Then, each codeword was labeled with the label of the majority of feature vectors falling into its Voronoi region. Finally, using the proposed editing algorithm, $N' < 16$ templates were selected from the 16 templates originally generated.

Table 3.1 shows the probability of error for $N' = 4, \ldots, 9$ for the feature vectors inside and outside the sample set. The values in the third column of Table 3.1 are the result of averaging the probability of error for several data sets generated according to the same distribution, but with different seeds. It can be seen from Table 3.1 that the first seven templates discarded had no overall effect on the probability of classification error.

Number of Templates	Probability of Error	
	in	out
9	0.0786	0.0793
8	0.0787	0.0795
7	0.0789	0.0802
6	0.0797	0.0813
5	0.0831	0.0852
4	0.0859	0.0874

Table 3.1 : Probability of classification error versus the number of templates.

3.3 Fast Nearest Neighbor Search Algorithms

The nearest neighbor search problem consists of finding the point closest to a query point among N points in k-dimensional space, $\mathbf{R}^k$, where N is equal to the total number of points in the sample set if no condensing is performed or is equal to the number of templates formed using a condensing algorithm such as the one discussed in the previous section. In one dimension, the problem can be easily solved in logarithmic time using a preliminary sorting. Sorting does not, however, generalize to higher dimensions and, in general, the computational complexity is a linear function of N [35]. To find the nearest neighbor to a new feature vector, we need to find its distance from each of N sample points, and then compare these distances in order to find the closest point. Therefore, for each vector, N distance computations and $N - 1$ comparisons are required. In the case of the Euclidean "distance" of (3.6), each distance calculation requires k multiplications and $2k - 1$ additions (subtractions). Thus, to classify each new pattern, $k \times N$ multiplications, $(2k - 1)N$ additions, and $N - 1$ comparisons need to be performed. The computational complexity can alternatively be expressed in terms of N multiplications, $(2 - \frac{1}{k})N$ additions, and $\frac{N-1}{k}$ comparisons per feature.

In the next three sections several new fast search algorithms are presented which can considerably reduce the number of required operations in comparison with those indicated above by performing appropriate tests <u>prior</u> to distance calculation for each template, thereby avoiding distance calculation for those templates which fail these tests. These algorithms can be useful in nearest neighbor search, in general, and in the case of condensed sample set, in particular. These algorithms can also be used most conveniently for reducing the complexity of speech and image coders based on vector quantization. In the latter case, the algorithms can be used both in the design and the encoding stages of a vector quantizer.

3.4 The Hypercube Algorithm

In the algorithm proposed here, the complexity is reduced by performing a simple test before computing the distance for each template (codeword) and rejecting those templates which fail the test[36]. The test consists of comparing the absolute error associated with each component of the codevector $\underline{\mathbf{y}}_i$, $| z_j - y_{ij} |$, with the square-root of the minimum distortion found thus far, where z_j and y_{ij} denote the jth component of the feature vector $\underline{\mathbf{z}}$ and the template $\underline{\mathbf{y}}_i$, respectively. This test is equivalent to first checking if a given template is inside a *hypercube* enscribing the hypersphere centered around the feature vector, with radius equal to the square-root of the minimum distortion found thus far, and if so, then performing checking inside the hypersphere itself. Eliminating the need to compute the distance for the rejected templates, this test results in a drastic reduction in the number of multiplications and additions. The number of subtractions is also considerably reduced. The number of comparisons increases, however, still there is a considerable saving in terms of the total number of required operations. For those templates which fail the test, we save k multiplications, $k-1$ additions, and the remaining subtractions. Denoting the average number of rejected codewords by N', the average number of required multiplications per sample is $N_m = N - N'$ and the average number of additions is $N_a = (1 - \frac{1}{k})(N - N')$. The average number of subtractions required is $N_s = (N - N') + N' \times \frac{\overline{k}}{k}$, where $\overline{k}$ is the average number of components checked before a codeword is rejected. Since we perform one comparison after each subtraction and one after each set of k multiplications, the number of comparisons is,

$$N_c = N_s + \frac{N_m}{k} = (N - N') + N' \times \frac{\bar{k}}{k} + \frac{N - N'}{k}$$

The square-root introduced in this test is an infrequent operation which does not have a significant effect on the overall complexity. Furthermore, it can be performed using either a look-up table or a crude approximation.

3.4.1 Simulation Results for Hypercube Algorithm

We used a sample set consisting of 5000 k-dimensional vectors drawn from a first order Gauss-Markov process having a correlation coefficient of $\rho = 0.9$, and using the LBG algorithm, with a stopping threshold ϵ=0.05, we designed a codebook containing $N = 2^k$ templates. In terms of data compression terminology, this corresponds to a rate of one bit/sample, since the rate r is related to N and k by $r = \frac{\log_2 N}{k}$ bits/sample.

Dimension	Conventional Method				Hypercube Method				Required ×	Total operations
k	×	+	-	comp	×	+	-	comp	%	%
3	8	5 33	8	2 33	3 92	2 61	5 60	6 90	49	80 43
4	16	12	16	3 75	4 90	3 68	8 96	10.19	30 62	58.07
5	32	25 60	32	6 2	6 68	5 35	14 88	16 22	16 62	45 02
6	64	53 33	64	10 50	9 80	8 17	25 49	27 12	12 76	36 79
7	128	109 71	128	18 14	14.38	12 32	45 71	47.76	11 23	31 30
8	256	224	256	31 87	23 35	20 43	84 83	87 70	9 12	28 17

Table 3.2 : Comparison of the complexity of the hypercube algorithm to the conventional algorithm.

Table 3.2 compares the number of multiplications, additions, subtractions, and comparisons per sample (feature) for the new algorithm and the conventional full search algorithm, for dimensions 3 to 8. From Table 3.2, it can be seen that for large k, the number of multiplications is reduced to as low as 9% and the total number of operations is reduced to 28%, both referenced to the corresponding values for the full search method.

Using the partial distortion calculation method proposed by Cheng et al. [37] and Da Bel and Gray [38], while computing the distortion for those codewords which pass the hypercube test, can further reduce the number of multiplications and additions, at the price of a further increase in the number of comparisons. Here, instead of computing the complete distortion for a codeword $\underline{\mathbf{y}}_i$, i.e., $\sum_{j=1}^{k} (y_{ij} - z_j)^2$, and then comparing it with the minimum distortion found thus far, a comparison is performed after adding each term, and if for any $l < k$, the partial distortion $\sum_{j=1}^{l} (y_{ij} - z_j)^2$ exceeds the previous minimum distortion, the codeword is rejected without completing the distortion calculation.

ρ	HC1				HC2			
	×	+	-	comp.	×	+	-	comp.
0.9	23.35	20.43	84.83	87.70	16.33	13.98	84.83	98.81
0.5	39.25	34.33	117.98	122.89	26.67	21.97	117.98	139.95
0.0	50.82	44.47	131.46	137.77	31.02	24.39	131.46	155.85

Table 3.3 : The effect of inclusion of partial distortion method,

$k = 8, N = 256.$

Table 3.3 compares the performance of the new method with (HC2) or without (HC1) the partial distortion calculation, for $k = 8$ and cases of highly correlated ($\rho = 0.9$) and weakly correlated ($\rho = 0.5$) Gauss-Markov and memoryless ($\rho = 0.$) Gaussian sources. Table 3.3 shows that the effect of inclusion of the partial distortion method is more profound for less correlated sources.

3.4.2 Comparison with Other Methods

Table 3.4 compares the performance of the new algorithm with several methods found in the literature. Since the algorithm described here does not require any precomputations and/or extra memory, we have limited the comparison to those methods which require none or little precomputation and extra memory. The methods used for comparison are the partial distortion method of Da Bel and Gray [38], the minimax method of Cheng et al. (with step 5′) [37],

Method	$\rho = 0.9$				$\rho = 0.5$			
	×	+	-	comp.	×	+	-	comp.
HC1	23.35	20.43	84.83	87[illegible]	39.25	34.34	117.98	122.89
HC2	16.33	13.98	84.83	98.81	26.67	21.97	117.98	139.95
Da Bel	69.39	37.39	69.39	69.39	93.49	61.49	93.49	93.49
Minimax	3.83	2.99	256	308.87	7.25	5.43	256	404.99
k-d tree	45.28	45 28	69.30	103.84	102.70	102.70	145.34	213.71

Table 3.4 : Comparison of performance of the proposed algorithm to other algorithms, $k = 8$, $N = 256$.

and the k-d tree of Friedman et al. [27]. The comparison is done for dimension $k = 8$ and strongly correlated ($\rho = 0.9$) and weakly correlated ($\rho = 0.5$) Gauss-Markov sources.

Table 3.4 shows that in terms of the number of multiplications, the given method is only inferior to the minimax method. In terms of the total number of operations, the new algorithm outperforms all of the above methods.

3.5 Improved Hypercube Method

The method discussed above can be made even more efficient, especially in terms of reduction in the number of required multiplications, by first finding a tentative match for the input vector and then, confining the search to a small area around that tentative match [39]. The algorithm discussed here, consists of two phases. In the first phase of the algorithm a tentative match for the input vector is found using a procedure called the shrinking hypercube method. The result of this method is the same as that of the minimax method of [37], i.e., it finds the codevector which minimizes the l_∞ norm of $\underline{z} - \underline{y}_i$, but requires far less subtractions and comparisons. In the second phase of the algorithm, other templates are checked against the tentative match found in the first phase using the hypercube test previously discussed.

In the shrinking hypercube method, we start from a hypercube centered around the input vector. The hypercube is assumed large enough to ensure that at least one codeword is contained inside it. We then search through the codebook for a template lying within this hypercube. When the first such template, say $\mathbf{y}_i$, is found, we reduce the side of the hypercube to $2r_m$, where,

$$r_m = \max_j | z_j - y_{ij} |$$

It is clear that this is a smaller hypercube than the previous one and also it does not contain any of the previously rejected templates. To avoid the need for searching the entire codebook in the second phase of the algorithm, we can keep the indices of the rejected templates in separate lists, depending on the value of their last error magnitude checked before being rejected in the first phase. This procedure is continued until we find a hypercube with <u>no</u> templates inside, but with one template on one of its sides. We take this template as a tentative "best" match and confine our search to inside a hypersphere with radius equal to the square-root of the distortion due to this tentative template. Using the hypercube test discussed before, we then search for a possibly better match among the templates in the list(s) with corresponding value(s) less than the square-root of the distortion between the tentative codeword and the input vector.

3.5.1 Simulation Results for the Improved Algorithm

A discrete Gauss-Markov source having a correlation coefficient of $\rho = 0.9$ is used. In each case, we have used a training set consisting of 5000 vectors. The codebook size is $N = 2^k$. Table 3.5 compares the number of multiplications, additions (plus subtractions), and comparisons per sample for the new algorithm and the conventional full search algorithm, for the dimensions 3 to 8. Table 3.5 shows that using the improved method, the number of multiplications required can be reduced to as low as 2.67% and the total number of operations can be as low as 24.6%, both compared to the full search method.

Dimension	Full Search Method			Improved Method			Required	Total
	×	+	comp.	×	+	comp	× %	operations %
3	8	13.33	2.33	1.22	5.25	8.30	15.25	62.43
4	16	28	3.75	1.49	8.38	12.63	9.31	47.12
5	32	57.6	6.2	2.16	14.04	20.49	6.75	38.30
6	64	117.33	10.5	3.39	23.95	34.40	5.30	32.18
7	128	237.71	18.14	5.32	41.22	59.42	4.16	27.60
8	256	480	31.87	6.83	77.02	105.07	2.67	24.60

Table 3.5 : Comparison of the complexity of the improved algorithm to the full search algorithm.

3.5.2 The Hyperpyramid Test

While the hypercube test rejects the majority of unsuitable templates, for higher dimensions and larger codebooks there is still a fairly high probability that a template passing this test does not lie inside the hypersphere inscribed by the hypercube. For this reason, an extra test based on the following inequality may prove useful in further reducing the number of multiplications. However, this reduction in the number of required multiplications is at the price of a further increase in the number of both comparisons and additions. Therefore, use of this test is only justified if the cost of multiplication is considered to be much more expensive than that of comparison and addition for the given operating environment and implementation.

The Hyperpyramid Inequality:

$$\text{If} \quad \sum_{j=1}^{k} (z_j - y_{ij})^2 < d^2 , \qquad \text{then} \quad \sum_{j=1}^{k} | z_j - y_{ij} | < d\sqrt{k} .$$

The proof of the inequality may be found in Appendix 3.B. The name of the inequality is due to the fact that $\sum_{j=1}^{k} | z_j - y_{ij} | = d\sqrt{k}$ describes the surface of a hyperpyramid in k-dimensional space [40].

Now, let the minimum distortion found before testing the template $\mathbf{y}_i$ be d^2. After ensuring that $| z_j - y_{ij} | < d$, $j = 1,...,k$, we start adding up the absolute errors, and after each addition compare the accumulated absolute error with $d\sqrt{k}$. If it is greater, we reject that template. Table 3.6 shows the effect of the inclusion of the hyperpyramid test for dimensions k=6,7,8 for the same Gauss-Markov source.

Dimension	×	+	comp.
6	1.62	24.18	35.25
7	1.86	42.79	61.39
8	2.13	77.83	109.65

Table 3.6 : The effect of inclusion of the hyperpyramid test.

3.6 Voronoi Region Algorithm

An important characteristic of the algorithms discussed so far is the fact that they do not require any precomputations or extra memory. In this section, we present a method which requires a small amount of precomputation and extra storage space. However, for this algorithm, the increase in the number of comparisons for a given reduction in the number of multiplications is less than that of previously discussed algorithms, and, therefore, it is more suitable in the case where all three basic operations are considered to be of almost the same complexity. This is in fact the case with some of the new general purpose digital signal processors such as the TMS32020. Furthermore, this algorithm has a simple encoding procedure which makes it appealing for microprocessor implementation of real-time speech coders and speech recognizers. The algorithm is based on the one-to-one correspondence between the codevectors and the Voronoi regions associated with them [42].

The templates $\underline{\mathbf{y}}_i$, $i=1, \cdots, N$, partition the the space $\mathbf{R}^k$ into N regions S_i , $i=1, \cdots, N$, such that:

$$S_i = \left\{ \underline{\mathbf{z}} : \| \underline{\mathbf{z}} - \underline{\mathbf{y}}_i \|^2 \leq \| \underline{\mathbf{z}} - \underline{\mathbf{y}}_j \|^2 , \quad \text{all } j \right\} . \tag{3.11}$$

There is a one-to-one correspondence between the templates and the Voronoi regions, i.e., if an input vector $\underline{\mathbf{z}}$ is contained in S_i, then the template nearest to it is $\underline{\mathbf{y}}_i$. Conversely, if $\underline{\mathbf{z}}$ does not lie inside S_i, evidently $\underline{\mathbf{y}}_i$ can not be its nearest neighbor. We use this negative aspect of the correspondence in order to reject a large number of codewords from consideration without calculating their distance from $\underline{\mathbf{z}}$.

For each template $\underline{\mathbf{y}}_i$, let $r_i = \sqrt{d_i}$, where,

$$d_i = \max_{\underline{\mathbf{z}} \in S_i} \| \underline{\mathbf{z}} - \underline{\mathbf{y}}_i \|^2 . \tag{3.12}$$

Now, for a given input vector $\underline{\mathbf{z}}$ if,

$$| z_j - y_{ij} | > r_i , \tag{3.13}$$

for some $j \in \{1, ..., k\}$, then we can be sure that $\underline{\mathbf{y}}_i$ is not the nearest codevector to $\underline{\mathbf{z}}$.

Also notice that if the smallest distortion found before checking $\underline{\mathbf{y}}_i$ is d, then $\underline{\mathbf{y}}_i$ cannot be a better match than the previous "best" match if,

$$| z_j - y_{ij} | > \sqrt{d} \qquad (3.14)$$

Therefore, while encoding an input vector $\underline{\mathbf{z}}$, we can reject the codeword $\underline{\mathbf{y}}_i$ if,

$$| x_j - y_{ij} | > r_i' \quad , \qquad (3.15)$$

where $r_i' = \min(r_i \ , \sqrt{d}\ )$.

3.6.1 Required Precomputation and Extra Memory

Using a sufficiently long training sequence, we determine d_i's and, therefore, r_i's. In order to make sure that the algorithm performs well for vectors outside the training sequence, we can add some margin to the r_i's. Then, we sort the r_i's in ascending order and sort the codebook accordingly. This sorting is performed in order to avoid the comparison to find the minimum of r_i and $\sqrt{d}$ after a candidate template is found. That is, after finding some codevector $\underline{\mathbf{y}}_l$ such that,

$$|| \underline{\mathbf{z}} - \underline{\mathbf{y}}_l ||^2 = d \leq r_l^2 \ , \qquad (3.16)$$

we have,

$$\sqrt{d} \leq r_l \leq r_i \quad \text{for all } i > l \ , \qquad (3.17)$$

and, therefore,

$$r_i' = \min(r_i \ , \sqrt{d}\ ) = \sqrt{d} \ . \qquad (3.18)$$

Thus, there is no need for fetching r_i and comparing it with $\sqrt{d}$, for $i > l$.

The complexity of finding d_i's is the same as one iteration of the design procedure plus one comparison per training vector. This can be performed during the last stages of the design procedure, and will add one comparison per training vector per iteration to the overall computation required for designing the codebook. N square-root operations are required to find the r_i's when d_i's are found. The sorting of the codebook can be performed by $O(N\log N)$ comparisons, using the mergesort algorithm [41]. We need an extra memory area to store N scalars, i.e., the r_i's. Since the memory required to store the codebook itself is kN, the algorithm increases the memory requirement by a factor of $\frac{1}{k}$.

3.6.2 The Encoding Procedure

To encode an input vector $\underline{\mathbf{z}}$, we start from the first codeword, i.e., the one with the smallest r_i, and search for a codeword for which $|z_j - y_{ij}| \le r_i$ holds for all $j \in \{1, \ldots, k\}$. After finding such a codevector, we start calculating its distance with $\underline{\mathbf{z}}$ using the "partial distance" method. This means that if for some $l < k$ the partial distance $\sum_{j=1}^{l} (y_{ij} - z_j)^2$ exceeds d_i, we reject $\underline{\mathbf{y}}_i$ without completing the distortion calculation. If $\sum_{j=1}^{k} (y_{ij} - z_j)^2 \le d_i$, we take $\underline{\mathbf{y}}_i$ as a tentative match, and continue the search for finding a possibly better match among the remaining codevectors. However, for the remaining codewords, instead of fetching r_i, we use $\sqrt{d}$ for comparison, where d is the smallest distortion found thus far. A Fortran program describing the encoding procedure may be found in Appendix 3.C.

3.6.3 An Alternative to r_i

Instead of r_i, we can use t_i defined as,

$$t_i = \max_{\underline{\mathbf{z}} \in S_i} \max_j |y_{ij} - z_j| \qquad (3.19)$$

In other words, t_i is the maximum error magnitude for input vectors in S_i. Since the error due to each component is normally less than the square-root of the total distortion, using t_i instead of r_i may result in a more efficient algorithm. However, the precomputation required for finding t_i is more than that for r_i. In fact, we require k comparisons per training vector, i.e., $k-1$ comparisons for finding the largest component error of each training vector and one comparison to compare it with the previous t_i.

The encoding procedure is similar to that of the previous case, except for the fact that we need to fetch t_i's until we find a codevector whose distortion is less than t_i^2. The required modifications for this case are included in the program in Appendix 3.C.

3.6.4 Simulation Results for Voronoi Region Algorithm

A discrete Gauss-Markov source having a correlation coefficient of $\rho = 0.9$ is used. Table 3.7 compares the number of multiplications, additions, and comparisons per sample for the algorithm, when r_i is used, and the conventional full search algorithm, for the dimensions 4 to 8. In each case, we have used a training set consistifig of 5000 vectors. The number of templates is $N = 2^k$. Table 3.8 shows the corresponding values for the case where t_i is used. The entries of the last two columns indicate that, for large codebooks, by using the new method the number of required multiplications can be reduced to as low as 2.23% of those required by the conventional full search method, while the total number of operations can be reduced to as low as 22%. The tables also show that using t_i results in lower complexity, but the difference is not considerable.

Tables 3.9 and 3.10 demonstrate the application of this method to speech samples. In each case, 262,656 speech samples extracted from the Texas Instruments connected digit database are used.

Dimension	Full Search Method			New Method			% required	% required
	×	+	comp	×	+	comp	×	operations
4	16	28	3 75	3 06	9 41	9.41	19 13	45 82
5	32	57 6	6 2	4 05	16 20	16 20	12 66	38.05
6	64	117 33	10.5	5 21	26 74	26 74	8 14	30.59
7	128	237 71	18 14	6 56	45 91	45 91	5 13	25 63
8	256	480	31 87	8 65	80 54	80 54	3 38	22.10

Table 3.7 : Comparison of the complexity of the new algorithm (r_i used) to the full search algorithm.

(Gauss-Markov source)

Dimension	Full Search Method			New Method			% required	% required
	×	+	comp	×	+	comp	×	operations
4	16	28	3.75	2 34	9 01	9 36	14.63	43.37
5	32	57 6	6 2	3 28	15 58	15 95	10 25	36 34
6	64	117 33	10.5	3 83	25 50	25.83	5.98	28 75
7	128	237 71	18 14	4 55	44 00	44 31	3 55	24.19
8	256	480	31 87	5 70	79 32	79 60	2 23	21.44

Table 3.8 : Complexity of the new algorithm ,when t_i is used, versus the full search method.

(Gauss-Markov source)

Dimension	Full Search Method			New Method			% required	% required
	×	+	comp	×	+	comp	×	operations
4	16	28	3.75	2 07	7 08	7 08	12 93	45 82
5	32	57 6	6.2	3 23	12 49	12 49	10 09	29 45
6	64	117 33	10 5	3.94	19 66	19 66	6.16	22.55
7	128	237 71	18 14	6 25	35 82	35 82	4 88	20 29
8	256	480	31 87	7 77	60.81	60 81	3 04	16 85

Table 3.9 : Comparison of the complexity of the new algorithm (r_i used) to the full search algorithm.

(Speech samples)

Dimension	Full Search Method			New Method			% required	% required
	×	+	comp	×	+	comp	×	operations
4	16	28	3 75	1 74	7 01	7 29	10 88	33 59
5	32	57 6	6 2	2 82	12 26	12.51	8 81	28 80
6	64	117 33	10 5	3 25	18 90	19 12	5.08	21 51
7	128	237 71	18 14	4 55	33.14	33 38	3 55	18.52
8	256	480	31 87	6 03	58 08	58 32	2 36	15.94

Table 3.10 : Complexity of the new algorithm ,when t_i is used, versus the full search method.

(Speech samples)

3.7 Discussion

In this chapter, different nearest neighbor classification rules were discussed. However, main attention was placed on the single nearest neighbor rule. In order to overcome the computational complexity problem, a new condensing algorithm for sample size reduction, based on the application of K-means clustering algorithm or vector quantization, was presented and a typical example was worked out. Also, several new fast nearest search algorithms were proposed which can considerably reduce the search time. These fast algorithms may prove specially useful in vector recognition discussed in Chapter 6. In this case, several consecutive feature vectors are treated at the same time; therefore, the complexity of searching for the nearest neighbor is a more serious consideration.

Appendices

3.A Probability of Error for the Illustrative Example

Since the covariance matrices Σ_1 and Σ_2 are real, symmetric, and positive definite, a matrix R can be found such that,

$$R^T \Sigma_1 R = I \quad \text{and} \quad R^T \Sigma_2 R = \Lambda$$

where, Λ is a diagonal matrix with diagonal elements $\lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \lambda_4$ which are the roots of the equation $| \Sigma_2 - \lambda \Sigma_1 | = 0$, and the ith column of R is the eigenvector of the pair (Σ_1 , Σ_2) corresponding to λ_i. For the values of $\rho_1 = 0.3$ and $\rho_2 = 0.7$, we have,

$$R = \begin{bmatrix} 0.362 & -0.606 & -0.647 & -0.427 \\ -0.846 & 0.539 & -0.218 & -0.379 \\ 0.846 & 0.539 & 0.218 & -0.379 \\ -0.362 & -0.606 & 0.647 & -0.427 \end{bmatrix}$$

and

$$\Lambda = \begin{bmatrix} 0.344 & 0 & 0 & 0 \\ 0 & 0.419 & 0 & 0 \\ 0 & 0 & 0.697 & 0 \\ 0 & 0 & 0 & 1.748 \end{bmatrix}$$

Performing the change of variable $\underline{y} = R^T(\underline{z} - \underline{\mu}_1)$, we have, $p(\underline{y} \mid x_1) = \mathrm{N}(0 , I)$ and $p(\underline{y} \mid x_2) = \mathrm{N}(\nu , \Lambda)$ where,

$$\nu = R^T(\underline{\mu}^2 - \underline{\mu}_1) \doteq (0 , 0.268 , 0 , 3.224)$$

Now, using (2.51) and (2.52) it is easy to show that,

$$P_E^* = \frac{1}{2} Pr\left[\sum_{i=1}^{4}\{y_i^2 - \frac{(y_i - \nu_i)^2}{\lambda_i}\} > \sum_{i=1}^{4}\log\lambda_i\right]$$

$$+ \frac{1}{2} Pr\left[\sum_{i=1}^{4}\{(y_i\sqrt{\lambda_i} + \nu_i)^2 - y_i^2\} < \sum_{i=1}^{4}\log\lambda_i\right]$$

where the y_i's are i.i.d. $N(0, 1)$ random variables. A value of $P_E^* \approx 0.0718$ was found using a Monte Carlo approach with a sample size of 80000.

3.B Proof of the Hyperpyramid Inequality

Proof of the Hyperpyramid inequality :

Letting $w_j = | z_j - y_{ij} |$, $j = 1,\ldots, k$, we can write the inequality as follows :

$$\text{If for } w_j \geq 0, \quad \sum_{j=1}^{k} w_j^2 < d^2 , \quad \text{then } \sum_{j=1}^{k} w_j < d\sqrt{k}$$

Proof. We prove the inequality using induction. For $k = 1$, it is trivial. Therefore, it is enough to show that if it is true for $k - 1$, it is also true for k, $k > 1$.

Now, assume that the inequality is true for $k - 1$, i.e.,

$$\sum_{j=1}^{k-1} w_j^2 < d^2 \Rightarrow \sum_{j=1}^{k-1} w_j < d\sqrt{k-1}$$

Then,

$$\sum_{j=1}^{k} w_j^2 < d^2 \Rightarrow \sum_{j=1}^{k-1} w_j^2 < d^2 - w_k^2$$

$$\Rightarrow \sum_{j=1}^{k-1} w_j < \sqrt{(k-1)(d^2 - w_k^2)}$$

$$\Rightarrow \sum_{j=1}^{k} w_j < \sqrt{(k-1)(d^2 - w_k^2)} + w_k$$

The only thing that we need to prove is ;

$$\sqrt{(k-1)(d^2 - w_k^2)} + w_k < d\sqrt{k}$$

Write

$$f\ (w_k) = \sqrt{(k - 1)(d^2 - w_k{}^2)} + w_k$$

For $w_k \in [0,\ d]$, this function has a maximum,

$$f_{\max}(w_k) = \frac{(2k - 1)}{\sqrt{4k - 3}}d$$

at $w_k = \dfrac{d}{\sqrt{4k - 3}}$

So,

$$\sqrt{(k - 1)(d^2 - w_k{}^2)} + w_k \le \frac{(2k - 1)}{\sqrt{4k - 3}}d < \sqrt{k}\,d$$

the last inequality is true for all $k > 1$.

3.C Encoding Procedure for Voronoi Region Algorithm

In this appendix, we give the Fortran program describing the encoding procedure, for the case where r_i is used. Unnecessary details are omitted for the sake of brevity. The necessary changes for the case of t_i are added as comments.

```
      program fast
       read, (z(j), j=1,k)
c     When t_i is used, the following
c     line should be added.
c     dm=∞
      do 25  i=1,N
      do 15  j=1,k
      w(j)= abs(y(i,j)-z(j))
      if (w(j).gt.r(i)) goto 25
15     continue
c     When t_i is used, the following
c     line is not necessary
      dm=r(i)**2.
      d=w(1)**2.
      do 20  j=2,k
      d=d+w(j)**2.
      if (d.gt.dm) goto 25
20     continue
      dm=d
      t=sqrt(dm)
      m=i
      goto 26
c     When $t_i$ is used, the above line
c     should be replaced by the following line.
c      if (t.le.r(i)) goto 26
25     continue
26     do 40 i=m+1,N
      do 30 j=1,k
      w(j)=abs(y(i,j)-z(j))
      if (w(j).gt.t) goto 40
30     continue
      d=w(1)**2.
      do 35 j=2,k
      d=d+w(j)**2.
      if (d.gt.dm) goto 40
35     continue
      dm=d
      t=sqrt(dm)
      m=i
40     continue
      print, m
      stop
      end
```

CHAPTER 4

The Rate-Distortion Theoretic Model for Pattern Recognition

In this chapter, we present a model for the pattern recognition process, based on the generalized communication system model proposed by Dobrushin and Tsybakov[13]. We begin this chapter with an introduction to rate-distortion theory and its generalization. After this introduction, the model for the pattern recognition problem is discussed.

4.1 Rate-Distortion Theory

Rate-distortion theory is the mathematical theory of data compression. For any statistically modeled source and a given distortion measure, it establishes the relationship between the minimum number of bits per source sample required to encode the source and the average distortion after decoding.

The classical communications system model of Shannon[43], [44] is shown in Figure 4.1. The designer does not generally have any control over the source, user, and channel but is usually free to construct the encoders and decoders. The channel coding theorem [43], [15], states that channel encoders and channel decoders can be found which ensure an arbitrarily small error probability for messages transmitted through the channel encoder, channel, and channel decoder, as long as the message rate does not exceed the capacity of the channel. For this reason, in the development of rate-distortion theory, it is assumed that the transmission between the source encoder and source decoder is noiseless. This assumption results in the source coding model shown in Figure 4.2. This model can also be adopted for the storage of data in a computer, where the capacity of the noiseless channel corresponds to the limited amount of memory allowed per source symbol [45]. In pattern recognition applications, the capacity of the noiseless channel may represent the storage required to store the decision rules [4].

4.1.1 Definition of the Rate-Distortion Function

For a discrete memoryless source, the rate-distortion function is defined as follows. Assume a discrete memoryless source with output X taking values x with probabilities $P(x)$, where $x \in \{1,\ldots,M\}$. The set $A_M = \{1,\ldots,M\}$ is called the source alphabet. Denote the reproducing alphabet, i.e., the set of possible values of $\hat{x}$, by $A_N = \{1,\ldots,N\}$. To evaluate the quality of reconstruction of the source output by the encoder-decoder pair we need to define a fidelity criterion. Let $\rho(x,\hat{x})$ be the distortion caused when the source output is x and the output of decoder is $\hat{x}$. The quality of the reproduction can be judged by the ensemble average of $\rho(x,\hat{x})$ over the joint probability distribution for x and $\hat{x}$, i.e.,

$$d(X,\hat{X}) = E\left\{\rho(X,\hat{X})\right\} = \sum_x \sum_{\hat{x}} P(x) Q(\hat{x} \mid x) \rho(x,\hat{x}) , \tag{4.1}$$

where $Q(\hat{x} \mid x)$ is the conditional probability distribution of $\hat{x}$ given $X = x$. For a given distortion D, we define the set of admissible conditional distributions of $\hat{x}$ given x as,

$$Q_D = \left\{ Q(\hat{x} \mid x) : E\{\rho(X,\hat{X})\} \leq D \right\} . \tag{4.2}$$

Then the rate-distortion function is defined as follows,

$$R(D) := \min_{Q(\hat{x} \mid x) \in Q_D} I(X;\hat{X}) , \tag{4.3}$$

where $I(X;\hat{X})$ is the average mutual information between X and $\hat{X}$ given by,

$$I(X;\hat{X}) = \sum_x \sum_{\hat{x}} P(x) Q(\hat{x} \mid x) \log \frac{Q(\hat{x} \mid x)}{Q(\hat{x})} . \tag{4.4}$$

The rate-distortion function R(D) can be found by minimizing $I(X;\hat{X})$ of (4.4) subject to the following constraints:

$$Q(\hat{x} \mid x) \geq 0 , \tag{4.5}$$

$$\sum_{\hat{x}} Q(\hat{x} \mid x) = 1 , \tag{4.6}$$

and,

$$\sum_x \sum_{\hat{x}} P(x) Q(\hat{x} \mid x) \rho(x, \hat{x}) \leq D . \tag{4.7}$$

Constraints (4.5) and (4.6) are needed due to the fact that $Q(\hat{x} \mid x)$ is a conditional probability which should be non-negative and sum to one for each given x. The constraint (4.7) is merely the average distortion condition of (4.2). This minimization can be performed using the method of Lagrange multipliers.

Before stating the results of such a minimization, we consider two important quantities, $D_{\min}$ and $D_{\max}$. The quantity $D_{\min}$ denotes the minimum average distortion achievable. From (4.7) it is seen that the minimum possible value of the average distortion is found by setting $Q(\hat{x} \mid x) = 1$ for $\hat{x}$ which minimizes $\rho(x, \hat{x})$, i.e.,

$$D_{\min} = \sum_x P(x) \rho(x) , \tag{4.8}$$

where,

$$\rho(x) = \min_{\hat{x}} \rho(x, \hat{x}) . \qquad (4.9)$$

If $\rho(x,\hat{x}) = 0$ for at least one $\hat{x}$ for each x, then $D_{\min} = 0$. Otherwise, we can use the modified distortion measure $\tilde{\rho}(x,\hat{x}) = \rho(x,\hat{x}) - \rho(x)$. Then, the relation between the rate-distortion function of the source with respect to $\tilde{\rho}(x,\hat{x})$, say $\tilde{R}(D)$, and the original rate-distortion function, $R(D)$, will then be $R(D) = \tilde{R}(D - D_{\min})$.

$D_{\max}$ is the least average distortion achievable when $I(X;\hat{X}) = 0$, i.e., $D_{\max}$ is the average distortion associated with the best guess we can make when we know only the statistics of the source, but we do not have any knowledge of the specific outputs. The average mutual information is zero if and only if X and $\hat{X}$ are statistically independent, i.e., when $Q(\hat{x} \mid x) = Q(\hat{x})$ for all x and $\hat{x}$. Therefore, in this case, $d(X,\hat{X})$ can be written as,

$$d(X,\hat{X}) = \sum_{\hat{x}} Q(\hat{x}) \sum_{x} P(x) \rho(x,\hat{x}) . \qquad (4.10)$$

The minimum of $d(X,\hat{X})$ of (4.10) is found by setting $Q(\hat{x})=1$ for $\hat{x}$ which minimizes $\sum_{x} P(x) \rho(x,\hat{x})$, i.e.,

$$D_{\max} = \min_{\hat{x}} \sum_{x} P(x) \rho(x,\hat{x}) . \qquad (4.11)$$

$R(D)$ is only defined for $D \geq D_{\min}$, is positive for $D_{\min} \leq D < D_{\max}$, and vanishes for $D \geq D_{\max}$. It can also be shown that $R(D)$ is monotonically decreasing and downward convex in the interval $D_{\min} \leq D < D_{\max}$ [7].

4.1.2 The Parametric Expression for $R(D)$

As stated above, the variational problem defining $R(D)$ can be solved using the method of Lagrange multipliers which results in the following set of parametric expressions for D and R [7],

$$D = \sum_x \sum_{\hat{x}} \lambda(x) P(x) Q(\hat{x}) e^{s\rho(x,\hat{x})} \rho(x,\hat{x}), \tag{4.12}$$

and

$$R = sD + \sum_x P(x) \log \lambda(x), \tag{4.13}$$

where,

$$\lambda(x) = \left[\sum_{\hat{x}} Q(\hat{x}) e^{s\rho(x,\hat{x})} \right]^{-1} \tag{4.14}$$

and

$$Q(\hat{x}) = \sum_x P(x) Q(\hat{x} \mid x) \tag{4.15}$$

is the marginal probability of $\hat{x}$. The coefficients $\lambda(x)$ should satisfy the following relations:

$$c(\hat{x}) = \sum_x \lambda(x) P(x) e^{s\rho(x,\hat{x})} = 1 \qquad \text{if} \quad Q(\hat{x}) > 0, \tag{4.16a}$$

and

$$c(\hat{x}) = \sum_x \lambda(x) P(x) e^{s\rho(x,\hat{x})} \leq 1 \qquad \text{if} \quad Q(\hat{x}) = 0. \tag{4.16b}$$

Each value of $s \in (-\infty, 0]$ defines a point (D_s, R_s) on R(D). It can be shown that s is the slope of the rate-distortion function, i.e. , $R'(D_s) = s$. To find the rate-distortion function, we should first solve (4.14) and (4.16) for $Q(\hat{x})$ and

$\lambda(x)$ and then substitute these in (4.12) and (4.13). This is not always an easy task, with the analytic form of the rate-distortion function not known, except for some specific cases for which some sort of symmetry holds. A very useful computational algorithm is available, however, which can be used to numerically obtain the R(D) curve. This algorithm is due to Blahut [46] and is applicable to a variety of memoryless sources and can also be used to numerically evaluate an upper bound for the rate-distortion function for sources with memory by computing the rate-distortion function of n-tuples of the source for moderate values of n.

4.1.3 The Rate-Distortion Function for the Source With Memory

For the case of a source with memory, the rate-distortion function is defined as,

$$R(D) := \lim_{n \to \infty} R_n(D) \tag{4.17}$$

where,

$$R_n(D) = n^{-1} \min_{Q(\hat{\underline{x}} \mid \underline{x}) \in Q_D} I(\underline{\mathbf{X}} ; \hat{\underline{\mathbf{X}}}) . \tag{4.18}$$

The n-tuple $\underline{\mathbf{x}} = (x_1, \ldots, x_n)$ consists of n consecutive source letters. Similarly, $\hat{\underline{\mathbf{x}}} = (\hat{x}_1, \ldots, \hat{x}_n)$ consists of n consecutive reproducing letters. The quantities Q_D and $I(\underline{\mathbf{X}} ; \hat{\underline{\mathbf{X}}})$ are defined in the same fashion as (4.2) and (4.4), except for the fact that the probabilities involved here are defined over n-vectors of source and reproducing alphabets.

4.1.4 The Source Coding Theorems for the Classical Model

The importance of the rate-distortion function lies in the fact that $R(D)$ is the minimum possible rate at which the source can be encoded with an average distortion not exceeding D. This can be elegantly and concisely stated in terms of the source coding theorem and its converse [7]:

Theorem 4.1- Source Coding Theorem:

Given any $\epsilon > 0$ and N large enough, a block code can be found with blocklength N and rate $R \leq R(D) + \epsilon$ such that the average distortion $d \leq D + \epsilon$.

Theorem 4.2- Converse Source Coding Theorem:

For any source encoder-decoder pair, it is impossible to achieve average distortion less than or equal to D whenever $R < R(D)$.

4.2 A Generalized Communications System Model

Up to this point we have assumed that the designer has access to the output of the information source and, therefore, can encode it directly. It was also assumed that the output of the decoder is delivered to the user without further distortion. Dobrushin and Tsybakov [13] have given different examples of several cases for which the above assumptions are not true, and have proposed a generalization of the classical communication system model of Figure 4.1. This model has been further studied by Wolf and Ziv [47] and its discrete case has been formulated by Berger [7]. In this model, two extra mappings representing the noise at the source and receiver have been added to the classical model of Figure 4.1. The more complete, or general, model is shown in Figure 4.3. In this model, the conditional probabilities $Q_1(z \mid x)$ and $Q_2(\hat{x} \mid \hat{z})$ represent the noise at the source and receiver, respectively. In the special case where there is no additional noise at the source and receiver, the generalized model clearly reduces to the classical one.

Assuming distortionless transmission through the channel encoder, channel, and channel decoder, as in the previous case, results in the source coding model of Figure 4.4. In this case, the rate-distortion function can be found by minimizing the average mutual information between Z and $\hat{Z}$, i.e., $I(Z;\hat{Z})$, subject to the condition that the average distortion between X and $\hat{X}$ be less than or equal to a certain value D. We notice here, that while the mutual information between Z and $\hat{Z}$ is considered, the minimization is performed with respect to a condition imposed on X and $\hat{X}$. In an attempt to make the equations defining the rate-distortion function in this case similar to the corresponding equations derived for the previously considered case, we might try to replace the condition on X and $\hat{X}$ with an equivalent condition on Z and $\hat{Z}$.

For each conditional probability distribution $Q(\hat{z} \mid z)$, the corresponding conditional probability distribution $P(\hat{x} \mid x)$ over $\hat{x}$ and x is,

$$P(\hat{x} \mid x) = \sum_z \sum_{\hat{z}} Q_1(z \mid x) Q_2(\hat{x} \mid \hat{z}) Q(\hat{z} \mid z) . \qquad (4.19)$$

Now, if we define the modified distortion measure $\hat{\rho}(z,\hat{z})$ as,

$$\hat{\rho}(z,\hat{z}) = \frac{1}{Q_1(z)} \sum_x \sum_{\hat{x}} P(x) Q_1(z \mid x) Q_2(\hat{x} \mid \hat{z}) \rho(x,\hat{x}) , \qquad (4.20)$$

where,

$$Q_1(z) = \sum_x P(x) Q_1(z \mid x) , \qquad (4.21)$$

then, it is easy to show that,

$$\underset{z,\hat{z}}{E}\{\hat{\rho}(Z,\hat{Z})\} = \sum_{z}\sum_{\hat{z}} Q_1(z)Q(\hat{z} \mid z)\hat{\rho}(z,\hat{z})$$

$$= \sum_{x}\sum_{\hat{x}} P(x)P(\hat{x} \mid x)\rho(x,\hat{x})$$

$$= \underset{x,\hat{x}}{E}\{\rho(X,\hat{X})\}$$

In other words, $\hat{\rho}(z,\hat{z})$ as defined by (4.20), when applied to Z and $\hat{Z}$, has the same effect as applying $\rho(x,\hat{x})$ to X and $\hat{X}$. Thus, reproducing X with an average distortion D with respect to $\rho(x,\hat{x})$ is equivalent to reproducing Z with an average distortion D with respect to $\hat{\rho}(z,\hat{z})$.

Now, we can define the rate-distortion function as follows,

$$R(D) := \min_{Q(\hat{z} \mid z) \in Q_D} I(Z;\hat{Z}), \tag{4.22}$$

where

$$Q_D = \left\{ Q(\hat{z} \mid z) \colon E\{\hat{\rho}(Z,\hat{Z})\} \leq D \right\}. \tag{4.23}$$

For the generalized communication system model of Figure 4.3, the source coding theorem and its converse can be stated as [13] :

Theorem 4.3- Source Coding Theorem for the Case of Additional Noise:

For all $\epsilon>0$ and $D \geq 0$, it is possible to design the encoder and decoder of Figure 4.4 so that the system reproduces the source output with fidelity $D+\epsilon$ and rate $R \leq R(D)+\epsilon$.

Theorem 4.4- Converse Source Coding Theorem for the Case of Additional Noise:

It is impossible to reproduce the source, using the system of Figure 4.4, with an average distortion less than or equal to D whenever $R < R(D)$.

4.3 Modeling of the Pattern Recognition Process

In this section, we use the generalized communications system model discussed in the previous section to model the pattern recognition process. We consider the feature vectors representing the patterns as the outputs of a noisy channel whose inputs are the objects belonging to two or more pattern classes. In this sense, the source can be viewed as nature, while the interference channel represents our measurement devices. The encoder can be viewed as a feature extractor which provides the decoder (the decision-making device) with the required information for classifying the patterns. The goal of the encoder-decoder pair is to minimize the probability of misclassification subject to some constraint on the computational complexity, e.g., the number of templates examined in template matching, or the average length of the tree when a decision tree is used.

The above mentioned model is shown in Figure 4.5. For the case of M class pattern recognition, the source outputs are $x_1, x_2, \ldots, x_n$, where $x_i \in \{0, \ldots, M-1\}$. The encoder only has access to the feature vectors $\underline{\mathbf{z}}_1, \ldots, \underline{\mathbf{z}}_n$, where $\underline{\mathbf{z}}_i = (z_{i1}, \ldots, z_{ik})$. The feature vectors $\underline{\mathbf{z}}_i$ can be either real-valued or discrete-valued (e.g., binary). We assume the latter, however, generalization to real-valued vectors is straightforward. The relation between the feature vectors and the classes to which they belong is given by the class conditional probability distribution $Q(\underline{\mathbf{z}} \mid x)$. For each sequence of feature vectors, $\underline{\mathbf{z}}_1, \ldots, \underline{\mathbf{z}}_n$, the encoder generates a binary sequence, to which the decoder assigns the sequence of classes $\hat{x}_1, \ldots, \hat{x}_n$. The performance of the pattern recognition system, then, can be evaluated by the relationship between the average number of information bits provided to the decoder, and the average distortion between the source sequence (actual classes), $x_1, \ldots, x_n$, and the decisions made by the decoder, i.e., $\hat{x}_1, \ldots, \hat{x}_n$. In terms of rate-distortion theoretic concepts, this relationship can be expressed as,

$$R(D) = \min_{\hat{P}_D} I(\underline{\mathbf{Z}};\hat{X}) , \tag{4.24}$$

where the quantity $I(\underline{\mathbf{Z}};\hat{X})$ is the average mutual information between $\underline{\mathbf{Z}}$ and $\hat{X}$,

$$I(\underline{\mathbf{Z}};\hat{X}) = \sum_{\underline{z}} \sum_{\hat{x}} Q(\underline{z}) \hat{P}(\hat{x} \mid \underline{z}) \log \frac{\hat{P}(\hat{x} \mid \underline{z})}{\hat{P}(\hat{x})} , \tag{4.25}$$

and

$$Q(\underline{z}) = \sum_{x} P(x) Q(\underline{z} \mid x) . \tag{4.26}$$

The minimum in (4.24) is taken over all encoder-decoder pairs, i.e., over all conditional probability distributions $\hat{P}(\hat{x} \mid \underline{z})$ which result in an average distortion (probability of error) between X and $\hat{X}$ less than or equal to a given value D, i.e.,

$$\hat{P}_D = \left\{ \hat{P}(\hat{x} \mid \underline{z}) : d(X,\hat{X}) \leq D \right\} , \tag{4.27}$$

where,

$$d(X,\hat{X}) = \sum_{x} \sum_{\hat{x}} P(x) \hat{P}(\hat{x} \mid x) \rho(x,\hat{x}) . \tag{4.28}$$

The quantity $\rho(x,\hat{x})$ is the penalty of deciding in favor of $\hat{x}$ when the pattern actually belongs to class x and,

$$\hat{P}(\hat{x} \mid x) = \sum_{\mathbf{\underline{z}}} Q(\mathbf{\underline{z}} \mid x)\hat{P}(\hat{x} \mid \mathbf{\underline{z}}) . \tag{4.29}$$

To translate the fidelity condition of (4.27) into a condition on $\mathbf{\underline{Z}}$ and $\hat{X}$, we define the equivalent distortion measure $\hat{\rho}(\mathbf{\underline{z}},\hat{x})$ as follows :

$$\hat{\rho}(\mathbf{\underline{z}},\hat{x}) = \frac{1}{Q(\mathbf{\underline{z}})} \sum_{x} P(x) Q(\mathbf{\underline{z}} \mid x) \rho(x,\hat{x}) . \tag{4.30}$$

Then the equivalent fidelity condition can be written as,

$$\hat{P}_D = \left\{ \hat{P}(\hat{x} \mid \mathbf{\underline{z}}) : \; d(\mathbf{\underline{Z}},\hat{X}) \le D \right\} , \tag{4.31}$$

where,

$$d(\mathbf{\underline{Z}},\hat{X}) = \sum_{\mathbf{\underline{z}}} \sum_{\hat{x}} Q(\mathbf{\underline{z}})\hat{P}(\hat{x} \mid \mathbf{\underline{z}})\hat{\rho}(\mathbf{\underline{z}},\hat{x}) . \tag{4.32}$$

4.3.1 Probability of Error Criterion

Since we are interested in minimizing the average probability of error, we use the following distortion measure,

$$\rho(x,\hat{x}) = \begin{cases} 1 , & \hat{x} \neq x , \\ 0 , & \hat{x} = x . \end{cases} \tag{4.33}$$

With this choice of $\rho(x,\hat{x})$, we have,

$$\begin{aligned} d(X,\hat{X}) &= \sum_{x} \sum_{\hat{x}} P(x) P(\hat{x} \mid x) \rho(x,\hat{x}) \\ &= \sum_{x} P(x) \sum_{\hat{x} \neq x} P(\hat{x} \mid x) \end{aligned}$$

$$= \sum_{x} P(x) Pr(\epsilon \mid x) = Pr(\epsilon),$$

where, $Pr(\epsilon)$ is the average probability of error and $Pr(\epsilon \mid x)$ is the probability of error for a given x.

For $\rho(x,\hat{x})$ described by (4.33), $\hat{\rho}(\underline{z},\hat{x})$ is,

$$\hat{\rho}(\underline{z},\hat{x}) = \frac{1}{Q(\underline{z})} \sum_{x \neq \hat{x}} P(x) Q(\underline{z} \mid x). \tag{4.34}$$

Since,

$$\sum_{x \neq \hat{x}} P(x) Q(\underline{z} \mid x) = \sum_{x} P(x) Q(\underline{z} \mid x) - P(X=\hat{x}) Q(\underline{z} \mid X=\hat{x})$$

$$= Q(\underline{z}) - P(X=\hat{x}) Q(\underline{z} \mid X=\hat{x}).$$

Therefore, $\hat{\rho}(\underline{z},\hat{x})$ can alternatively be expressed as,

$$\hat{\rho}(\underline{z},\hat{x}) = 1 - P(X=\hat{x} \mid \underline{z}), \tag{4.35}$$

where $P(X=\hat{x} \mid \underline{z})$ is the a posteriori probability of $\hat{x}$ given $\underline{z}$.

4.3.2 The parametric Form of the Rate-Distortion Function

The minimum value of the average distortion $D_{\min}$ is,

$$D_{\min} = \sum_{\underline{z}} Q(\underline{z}) \hat{\rho}(\underline{z}), \tag{4.36}$$

where,

$$\hat{\rho}(\underline{z}) = \min_{\hat{x}} \hat{\rho}(\underline{z},\hat{x}) = \min_{\hat{x}} \frac{1}{Q(\underline{z})} \sum_{x \neq \hat{x}} P(x) Q(\underline{z} \mid x)$$

$$= 1 - \max_{\hat{x}} P(X = \hat{x} \mid \underline{\mathbf{z}}) ; \qquad (4.37)$$

and, therefore,

$$D_{\min} = \sum_{\underline{\mathbf{z}}} \min_{\hat{x}} \sum_{x \neq \hat{x}} P(x) Q(\underline{\mathbf{z}} \mid x)$$

$$= 1 - \sum_{\underline{\mathbf{z}}} Q(\underline{\mathbf{z}}) \max_{\hat{x}} P(X = \hat{x} \mid \underline{\mathbf{z}}) . \qquad (4.38)$$

In other words, $D_{\min}$ is achieved when for each $\underline{\mathbf{z}}$, the pattern recognizer decides in favor of the class with highest a posteriori probability, i.e., if it uses the Bayesian decision rule.

The minimum expected distortion when the system does not have any knowledge of the specific feature vectors is,

$$D_{\max} = \min_{\hat{x}} \sum_{\underline{\mathbf{z}}} Q(\underline{\mathbf{z}}) \rho(\underline{\mathbf{z}}, \hat{x})$$

$$= \min_{\hat{x}} \sum_{\underline{\mathbf{z}}} Q(\underline{\mathbf{z}}) [1 - P(X = \hat{x} \mid \underline{\mathbf{z}})]$$

$$= \min_{\hat{x}} [1 - P(X = \hat{x})]$$

$$= 1 - \max_{\hat{x}} P(X = \hat{x}) . \qquad (4.39)$$

This result is expected, since, in this case, the only reasonable strategy is to always decide in favor of the class with the highest a priori probability.

The rate-distortion function $R(D)$ is positive and monotonically non-increasing for $D_{\min} \leq D < D_{\max}$ and is zero for $D \geq D_{\max}$. The rate-distortion function is not defined for $D < D_{\min}$. We note that, in this case, $D_{\min} \neq 0$ in general. To reduce $R(D)$ to a standard rate-distortion function, i.e., one with

$D_{\min}=0$, we define the following distortion measure,

$$\tilde{\rho}(\underline{z},\hat{x}) = \hat{\rho}(\underline{z},\hat{x}) - \hat{\rho}(\underline{z}) \tag{4.40}$$

$$= \frac{1}{Q(\underline{z})}\sum_{x\neq\hat{x}} P(x)Q(\underline{z}\mid x) - \min_{\hat{x}}\frac{1}{Q(\underline{z})}\sum_{x\neq\hat{x}} P(x)Q(\underline{z}\mid x)$$

$$= \max_{\hat{x}} P(X=\hat{x}\mid\underline{z}) - P(X=\hat{x}\mid\underline{z}) \ .$$

Then $R(D)=\tilde{R}(D - D_{\min})$ for all $D \geq D_{\min}$, where $\tilde{R}(D)$ is the rate-distortion function of the source $\underline{Z}$ with distribution $Q(\underline{z})$ with respect to fidelity criterion $\tilde{\rho}(\underline{z},\hat{x})$ and $R(D)$ is the original rate-distortion function.

The parametric expression for $\tilde{R}(D)$, similar to (4.11) and (4.12), is,

$$D = \sum_{\underline{z}}\sum_{\hat{x}}\lambda(\underline{z})Q(\underline{z})\hat{P}(\hat{x})e^{s\tilde{\rho}(\underline{z},\hat{x})}\tilde{\rho}(\underline{z},\hat{x}) \ , \tag{4.41}$$

and,

$$\tilde{R} = sD + \sum_{\underline{z}}Q(\underline{z})\log\lambda(\underline{z}) \ , \tag{4.42}$$

where $\lambda(\underline{z})$ is defined as,

$$\lambda(\underline{z}) = \left[\sum_{\hat{x}}\hat{P}(\hat{x})e^{s\tilde{\rho}(\underline{z},\hat{x})}\right]^{-1} \ , \tag{4.43}$$

and satisfies the following inequality,

$$\sum_{\underline{z}}\lambda(\underline{z})Q(\underline{z})e^{s\tilde{\rho}(\underline{z},\hat{x})} \leq 1 \ . \tag{4.44}$$

The inequality in (4.44) should hold with equality for those $\hat{x}$ for which $\hat{P}(\hat{x}) > 0$.

4.3.3 Numerical Computation of R(D)

As stated in the previous section, solving the above equations to derive the explicit expression for $\tilde{R}(D)$ is difficult for the general case. However, $\tilde{R}(D)$ can be numerically evaluated using the following recursion, called Blahut's algorithm [46].

For any $s<0$, we start from any probability M-vector $\underline{\hat{\mathbf{P}}}_0 = [\hat{P}_0(0), \ldots, \hat{P}_0(M-1)]$, having strictly positive components and recursively define $\underline{\hat{\mathbf{P}}}_n = [\hat{P}_n(0), \ldots, \hat{P}_n(M-1)]$ as,

$$\hat{P}_{n+1}(\hat{x}) = c_n(\hat{x})\hat{P}_n(\hat{x}) , \tag{4.45}$$

where,

$$c_n(\hat{x}) = \sum_{\underline{\mathbf{z}}} \lambda_n(\underline{\mathbf{z}}) Q(\underline{\mathbf{z}}) e^{s\tilde{\rho}(\underline{\mathbf{z}},\hat{x})} , \tag{4.46a}$$

and,

$$\lambda_n(\underline{\mathbf{z}}) = \left[\sum_{\hat{x}} \hat{P}_n(\hat{x}) e^{s\tilde{\rho}(\underline{\mathbf{z}},\hat{x})} \right]^{-1} . \tag{4.46b}$$

It can be shown that $\underline{\hat{\mathbf{P}}}_n$ converges to the probability M-vector $\underline{\hat{\mathbf{P}}}(s)$ that generates the point on $\tilde{R}(D)$ at which $\tilde{R}'(D) = s$ [46]. Furthermore, it can be shown that [7],

$$\tilde{R}_{n+1} \geq \tilde{R}(D_{n+1}) \geq \tilde{R}_{n+1} - \log[\max_{\hat{x}} c_n(\hat{x})] , \tag{4.47}$$

where $(D_{n+1}, \tilde{R}_{n+1})$ is the point found by substituting the $\hat{P}_{n+1}(\hat{x})$'s and $\lambda_{n+1}(\underline{\mathbf{z}})$'s into (4.41) and (4.42). Therefore, in order to generate a point lying less than ϵ above the $\tilde{R}(D)$ curve, we only need to iterate (4.45) and (4.46) until $\log[\max_{\hat{x}} c_n(\hat{x})] < \epsilon$.

4.3.4 An Illustrative Example

We conclude this chapter by finding the rate-distortion function for a typical example. Consider a four-class pattern recognition system, i.e., $x \in \{0, 1, 2, 3\}$, having ten binary features, $\underline{z}=(z_1, \ldots, z_{10})$, where $z_i \in \{0, 1\}$. Let the a priori probabilities be $P(0) = P(3) = \frac{3}{8}$ and $P(1) = P(2) = \frac{1}{8}$. The interference channel Q is a symmetric channel defined as follows: $Q(z_i \mid 0)$, $i=1, \ldots, 10$ are uniformly placed in the interval [0.1, 0.19]; $Q(z_i \mid 1)$'s are uniformly placed in the interval [0.2, 0.29]; $Q(z_i \mid 3)=1-Q(z_i \mid 0)$; and $Q(z_i \mid 2)=1-Q(z_i \mid 1)$, where,

$$Q(z_i \mid x) = Prob.(z_i=1 \mid X=x) \ .$$

The features are assumed independent and, therefore,

$$Q(\underline{z} \mid x) = \prod_{i=1}^{10} Q^{z_i}(z_i \mid x)[1-Q(z_i \mid x)]^{1-z_i} \ .$$

The rate-distortion curve for this example is given in Figure 4.6. This example not only demonstrates the application of the method discussed above, but also may prove useful in finding the rate-distortion function of the much more difficult and realistic case of a binary source with memory discussed in Chapter 6. This is due to the fact that for a binary symmetric Markov source with transition probability $\frac{1}{4}$, in order to find the rate-distortion function for $n=2$, i.e., $R_2(D)$, we have to consider four compound classes (1,1),(1,2),(2,1), and (2,2) with probabilities $\frac{3}{8}$, $\frac{1}{8}$, $\frac{1}{8}$, and $\frac{3}{8}$, respectively.

4.4 Discussion

In this chapter, the classical communication model of Shannon and its generalization due to Dobrushin and Tsybakov were introduced. The advantage of the generalized model lies in the fact that it can take into consideration the extra interferences at the source and receiver which are not under systems designer's control. This generalized model is suitable for modeling of the pattern recognition process, where patterns can be considered as the outputs of a noisy source. Therefore, in this chapter, we have used this generalized communication system in order to model the pattern recognition process and presented the interpretation of the elements of this model. In Chapter 5, we will apply this model to two-class pattern recognition problem with statistically independent patterns. The application of the model to correlated patterns together with certain interesting conclusions appear in Chapter 6.

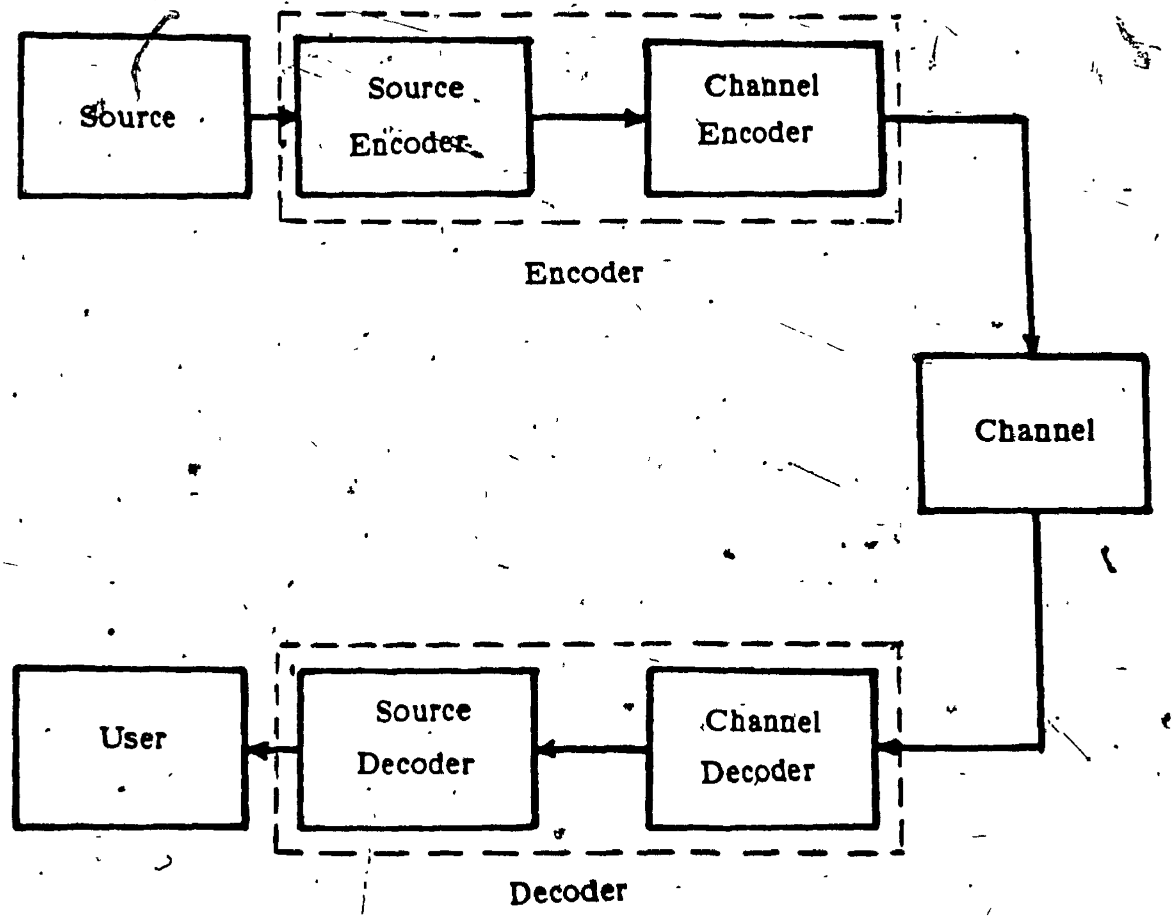

Figure 4.1 : Shannon's Classical Model for a Communication System.

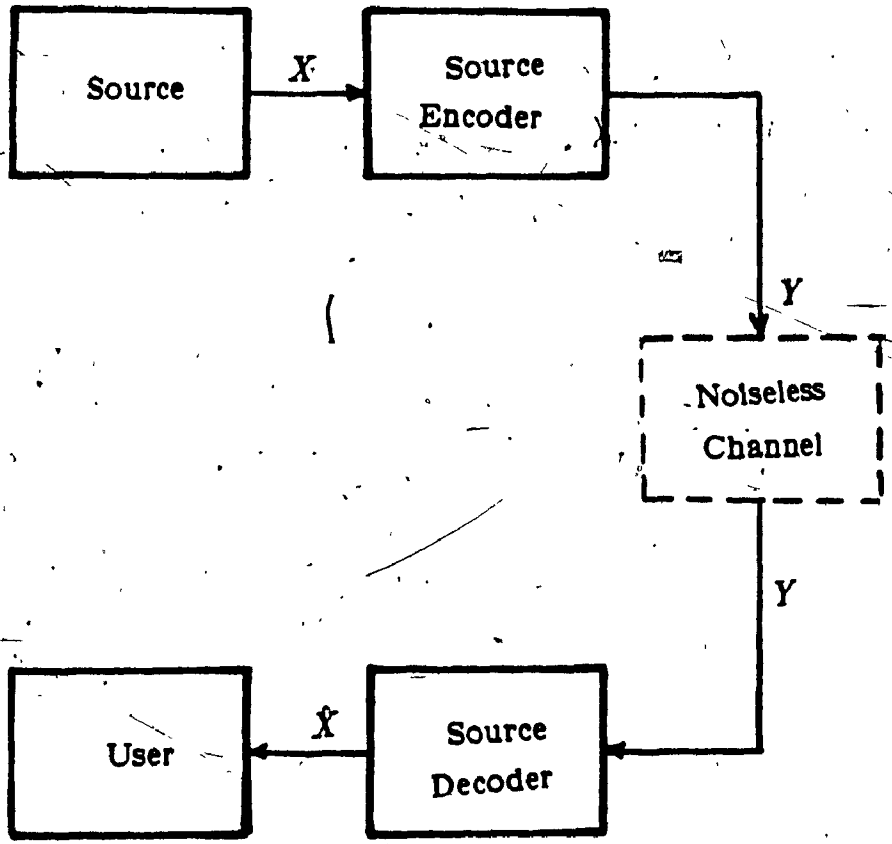

Figure 4.2 : The Classical Source Coding Model.

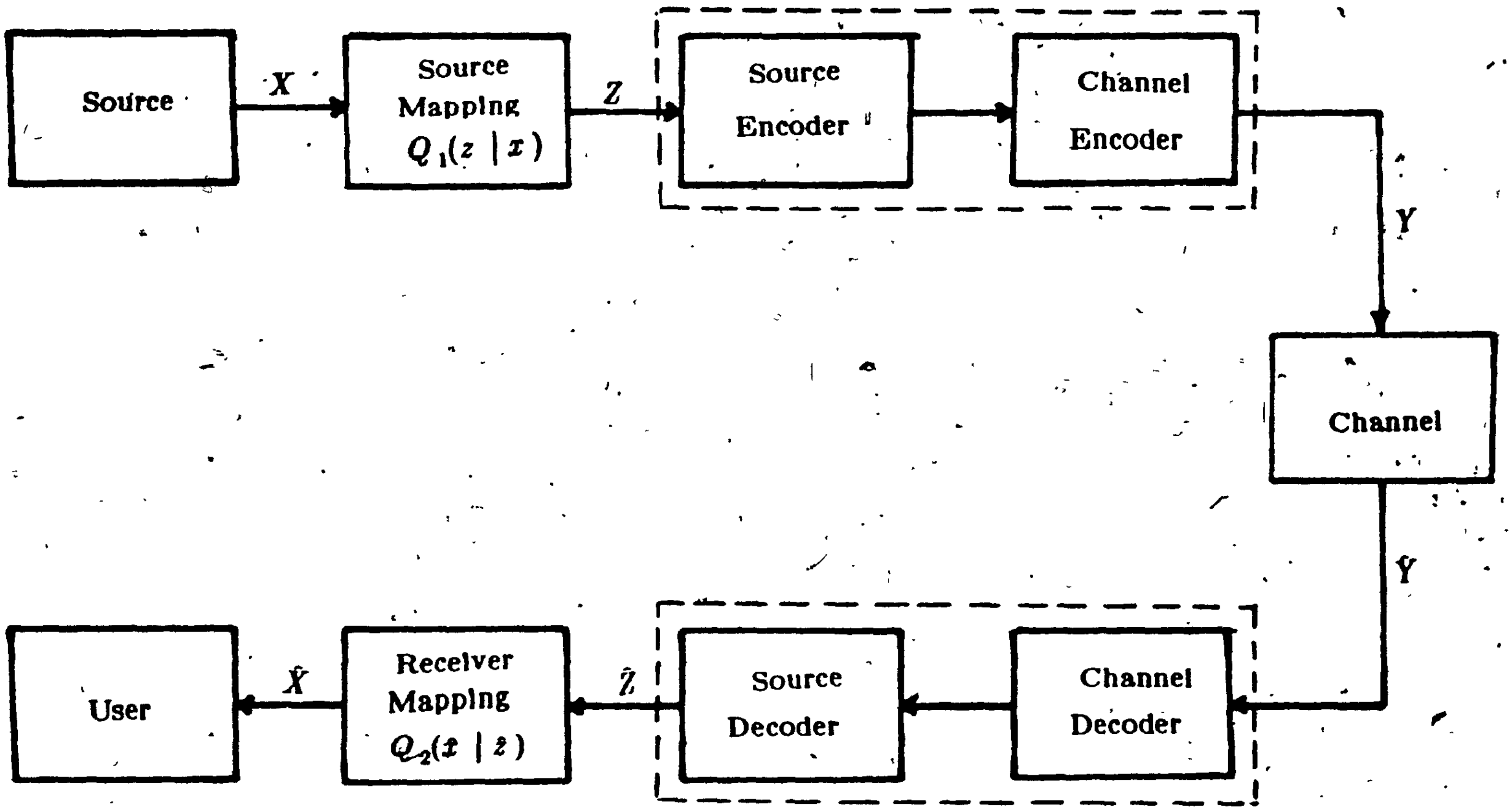

Figure 4.3 : The Generalized Model for a Communication System.

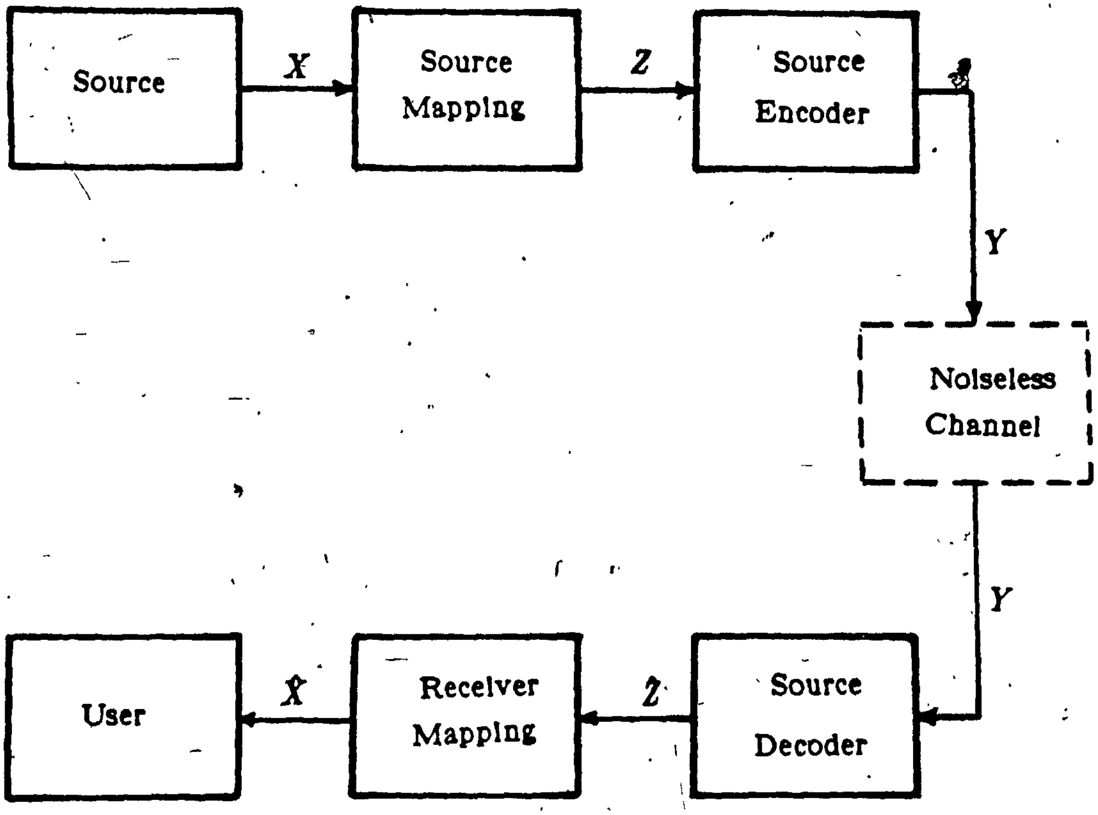

Figure 4.4 : The Source Coding Model for the Generalized Case.

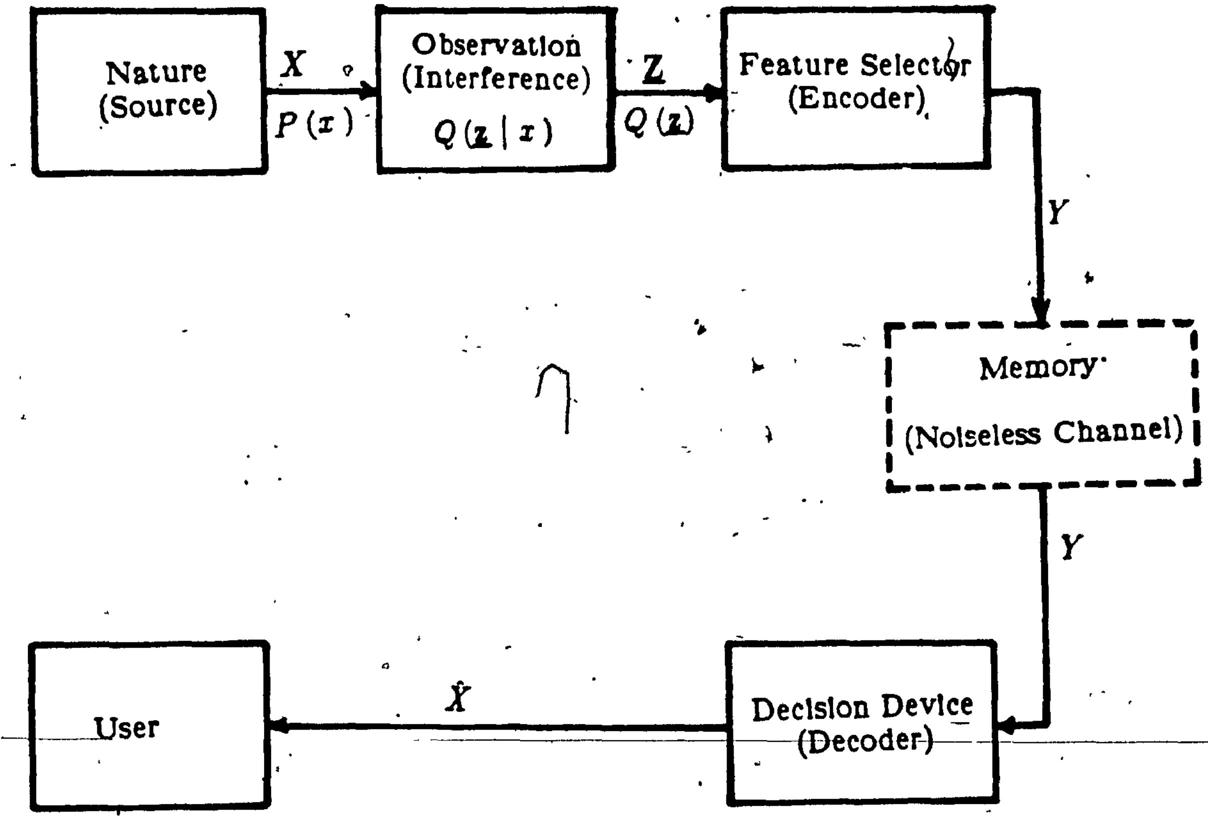

Figure 4.5 : The Model for the Pattern Recognition Process.

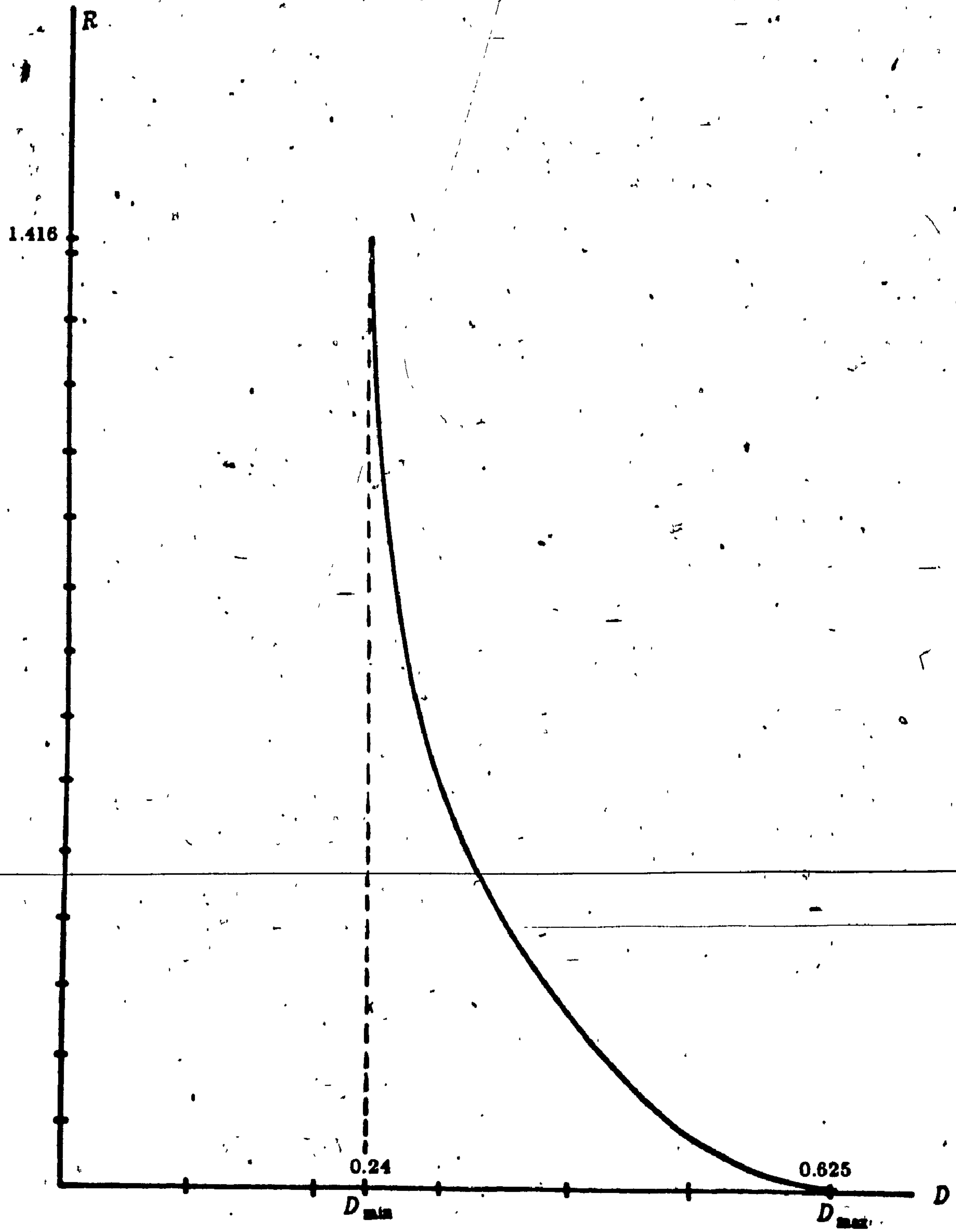

Figure 4.6 : The R(D) Curve for 4-Class, 10-Feature Example

CHAPTER 5

Application of the Model : Uncorrelated Patterns

In this chapter, we consider two-class pattern recognition, with statistically independent patterns, in the context of rate-distortion theory. First, we consider the general case, i.e., we do not make any assumptions restricting either the values of the a priori probabilities of the classes or the form of the interference channel modeling the pattern generation mechanism. For this general case, we specialize the method discussed in the previous chapter and derive a recursive formula for the numerical evaluation of the rate-distortion function. Second, we consider the case with equiprobable classes and a symmetry condition imposed on the interference channel. For this case, it is possible to derive the explicit form of the rate-distortion function.

5.1 General Two-Class Pattern Recognition

The general two-class pattern recognition problem can be formulated as follows. Assume that the patterns belong to classes 0 and 1 with probabilities p and 1-p, respectively. In terms of the model considered in the previous chapter, this means that X takes values $x \in \{0, 1\}$ with probabilities $P(X=0)=p$ and $P(X=1)=1-p$. Without any loss of generality, we assume that $p \le \frac{1}{2}$. The patterns are represented by the feature vectors $\mathbf{z}$ with class conditional distributions $Q(\mathbf{z} \mid 0)$ and $Q(\mathbf{z} \mid 1)$; therefore the marginal probability of the feature vector $\mathbf{z}$

is given by,

$$Q(\underline{z}) = pQ(\underline{z} \mid 0) + (1-p)Q(\underline{z} \mid 1) \quad . \tag{5.1}$$

The encoder takes one or several feature vectors as its input and generates a set of definitions or decision rules to be used by the decoder in order to estimate the class(es) to which the pattern(s) represented by the feature vector(s) actually belong. Since any set of definitions or rules can be expressed as a binary sequence, the output of the encoder can be viewed as a number of dichotomies asked by the decoder and answered by the encoder, before the former makes its decision. The goal of the encoder-decoder pair is to minimize the amount of information provided to the decoder by the encoder, i.e., to minimize the number of questions that require answers, while maintaining a certain level of reliability of the decisions made by the decoder. In fact, the encoder in our model represents a feature extractor and the rate-distortion function found gives an indication of the minimum number of features that must be examined in order to achieve a certain accuracy level of the decisions.

We denote the output of the decoder by $\hat{X}$ as before, where $\hat{X}$ takes values $x \in \{0, 1\}$. The operation of the encoder-decoder pair can be modeled using the transition probabilities $\hat{P}(\hat{x} \mid \underline{z})$. The problem then is to find the $\hat{P}(\hat{x} \mid \underline{z})$'s in such a way that the average information conveyed to the decoder about the patterns, i.e., $I(\underline{Z};\hat{X})$ is minimized. The quantity $I(\underline{Z};\hat{X})$ is the average mutual information between $\underline{Z}$ and $\hat{X}$ and is defined as,

$$I(\underline{Z};\hat{X}) = \sum_{\underline{z}} \sum_{\hat{x}=0}^{1} Q(\underline{z})\hat{P}(\hat{x} \mid \underline{z})\log\frac{\hat{P}(\hat{x} \mid \underline{z})}{\hat{P}(\hat{x})} \tag{5.2}$$

$$= \sum_{\underline{z}} Q(\underline{z})[\hat{P}(0 \mid \underline{z})\log\frac{\hat{P}(0 \mid \underline{z})}{\hat{P}(0)} + \hat{P}(1 \mid \underline{z})\log\frac{\hat{P}(1 \mid \underline{z})}{\hat{P}(1)}] ,$$

where $\hat{P}(\hat{x}) = \sum_{\mathbf{z}} Q(\mathbf{z})\hat{P}(\hat{x} \mid \mathbf{z})$ is the probability that the decoder decides in favor of the class $\hat{x}$.

As discussed earlier, this minimization is not an unconstrained optimization problem. In other words, the conditional probabilities which minimize $I(\mathbf{Z};\hat{X})$ should also guarantee a certain amount of reliability of the decoder decisions. The constraint to be satisfied is expressed as,

$$d(X,\hat{X}) = \sum_{x=0}^{1} \sum_{\hat{x}=0}^{1} P(x)\hat{P}(\hat{x} \mid x)\rho(x,\hat{x}) \leq \hat{D} \ , \tag{5.3}$$

where $\rho(x,\hat{x})$ is the cost of deciding in favor of $\hat{x}$ when the pattern actually belongs to the class x. As before, we take the probability of error fidelity criterion, i.e., $\rho(x,\hat{x}) = 1-\delta_{x\hat{x}}$, where $\delta_{x\hat{x}}$ is the Kronecker delta. This fidelity criterion can be expressed in the matrix form,

$$\rho = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} . \tag{5.4}$$

For this distortion measure, $d(X,\hat{X})$ will be equal to the average probability of error, i.e.,

$$d(X,\hat{X}) = P(0)\hat{P}(1 \mid 0) + P(1)\hat{P}(0 \mid 1) . \tag{5.5}$$

The fidelity condition expressed as (5.3) does not expose the relation to the transition probabilities $\hat{P}(\hat{x} \mid \mathbf{z})$. However, we note that,

$$P(\hat{x} \mid x) = \sum_{\mathbf{z}} Q(\mathbf{z} \mid x)\hat{P}(\hat{x} \mid \mathbf{z}) . \tag{5.6}$$

In deriving (5.6), we have utilized the fact that the output of the decoder for a given $\mathbf{z}$ does not depend on x, i.e., $\hat{P}(\hat{x} \mid \mathbf{z},x) = \hat{P}(\hat{x} \mid \mathbf{z})$. In other words, the variables X, $\underline{\mathbf{Z}}$, and $\hat{X}$ form a Markov chain. Now, substituting (5.6) in (5.3), we have,

$$d(X,\hat{X}) = \sum_{x=0}^{1} \sum_{\hat{x}=0}^{1} P(x)[\sum_{\underline{\mathbf{z}}} Q(\underline{\mathbf{z}} \mid x)\hat{P}(\hat{x} \mid \underline{\mathbf{z}})]\rho(x,\hat{x}) \tag{5.7}$$

$$= \sum_{\underline{\mathbf{z}}} \sum_{\hat{x}=0}^{1} \hat{P}(\hat{x} \mid \underline{\mathbf{z}})[\sum_{x=0}^{1} P(x)Q(\underline{\mathbf{z}} \mid x)\rho(x,\hat{x})] \leq D .$$

As before, we can define the equivalent distortion measure $\tilde{\rho}(\underline{\mathbf{z}},\hat{x})$ as,

$$\tilde{\rho}(\underline{\mathbf{z}},\hat{x}) = \frac{1}{Q(\underline{\mathbf{z}})} \sum_{x=0}^{1} P(x)Q(\underline{\mathbf{z}} \mid x)\rho(x,\hat{x}) , \tag{5.8}$$

and, therefore, write the constraint of (5.7) as,

$$\tilde{d}(\underline{\mathbf{Z}},\hat{X}) = \sum_{\underline{\mathbf{z}}} \sum_{\hat{x}=0}^{1} Q(\underline{\mathbf{z}})\hat{P}(\hat{x} \mid \underline{\mathbf{z}})\tilde{\rho}(\underline{\mathbf{z}},\hat{x}) \leq D . \tag{5.9}$$

For the probability of error distortion measure (5.4), the quantity $\tilde{\rho}(\underline{\mathbf{z}},\hat{x})$ can be written as,

$$\tilde{\rho}(\underline{\mathbf{z}},\hat{x}) = \begin{cases} \dfrac{(1-p)Q(\underline{\mathbf{z}} \mid 1)}{Q(\underline{\mathbf{z}})} , & \hat{x}=0 , \\[2ex] \dfrac{pQ(\underline{\mathbf{z}} \mid 0)}{Q(\underline{\mathbf{z}})} , & \hat{x}=1 . \end{cases} \tag{5.10}$$

The minimum value that the average distortion $d(\underline{\mathbf{Z}},\hat{X})$ can assume, i.e., $D_{\min}$ is found using (4.38),

$$D_{\min} = \sum_{\underline{\mathbf{z}}} \min[(1-p)Q(\underline{\mathbf{z}} \mid 1) , pQ(\underline{\mathbf{z}} \mid 0)] . \tag{5.11}$$

This value of $D_{\min}$ is achieved by deciding in favor of the class with the higher a posteriori probability for each pattern $\underline{\mathbf{z}}$.

The quantity $D_{\max}$, i.e., the minimum average probability of error achievable when no information is conveyed to the decoder concerning the specific patterns, is found from (4.39) as,

$$\begin{aligned} D_{\max} &= \min_{\hat{x}} \sum_{\underline{\mathbf{z}}} Q(\underline{\mathbf{z}}) \hat{\rho}(\underline{\mathbf{z}},\hat{x}) \\ &= \min[\sum_{\underline{\mathbf{z}}} (1-p) Q(\underline{\mathbf{z}} \mid 1) \, , \, \sum_{\underline{\mathbf{z}}} p Q(\underline{\mathbf{z}} \mid 0)] \\ &= \min[1-p \ , p\,] = p \ . \end{aligned} \tag{5.12}$$

Note that $D_{\max}$ is achieved by always deciding in favor of the class with higher a priori probability, in this case, class 1, since it is assumed that $p \leq \frac{1}{2}$.

The quantities $D_{\min}$ and $D_{\max}$ determine the end points of the rate-distortion function $R(D)$. The rate-distortion function is only defined for $D \geq D_{\min}$ and is positive and monotonically non-increasing in the interval $D_{\min} < D < D_{\max}$. It is zero for $D \geq D_{\max}$.

Since $\hat{\rho}(\underline{\mathbf{z}},\hat{x}) > 0$ for all $\underline{\mathbf{z}}$ and all $\hat{x}$, $D_{\min} \neq 0$. However, we can transform the rate-distortion function $R(D)$ into a rate-distortion function with $D_{\min}=0$ by introducing the translated distortion measure $\tilde{\rho}(\underline{\mathbf{z}},\hat{x})$ as,

$$\tilde{\rho}(\underline{\mathbf{z}},\hat{x}) = \hat{\rho}(\underline{\mathbf{z}},\hat{x}) - \hat{\rho}(\underline{\mathbf{z}}) \tag{5.13}$$

where,

$$\hat{\rho}(\underline{\mathbf{z}}) = \min_{\hat{x}} \hat{\rho}(\underline{\mathbf{z}},\hat{x})$$

5.1.1 Parametric Expression for the Rate-Distortion Function

Now, we can first find the rate-distortion function $\tilde{R}(D)$ of the source $\underline{\mathbf{z}}$ with probability distribution function $Q(\underline{\mathbf{z}})$ with respect to the distortion measure $\tilde{\rho}(\underline{\mathbf{z}},\hat{x})$ and then find the rate-distortion function $R(D)$, i.e., the rate-distortion function of the source $\underline{\mathbf{z}}$ with respect to the fidelity criterion $\bar{\rho}(\underline{\mathbf{z}},\hat{x})$ using the relation $\bar{R}(D) = \tilde{R}(D - \bar{D}_{\min})$ for all $D \geq D_{\min}$. For the distortion measure of (5.10) we have,

$$\tilde{\rho}(\underline{\mathbf{z}},0) = \begin{cases} 0\,, & \text{if } \underline{\mathbf{z}} \in Z_0\,, \\ \theta(\underline{\mathbf{z}})\,, & \text{if } \underline{\mathbf{z}} \in Z_1\,, \end{cases} \qquad (5.14a)$$

and,

$$\tilde{\rho}(\underline{\mathbf{z}},1) = \begin{cases} |\,\theta(\underline{\mathbf{z}})\,|\,, & \text{if } \underline{\mathbf{z}} \in Z_0\,, \\ 0\,, & \text{if } \underline{\mathbf{z}} \in Z_1\,, \end{cases} \qquad (5.14b)$$

where,

$$\theta(\underline{\mathbf{z}}) = \frac{(1-p)Q(\underline{\mathbf{z}}\,|\,1) - pQ(\underline{\mathbf{z}}\,|\,0)}{Q(\underline{\mathbf{z}})}\,, \qquad (5.15)$$

and the sets Z_0 and Z_1 are defined as,

$$Z_0 \equiv \{\underline{\mathbf{z}} \mid pQ(\underline{\mathbf{z}}\,|\,0) > (1-p)Q(\underline{\mathbf{z}}\,|\,1)\}\,, \qquad (5.16a)$$

and,

$$Z_1 \equiv \{\underline{\mathbf{z}} \mid pQ(\underline{\mathbf{z}}\,|\,0) < (1-p)Q(\underline{\mathbf{z}}\,|\,1)\}\,. \qquad (5.16b)$$

The rate-distortion function $\tilde{R}(D)$ can be expressed in the following parametric form using (4.41) and (4.42),

$$D = \sum_{Z_1} \lambda(\underline{z}) Q(\underline{z}) \hat{P}(0) \theta(\underline{z}) e^{s\theta(\underline{z})} - \sum_{Z_0} \lambda(\underline{z}) Q(\underline{z}) \hat{P}(1) \theta(\underline{z}) e^{-s\theta(\underline{z})} \qquad (5.17a)$$

and,

$$\tilde{R} = sD + \sum_{\underline{z}} Q(\underline{z}) \log \lambda(\underline{z}) , \qquad (5.17b)$$

where,

$$\lambda(\underline{z}) = [\hat{P}(0) + \hat{P}(1) e^{-s\theta(\underline{z})}]^{-1} \qquad \text{for } \underline{z} \in Z_0 , \qquad (5.18a)$$

and,

$$\lambda(\underline{z}) = [\hat{P}(0) e^{s\theta(\underline{z})} + \hat{P}(1)]^{-1} \qquad \text{for } \underline{z} \in Z_1 . \qquad (5.18b)$$

The $\lambda(\underline{z})$'s should satisfy,

$$\sum_{Z_0} \lambda(\underline{z}) Q(\underline{z}) + \sum_{Z_1} \lambda(\underline{z}) Q(\underline{z}) e^{s\theta(\underline{z})} = 1 , \qquad (5.19a)$$

and,

$$\sum_{Z_0} \lambda(\underline{z}) Q(\underline{z}) e^{-s\theta(\underline{z})} + \sum_{Z_1} \lambda(\underline{z}) Q(\underline{z}) = 1 . \qquad (5.19b)$$

Since, $\hat{P}(0) + \hat{P}(1) = 1$, then

$$\lambda(\underline{z}) = \frac{1}{\hat{P}(0) + (1-\hat{P}(0)) e^{-s\theta(\underline{z})}} , \qquad \text{for } \underline{z} \in Z_0 , \qquad (5.20a)$$

and,

$$\lambda(\underline{z}) = \frac{1}{\hat{P}(0) e^{s\theta(\underline{z})} + (1-\hat{P}(0))} , \qquad \text{for } \underline{z} \in Z_1 . \qquad (5.20b)$$

Substituting the $\lambda(\underline{z})$'s of (5.20) into (5.19), we obtain,

$$\sum_{\underline{z}} \frac{Q(\underline{z})}{\hat{P}(0) + (1-\hat{P}(0)) e^{-s\theta(\underline{z})}} = 1 , \qquad (5.21a)$$

and,

$$\sum_{\underline{z}} \frac{Q(\underline{z}) e^{-s\theta(\underline{z})}}{\hat{P}(0) + (1-\hat{P}(0)) e^{-s\theta(\underline{z})}} = 1 . \qquad (5.21b)$$

Multiplying the first summation by $\hat{P}(0)$ and the second summation by $1-\hat{P}(0)$ and adding the results, we have,

$$c(0)\hat{P}(0) + c(1)(1-\hat{P}(0)) = \sum_{\mathbf{z}} Q(\mathbf{z}) = 1$$

where $c(0)$ is the summation of (5.21a) and $c(1)$ is the summation of (5.21b). We note that,

$$c(0) = 1 \quad <=> \quad c(1) = 1 \;.$$

In other words, whenever one of the constraints of (5.21) is satisfied, the other one is also automatically satisfied. Therefore, in order to find the rate-distortion function $\tilde{R}(D)$, we can first solve,

$$\sum_{\mathbf{z}} \frac{Q(\mathbf{z})}{\hat{P}(0) + (1-\hat{P}(0))e^{-s\,\theta(\mathbf{z})}} = 1 \;, \tag{5.22}$$

for each $s<0$ to find $\hat{P}(0)$, then using (5.20), we can find the $\lambda(\mathbf{z})$'s, and finally substituting in (5.17), we can find the corresponding D and $\tilde{R}$.

5.1.2 Numerical Computation of the Rate-Distortion Function

In order to solve (5.22) numerically, we can use the following recursive method. Start with any $\hat{P}_0(0)>0$ and define,

$$c_n(0) = \sum_{\mathbf{z}} \frac{Q(\mathbf{z})}{\hat{P}_n(0) + (1-\hat{P}_n(0))e^{-s\,\theta(\mathbf{z})}} \;, \tag{5.23a}$$

and,

$$\hat{P}_{n+1}(0) = \hat{P}_n(0)c_n(0) \;. \tag{5.23b}$$

It is easy to verify the convergence of the above recursion. We note that, for any $0 < \hat{P}_n(0) < 1$,

$$\hat{P}_n(0) < \hat{P}_{n+1}(0) < 1 \ , \qquad \text{if} \quad c_n(0) > 1 \ ,$$

and,

$$0 < \hat{P}_{n+1}(0) < \hat{P}_n(0) \ , \qquad \text{if} \quad c_n(0) < 1 \ .$$

Thus starting from any $0 < \hat{P}_0(0) < 1$, the sequence $\hat{P}_n(0)$ remains bounded. Due to the Bolzano-Weierstrass theorem [48], any bounded sequence has at least one limit point and, therefore, the bounded (and monotonic) sequence $\hat{P}_n(0)$ has a limit point, say $\hat{P}^*(0)$. Combining (5.23a) and (5.23b), we obtain,

$$\hat{P}_{n+1}(0) = \hat{P}_n(0) \sum_{\underline{z}} \frac{Q(\underline{z})}{\hat{P}_n(0) + (1-\hat{P}_n(0))e^{-s\,\theta(\underline{z})}} \ , \tag{5.24}$$

in the limit, we have,

$$\hat{P}^*(0) = \hat{P}^*(0) \sum_{\underline{z}} \frac{Q(\underline{z})}{\hat{P}^*(0) + (1-\hat{P}^*(0))e^{-s\,\theta(\underline{z})}} \ ,$$

or,

$$\sum_{\underline{z}} \frac{Q(\underline{z})}{\hat{P}^*(0) + (1-\hat{P}^*(0))e^{-s\,\theta(\underline{z})}} = 1 \ ,$$

which establishes the convergence of the recursive procedure.

Therefore, in order to find a point on the rate-distortion function for any $s < 0$, we start with some $\hat{P}_0(0) > 0$ and iteratively use (5.23a) and (5.23b) until $c_n(0)$ is less than some predetermined threshold ϵ. Then, substituting the value of $\hat{P}_n(0)$ of the last iteration into (5.18), we can find $\lambda(\underline{z})$ and, finally, using (5.17a) and (5.17b), we can calculate the value of D and $\tilde{R}$.

5.2 Two Equiprobable Classes and Symmetric Features

Here, we consider two-class pattern recognition problem with equal <u>a priori</u> probabilities, i.e., $P(X=0) = P(X=1) = \frac{1}{2}$. Furthermore, we assume that the feature vector $\underline{\mathbf{z}}$ is the input x seen through k independent binary symmetric channels (BSC's) with crossover probabilities $q_1, \ldots, q_k$. This means that the feature vector $\underline{\mathbf{z}}$ can be expressed as the binary k-tuple $(z_1, \ldots, z_k)$ where z_j, $j \in \{1, \ldots, k\}$ is the output of a binary symmetric channel with crossover probability q_j. Each of the binary symmetric channels, say for example the jth one, can be described in terms of the diagram of Figure 5.1. The output of this BSC

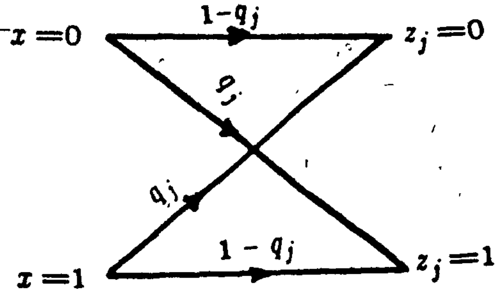

Figure 5.1: A Binary Symmetric Channel.

is different from its input with probability q_j, i.e., $P(z_j=1 \mid x=0) = P(z_j=0 \mid x=1) = q_j$, and, consequently, $P(z_j=1 \mid x=1) = P(z_j=0 \mid x=0) = 1 - q_j$. Since the BSC's are considered to be independent, the conditional probability of $\underline{\mathbf{z}}$ given x can be written as,

$$Q(\underline{\mathbf{z}} \mid x=0) = \prod_{j=1}^{k} q_j^{\,1-z_j} (1-q_j)^{z_j} , \tag{5.25a}$$

and,

$$Q(\underline{\mathbf{z}} \mid x=1) = \prod_{j=1}^{k} q_j^{\,z_j} (1-q_j)^{1-z_j} , \tag{5.25b}$$

with $Q(\underline{\mathbf{z}})$ given by,

$$Q(\underline{\mathbf{z}}) = \frac{1}{2} Q(\underline{\mathbf{z}} \mid 0) + \frac{1}{2} Q(\underline{\mathbf{z}} \mid 1) . \tag{5.26}$$

The distortion measure $\hat{\rho}(\underline{z},\hat{x})$ of (5.10) can be written as,

$$\hat{\rho}(\underline{z},\hat{x}) = \begin{cases} \dfrac{Q(\underline{z}\,|\,1)}{2Q(\underline{z})}\,, & \hat{x}=0\,, \\[2ex] \dfrac{Q(\underline{z}\,|\,0)}{2Q(\underline{z})}\,, & \hat{x}=1\,, \end{cases} \tag{5.27}$$

and,

$$\hat{\rho}(\underline{z}) = \min[\hat{\rho}(\underline{z},0)\,,\,\hat{\rho}(\underline{z},1)] = \frac{\min[Q(\underline{z}\,|\,1)\,,\,Q(\underline{z}\,|\,0)]}{2Q(\underline{z})}\,. \tag{5.28}$$

Therefore, $D_{\min}$ is found to be,

$$D_{\min} = \frac{1}{2}\sum_{\underline{z}}\min[Q(\underline{z}\,|\,1)\,,\,Q(\underline{z}\,|\,0)]\,. \tag{5.29}$$

Also, from (5.12) we have,

$$D_{\max} = \frac{1}{2}\,.$$

Let $\bar{\underline{z}}$ be the complement of the feature vector $\underline{z}$, i.e., $\bar{\underline{z}}$ is a vector whose components are the complements of those of $\underline{z}$. From (5.25) we have,

$$Q(\bar{\underline{z}}\,|\,0) = Q(\underline{z}\,|\,1)\,, \tag{5.30a}$$

and,

$$Q(\bar{\underline{z}}\,|\,1) = Q(\underline{z}\,|\,0)\,. \tag{5.30b}$$

and therefore,

$$Q(\bar{\underline{z}}) = Q(\underline{z})\,. \tag{5.31}$$

Substituting $p=\frac{1}{2}$ into (5.14), we have,

$$\tilde{\rho}(\underline{z},0) = \begin{cases} 0\ , & \text{if } Q(\underline{z}\,|\,0) > Q(\underline{z}\,|\,1)\ , \\ \theta(\underline{z})\ , & \text{if } Q(\underline{z}\,|\,0) < Q(\underline{z}\,|\,1)\ , \end{cases} \tag{5.32a}$$

and,

$$\tilde{\rho}(\underline{z},1) = \begin{cases} |\,\theta(\underline{z})\,|\ , & \text{if } Q(\underline{z}\,|\,0) > Q(\underline{z}\,|\,1)\ , \\ 0\ , & \text{if } Q(\underline{z}\,|\,0) < Q(\underline{z}\,|\,1)\ , \end{cases} \tag{5.32b}$$

where,

$$\theta(\underline{z}) = \frac{Q(\underline{z}\,|\,1) - Q(\underline{z}\,|\,0)}{2Q(\underline{z})}\ . \tag{5.33}$$

Note that,

$$\begin{aligned} \theta(\underline{\bar{z}}) &= \frac{(Q(\underline{\bar{z}}\,|\,1) - Q(\underline{\bar{z}}\,|\,0))}{2Q(\underline{\bar{z}})} \\ &= \frac{(Q(\underline{z}\,|\,0) - Q(\underline{z}\,|\,1)}{2Q(\underline{z})} = -\theta(\underline{z})\ . \end{aligned} \tag{5.34}$$

and also,

$$Q(\underline{z}\,|\,0) > Q(\underline{z}\,|\,1) \implies Q(\underline{\bar{z}}\,|\,1) > Q(\underline{\bar{z}}\,|\,0)\ , \tag{5.35a}$$

and,

$$Q(\underline{z}\,|\,1) > Q(\underline{z}\,|\,0) \implies Q(\underline{\bar{z}}\,|\,0) > Q(\underline{\bar{z}}\,|\,1)\ . \tag{5.35b}$$

Therefore,

$$\tilde{\rho}(\underline{\bar{z}},0) = \tilde{\rho}(\underline{z},1)\ , \tag{5.36a}$$

and,

$$\tilde{\rho}(\underline{\bar{z}},1) = \tilde{\rho}(\underline{z},0)\ . \tag{5.36b}$$

The symmetry existing in this case permits us to find an explicit expression for the rate-distortion function in terms of the parameter s. In fact, because of the symmetry and the fact that $p = \frac{1}{2}$, we conclude that $\hat{P}(0) = \hat{P}(1) = \frac{1}{2}$ for all values of $s > -\infty$. To verify this conclusion, substitute $\hat{P}(0) = \hat{P}(1) = \frac{1}{2}$ into the left hand side of (5.21a) and (5.21b) and denote the summations by $c(0)$ and $c(1)$,

$$c(0) = 2\sum_{\underline{z}} \frac{Q(\underline{z})}{1 + e^{-s\theta(\underline{z})}} , \tag{5.37a}$$

and,

$$c(1) = 2\sum_{\underline{z}} \frac{Q(\underline{z})e^{-s\theta(\underline{z})}}{1 + e^{-s\theta(\underline{z})}} . \tag{5.37b}$$

Now, it is easy to demonstrate that $c(0) = c(1) = 1$. First, adding (5.37a) and (5.37b) we obtain,

$$c(0) + c(1) = 2\sum_{\underline{z}} Q(\underline{z}) = 2 . \tag{5.38}$$

Then notice that complementing each $\underline{z}$ maps the set of k-tuples into itself, a change of variable $\underline{z} \rightarrow \overline{\underline{z}}$ in either (5.37a) or (5.37b) is merely equivalent to changing the order of summation and does not change the value of summations. Performing this change of variable in (5.37a), we obtain,

$$c(0) = 2\sum_{\underline{z}} \frac{Q(\overline{\underline{z}})}{1 + e^{-s\theta(\overline{\underline{z}})}} , \tag{5.39}$$

and using (5.31) and (5.34), we have,

$$\begin{aligned} c(0) &= 2\sum_{\underline{z}} \frac{Q(\underline{z})}{1 + e^{s\theta(\underline{z})}} \\ &= 2\sum_{\underline{z}} \frac{Q(\underline{z})e^{-s\theta(\underline{z})}}{1 + e^{-s\theta(\underline{z})}} = c(1) . \end{aligned} \tag{5.40}$$

Combining (5.38) and (5.40), we obtain $c\,(0) = c\,(1) = 1$. Therefore, $\hat{P}\,(0) = \hat{P}\,(1) = \frac{1}{2}$ is the the solution to the optimization problem for all values of s.

Now, to find the explicit form of the rate-distortion function $\tilde{R}\,(D\,)$, first we substitute $\hat{P}\,(0) = \hat{P}\,(1) = \frac{1}{2}$ into (5.18) to find $\lambda(\underline{z})$ as,

$$\lambda(\underline{z}) = 2\left[e^{s\tilde{\rho}(\underline{z},0)} + e^{s\tilde{\rho}(\underline{z},1)}\right]^{-1}, \tag{5.41}$$

and then substituting $\lambda(\underline{z})$ into (5.17a) and (5.17b), we find the expressions for D and $\tilde{R}$. Substituting (5.41) into (5.17a), we have,

$$D = \sum_{\underline{z}} \frac{Q\,(\underline{z})\,\tilde{\rho}(\underline{z},0)e^{s\tilde{\rho}(\underline{z},0)}}{e^{s\tilde{\rho}(\underline{z},0)} + e^{s\tilde{\rho}(\underline{z},1)} + \sum_{\underline{z}} \frac{Q\,(\underline{z})\,\tilde{\rho}(\underline{z},1)e^{s\tilde{\rho}(\underline{z},1)}}{e^{s\tilde{\rho}(\underline{z},0)} + e^{s\tilde{\rho}(\underline{z},1)}}}\,. \tag{5.42}$$

But, using the same argument as the one leading to (5.40), we obtain,

$$\sum_{\underline{z}} \frac{Q\,(\underline{z})\,\tilde{\rho}(\underline{z},1)e^{s\tilde{\rho}(\underline{z},1)}}{e^{s\tilde{\rho}(\underline{z},0)} + e^{s\tilde{\rho}(\underline{z},1)}} = \sum_{\underline{z}} \frac{Q\,(\bar{\underline{z}})\,\tilde{\rho}(\bar{\underline{z}},1)e^{s\tilde{\rho}(\bar{\underline{z}},1)}}{e^{s\tilde{\rho}(\bar{\underline{z}},0)} + e^{s\tilde{\rho}(\bar{\underline{z}},1)}}$$

$$= \sum_{\underline{z}} \frac{Q\,(\underline{z})\,\tilde{\rho}(\underline{z},0)e^{s\tilde{\rho}(\underline{z},0)}}{e^{s\tilde{\rho}(\underline{z},0)} + e^{s\tilde{\rho}(\underline{z},1)}},$$

and therefore,

$$D = 2\sum_{\underline{z}} \frac{Q\,(\underline{z})\,\tilde{\rho}(\underline{z},0)e^{s\tilde{\rho}(\underline{z},0)}}{e^{s\tilde{\rho}(\underline{z},0)} + e^{s\tilde{\rho}(\underline{z},1)}}\,. \tag{5.43}$$

Substituting $\tilde{\rho}(\underline{z},0)$ and $\tilde{\rho}(\underline{z},1)$ from (5.32) into (5.43), we have,

$$D = 2\sum_{Z_1} \frac{Q\,(\underline{z})\theta(\underline{z})e^{s\,\theta(\underline{z})}}{1 + e^{s\,\theta(\underline{z})}}\,, \tag{5.44}$$

where,

$$Z_1 \equiv \{\mathbf{z} \mid Q(\mathbf{z} \mid 0) < Q(\mathbf{z} \mid 1)\}\,. \tag{5.45}$$

To find the expression for $\tilde{R}$, we substitute (5.41) into (5.17b),

$$\tilde{R} = sD + \sum_{\mathbf{z}} Q(\mathbf{z})\log \frac{2}{e^{s\tilde{\rho}(\mathbf{z},0)} + e^{s\tilde{\rho}(\mathbf{z},1)}}\,, \tag{5.46}$$

or,

$$\tilde{R} = sD + 2\sum_{Z_1} Q(\mathbf{z})\log \frac{2}{1 + e^{s\theta(\mathbf{z})}}\,. \tag{5.47}$$

So, the rate-distortion function $\tilde{R}(D)$ can be evaluated using (5.44) and (5.47) for each value of s. To find $R(D)$, we need to shift $\tilde{R}(D)$ by $D_{\min}$, since, $R(D) = \tilde{R}(D - D_{\min})$.

We conclude this chapter with a following example to clarify the ideas discussed so far.

5.3 A Typical Example

We consider the example discussed by Gray and Chou[6]. We assume a memoryless source X with $P(X=0) = P(X=1) = \frac{1}{2}$, as seen through three independent binary symmetric channels with crossover probabilities $\frac{1}{3}$, $\frac{1}{4}$, and $\frac{1}{5}$. In the next chapter, we consider the more general case where the source is a binary symmetric Markov source rather than a memoryless source. Finding the exact $R(D)$ curve for the case of a binary symmetric Markov source is an extremely difficult task. In fact, the rate-distortion function for this source is not known even without extra interference, except for small values of D [49]. However, the rate-distortion found for the memoryless source can serve as an upper bound for the more complex case of the source with memory. Furthermore, using the Blahut's algorithm, discussed in the previous chapter, tighter bounds can be found. In the next chapter, we will use the Blahut's algorithm to find such bounds.

The feature vector $\underline{z}$ takes 8 values from 000 to 111. Table 5.1 shows the values of the probabilities $Q(\underline{z} \mid x)$, $Q(\underline{z})$,-the distortion measure $\hat{\rho}(\underline{z},\hat{x})$, and also $\hat{\rho}(\underline{z})$.

$\underline{z}$	000	001	010	011	100	101	110	111
$Q(\underline{z} \mid 0)$	$\frac{24}{60}$	$\frac{6}{60}$	$\frac{8}{60}$	$\frac{2}{60}$	$\frac{12}{60}$	$\frac{3}{60}$	$\frac{4}{60}$	$\frac{1}{60}$
$Q(\underline{z} \mid 1)$	$\frac{1}{60}$	$\frac{4}{60}$	$\frac{3}{60}$	$\frac{12}{60}$	$\frac{2}{60}$	$\frac{8}{60}$	$\frac{6}{60}$	$\frac{24}{60}$
$Q(\underline{z})$	$\frac{25}{120}$	$\frac{10}{120}$	$\frac{11}{120}$	$\frac{14}{120}$	$\frac{14}{120}$	$\frac{11}{120}$	$\frac{10}{120}$	$\frac{25}{120}$
$\hat{\rho}(\underline{z},0)$	$\frac{1}{25}$	$\frac{4}{10}$	$\frac{3}{11}$	$\frac{12}{14}$	$\frac{2}{14}$	$\frac{8}{11}$	$\frac{6}{10}$	$\frac{24}{25}$
$\hat{\rho}(\underline{z},1)$	$\frac{24}{25}$	$\frac{6}{10}$	$\frac{8}{11}$	$\frac{2}{14}$	$\frac{12}{14}$	$\frac{3}{11}$	$\frac{4}{10}$	$\frac{1}{25}$
$\hat{\rho}(\underline{z})$	$\frac{1}{25}$	$\frac{4}{10}$	$\frac{3}{11}$	$\frac{2}{14}$	$\frac{2}{14}$	$\frac{3}{11}$	$\frac{4}{10}$	$\frac{1}{25}$

Table 5.1: Values of the probabilities $Q(\underline{z} \mid \hat{x})$ and $Q(\underline{z})$, distortion measure $\hat{\rho}(\underline{z},\hat{x})$, and $\hat{\rho}(\underline{z})$.

We have $D_{\max} = \frac{1}{2}$ and from Table 5.1, we find,

$$D_{\min} = \sum_{\underline{z}} Q(\underline{z})\hat{\rho}(\underline{z}) = \frac{1}{6}$$

We can find $\tilde{\rho}(\underline{z},\hat{x}) = \rho(\underline{z},\hat{x}) - \hat{\rho}(\underline{z})$, with the values of $\tilde{\rho}(\underline{z},\hat{x})$ tabulated in Table 5.2.

$\underline{z}$	000	001	010	011	100	101	110	111
$\tilde{\rho}(\underline{z},0)$	0	0	0	$\frac{10}{14}$	0	$\frac{5}{11}$	$\frac{2}{10}$	$\frac{23}{25}$
$\tilde{\rho}(\underline{z},1)$	$\frac{23}{25}$	$\frac{2}{10}$	$\frac{5}{11}$	0	$\frac{10}{14}$	0	0	0

Table 5.2: Values of the translated distortion measure $\tilde{\rho}(\underline{z},\hat{x})$.

Now, we can find the parametric expression for $\tilde{R}(D)$ by substituting these values of $\tilde{\rho}(\underline{z},\hat{x})$ into (5.43) and (5.46),

$$D = \frac{1}{60}\left[\frac{23\alpha}{1+\alpha} + \frac{2\beta}{1+\beta} + \frac{5\gamma}{1+\gamma} + \frac{10\sigma}{1+\sigma}\right], \tag{5.48}$$

and,

$$\tilde{R} = sD + \frac{1}{60}\left[25\log\frac{2}{1+\alpha} + 10\log\frac{2}{1+\beta} + 11\log\frac{2}{1+\gamma} + 14\log\frac{2}{1+\sigma}\right], \tag{5.49}$$

where $\alpha = e^{\frac{23}{25}s}$, $\beta = e^{\frac{2}{10}s}$, $\gamma = e^{\frac{5}{11}s}$, and $\sigma = e^{\frac{10}{14}s}$.

Since $R(D) = \tilde{R}(D - D_{\min}) = \tilde{R}(D - \frac{1}{6})$, the parametric expression for $R(D)$ is found to be,

$$D = \frac{1}{60}\left[\frac{23\alpha}{1+\alpha} + \frac{2\beta}{1+\beta} + \frac{5\gamma}{1+\gamma} + \frac{10\sigma}{1+\sigma}\right] + \frac{1}{6}, \tag{5.50}$$

and,

$$R = s\,(D-\frac{1}{6})+\frac{1}{60}\left[25\log\frac{2}{1+\alpha}+10\log\frac{2}{1+\beta}+11\log\frac{2}{1+\gamma}+14\log\frac{2}{1+\sigma}\right] . \qquad (5.51)$$

The rate-distortion function $R\,(D)$ as defined by (5.50) and (5.51) is shown in Figure 5.2.

5.4 Discussion

As stated earlier, a rate-distortion function of the type given in Figure 5.2 provides us with the relationship between the minimum average information conveyed to the decoder in order to be able to classify the patterns with an average probability of error not exceeding a certain value D. Since in the derivation of $R\,(D)$, we have not made any assumption on the structure of the encoder and decoder, the encoder is allowed to extract any function of the components of the feature vector and deliver it to the decoder. Now, use of a specific pattern recognition scheme which has certain structure is equivalent to imposing a constraint on the choice of the features used by the encoder and decoder and, therefore, we should expect that the given pattern recognition scheme would perform worse than that given by the $R\,(D)$ curve. For example, if we use a decision tree for pattern recognition, we have limited the choice of features for the encoder-decoder pair to certain permutations of the original features; thus, it comes of no surprise if the relation between the average length of the tree and the minimum average distortion be bounded away from the rate-distortion curve, even in the case of correlated patterns. In fact, for each point on the $R\,(D)$ curve, D represents the minimum average probability of error attainable when the feature vectors have gone through a feature compression process which allows only R bits per feature vector. Since the minimum probability of error is achieved through maximum likelihood or Bayesian methods, we conclude that the points on the rate-distortion curve correspond to the average Bayesian probability of error for a certain amount of feature compression. Now, if we use any scheme other than maximum likelihood, it translates itself, in terms of the formulation we have followed so far, into using a fidelity criterion other than the one dictated by (5.10). Therefore, the rate-distortion should be modified accordingly, employing the relationship between the new fidelity criterion and that of (5.10).

One alternative to Bayes decision, for example, is nearest neighbor pattern recognition. The use of a nearest neighbor scheme is equivalent to using another distortion measure, for example, Euclidean distance (Hamming distance in the case of binary features), instead of the distortion measure of (5.10). Cover and Hart[24] have shown that $D \leq D' \leq 2D(1-D)$, where D' and D are the average probability of error for the nearest neighbor and the Bayes decision, respectively. Therefore, the curve in Figure 5.2 can be be viewed as a lower bound for the rate-distortion for the nearest-neighbor pattern recognition scheme, i.e., $R_L(D') = R(D)$. Similarly, an upper bound can be found for $R(D')$ by drawing the curve defined by the points (R, D^*) where $D^* = 2D(1-D)$, i.e., $R_U(D') = R[2D(1-D)]$.

The fact that the points on the rate-distortion function correspond to the maximum likelihood decision rule leads to the logical and intuitively clear argument that since the most economical and informative feature is the decision function itself, it is ineffective to divide the pattern recognition problem into two separate phases, i.e, the search for a set of good features and then the construction of the decision function based on these features; therefore, the decision should be sought as a function of the original measurements and, if some reduction in the volume or some generalization of the initial data turns out to be possible, this possibility will unavoidably manifest itself during the construction of the simplest computational algorithm for the decision function[14]. As pointed in Chapter 2, in parametric pattern classification, this argument leads to the methodology adopted by Datta and Morgera[19], Morgera[50], and Soleymani and Morgera[51], in tackling the problem of feature selection in Gaussian pattern recognition, where instead of dealing with distance measures which are only indirectly related to the probability of error, the point of departure in selecting the features has been the expression for the probability of error itself.

Another point which is worth noting is the fact that the points on the rate-distortion curve can usually be achieved for large enough size of pattern set, n; therefore, it is advantageous to consider the patterns in block instead of considering each pattern separately. While, as we will show in the next chapter, this is true in the case of correlated patterns, where one can make use of the correlation by delaying the decision, it is also true in the case of independent patterns, however, the gain in the latter case is less appreciable. The use of block or vector recognition also enables us to use fractional rates, for example, decision trees of average length less than n, when a decision tree is used, or template sets of size less than 2^n, when template matching is used. We feel that these observations will prove important in future pattern recognition system developments.

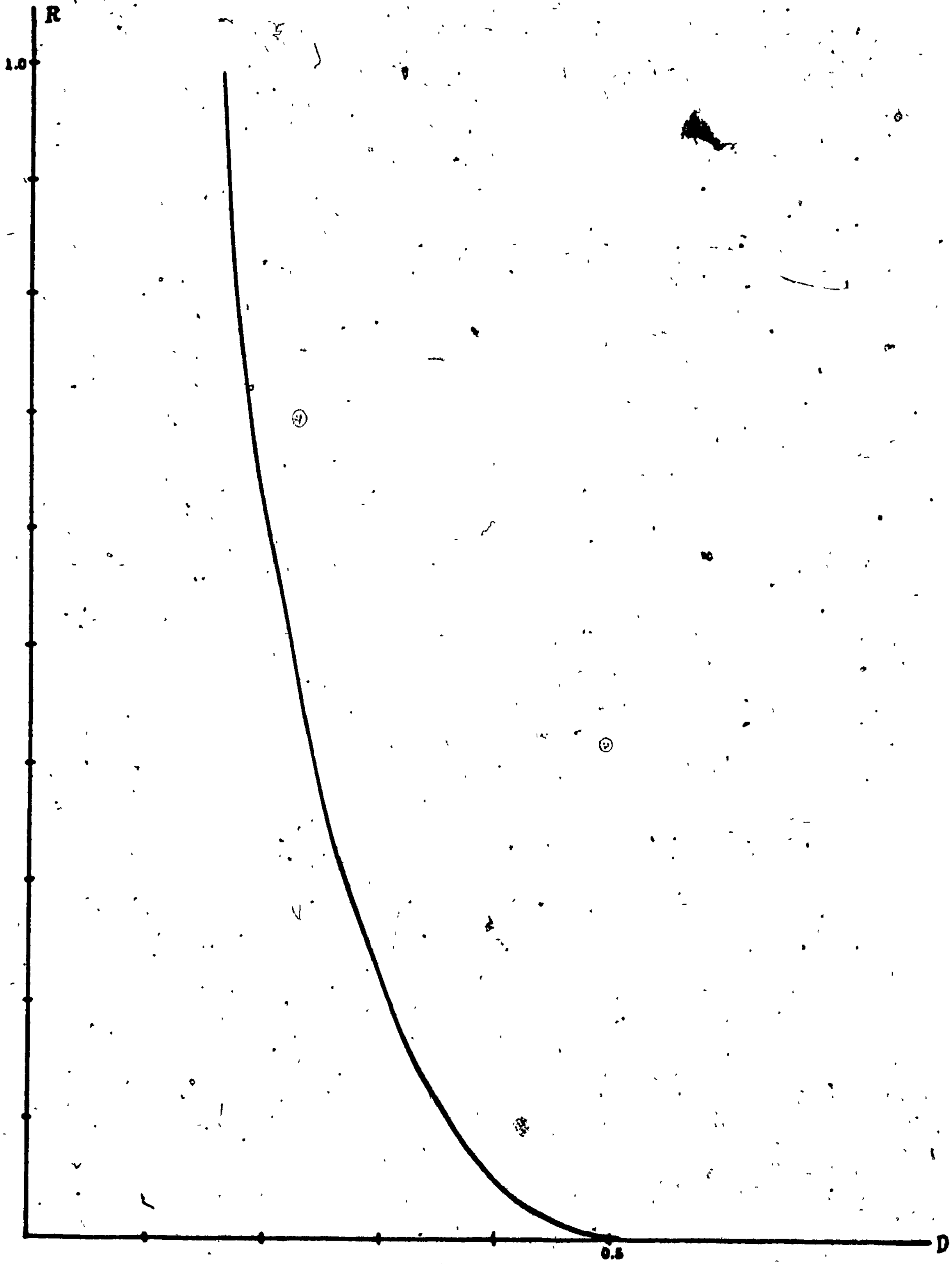

Figure 5.2 : The R(D) curve for the equiprobable memoryless source seen through three binary symmetric channels with crossover probabilities $\frac{1}{3}$, $\frac{1}{4}$, and $\frac{1}{5}$.

CHAPTER 6

Application of the Model : Correlated Patterns

Up to this point, we have considered the cases for which patterns have been uncorrelated, i.e., we have assumed that the probability that the present pattern belongs to a certain class does not depend on the classes to which the previous patterns belonged. In terms of the communications system model considered so far, it means that we have assumed the source X to be memoryless, i.e.,

$$P(x_n \mid x_{n-1}, x_{n-2}, \ldots) = P(x_n) . \tag{6.1}$$

However, this is not always true and usually consecutive patterns are correlated or, at least, it is more general to treat them as such. For example, it is conceivable that two consecutive patterns are more likely to belong to the same class than to two different classes. This implies a certain inertia in the state of nature, that is, it tends to stay in the same state (generating patterns belonging to the same pattern class) rather than moving to another state. The above mentioned situation gives a set of, say n, consecutive patterns a certain structure which may be used in deriving a more economical description of the pattern set.

6.1 Binary Symmetric Markov Source

It is a common practice to approximate sources with memory with Markov sources. In fact, any well-behaved stationary source can, at least approximately, be represented by a Markov source[15]. For an Lth order Markov source, we have,

$$P(x_n \mid x_{n-1}, x_{n-2}, \ldots) = P(x_n \mid x_{n-1}, x_{n-2}, \ldots, x_{n-L}) . \tag{6.2}$$

This means that the state of the system at any time instant n, only depends on its state in L previous time instants. The simplest Markov source is the 1st order Markov source for which we have,

$$P(x_n \mid x_{n-1}, x_{n-2}, \ldots) = P(x_n \mid x_{n-1}) . \tag{6.3}$$

In this chapter, we consider the same example as the one treated at the end of the previous chapter, but with the source being a first order binary symmetric Markov source. By computing the rate-distortion function for this example, for several dimensions(number of patterns treated together), we will show that considering the patterns <u>in block</u> results in a lower average probability of error for the same description cost(rate). Furthermore, we will calculate the rate-distortion function for the same binary symmetric Markov source, but with interference channels having different crossover probabilities and show that the more reliable the features, the less profound is the effect of memory.

A binary symmetric Markov source can be specified in terms of its transition probability q. Each output of the source is different from the previous output with probability q, i.e., $Pr(x_n \neq x_{n-1}) = q$. The source can be described using the following recursive expression,

$$X_n = X_{n-1} \oplus Y_n . \tag{6.4}$$

where Y_n are i.i.d. binary random variables assuming values 1 and 0 with probabilities q and $1-q$, respectively. Gray[49] has shown that for a small range of values of D, the rate-distortion for the source $\{X_n\}$ coincides with that of $\{Y_n\}$, i.e.,

$$R_X(D) = R_Y(D) = H_b(q) - H_b(D), \qquad D \le D_c, \tag{6.5}$$

where

$$H_b(x) = -x\log(x) - (1-x)\log(1-x), \tag{6.6}$$

and,

$$D_c = \frac{1}{2}\left[1-\sqrt{1-(\frac{m}{1-m})^2}\right], \tag{6.7}$$

where $m = \min(q, 1-q)$. For large values of D, the rate-distortion function for this source is unknown. However, good upper bounds to $R_X(D)$ can be found by numerically computing $R_n(D)$ for moderate values of n using Blahut's algorithm[46]. Using the above upper bound and a theorem due to Wyner and Ziv[52], a lower bound can be computed and, therefore, the rate-distortion of the binary symmetric Markov source can, at least numerically, be quite tightly bounded. Furthermore, explicit lower and upper bounds have been found by Berger[53] which are tight for certain ranges of values of D. However, as was pointed out, the exact value of the rate-distortion function for the values of $D > D_c$ remains unknown and is one of the challenging problems of information theory.

6.2 Performance of the Classifier versus the Number of Patterns

The above discussion concerning the extreme difficulty of finding the rate-distortion ~~function~~ for the binary symmetric Markov source itself discourages an effort to derive the analytical form of $R(D)$ for the more complex case where an extra interference is added at the source. Therefore, in the sequel, we try to compute the rate-distortion function $R_n(D)$ for certain values of n, using Blahut's algorithm. Fortunately, the curves tend to converge for values of $n \geq 4$ and therefore, $R_4(D)$ gives a reasonably tight upper bound to $R(D)$. This should suffice to serve our purpose, which is to conceptually justify the validity of the argument that considering several patterns at a time is more efficient than considering each pattern separately.

Figure 6.1 shows $R_n(D)$ for a binary symmetric Markov source with transition probability $q = \frac{1}{4}$ as seen through three BSC's with crossover probabilities $\frac{1}{3}$, $\frac{1}{4}$, and $\frac{1}{5}$, for the values of $n = 1,2,3,4$. Note that for each value of n, the source vector $\mathbf{x} = (x_1, \ldots, x_n)$ takes 2^n values. The same is true for the reproducing vectors $\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_n)$. However, the output of the BSC's, i.e., $\mathbf{z}$ takes 2^{3n} values, since, corresponding to each source letter x_i we have 2^3 values for the output of the BSC's.

From Figure 6.1, we notice that for a given rate, the distortion decreases with n. This observation supports the idea that the performance of the pattern recognition system is improved by treating the patterns in batch instead of considering them separately. This idea is closely allied with the ideas of compound decision theory[54]. This idea also leads to the idea of using global features instead of strictly local features. This may prove useful, for example, in image classification where one can benefit from using features defined over an image window rather a pixel. Examples of such features are average gray level or different "busyness" measures[55].

6.3 The Effect of the Quality of Features

Next, in order to investigate the effect of the quality of observations on the performance of the pattern recognition system, we calculate the rate-distortion function for the same binary symmetric Markov source with three different sets of BSC's. Figure 6.2 shows $R_n(D)$ for $n = 1,2,3$ for the binary Markov source with transition probability $q = \frac{1}{4}$ as seen through three sets of BSC's with crossover probabilities (0.10, 0.11, 0.12), (0.20, 0.22, 0.24), and (0.30, 0.33, 0.36). Apart from the obvious conclusion that the average probability of error for a given rate is higher for more degraded features (higher crossover probabilities), we notice that the amount of degradation in the average probability of error versus the rate is more appreciable for less degraded patterns. This observation implies that the complexity of the pattern recognition system should be proportional to the goodness of measured features. In other words, it is a waste to use a complex system (high rate) for very poor observations. We also notice that the vertical spacing of the rate-distortion curves for different values of n is more noticeable for more degraded feature sets. This shows that the effect of memory is more noticeable for "bad" features than for "good" features and, in turn, implies that heuristic decision rules based on the observation of long sequences of feature vectors are more effective when the features are quite degraded.

To clarify the above idea, we consider the following example. Enumerating the outputs of the interference channel, we notice that the number of one's in a set of feature vectors gives a good estimate of the class dependence of the patterns present at the input of the channel. In particular, if we consider the feature vectors separately, it is easily seen that those feature vectors, i.e., binary 3-tuples, with more one's than zero's have a higher <u>a posteriori</u> probability of belonging to the second class and, therefore, in this case a question of the form "Is k (the number of one's) greater than one ?" is sufficient for classifying the patterns based on the maximum likelihood decision rule. This classification strategy corresponds to the rate of one bit per pattern, since one dichotomy is required to be answered for classification of each pattern. Now, suppose that we base our decision on the observation of <u>two</u> consecutive feature vectors, i.e., six bits of the interference channel output. Because of the Markovian relationship between the patterns, it is more likely that two consecutive patterns belong to the same pattern class than to two different classes. So, we may use the following simple classification rule. If the number of one's in six bits is more than three we decide that both patterns belong to the second class (class 1), otherwise we assume that both patterns belong to class 0. Since, in this ad hoc scheme, only one question, i.e., "Is the number of ones greater than three?" is required to classify two consecutive patterns, the rate is only 0.5. Simple calculation shows that the average probability of error using this scheme is 0.133 for the case of BSC's with cross-over probabilities (0.10 , 0.11 , 0.12) and 0.278 for the case of (0.30 , 0.33 , 0.36). Since the minimum possible average probability of error is 0.033 and 0.238 for the first and second case, respectively, we observe that using this ad hoc classification method results in far less degradation in probability of error in the case of "bad" features than in the case of "good" features.

6.4 Vector Pattern Recognition

In section 6.2, computing the rate-distortion function for a typical source for different dimensions, we showed that improvement in the classifier performance is possible through increasing the number of patterns considered at a time. Also, as mentioned in chapter 5, treating patterns in block enables us to use fractional rates. These observations parallel those suggesting the use of vector quantization in source coding problems, e.g., in speech and image processing. Therefore, it is natural to think of vector pattern recognition, as an alternative to conventional pattern-by-pattern classification. Here, instead of making a decision $\hat{x}$ after observing a particular pattern $\underline{z}$, a compound decision $(\hat{x}_1, \ldots, \hat{x}_n)$ is made based on the observation of n consecutive patterns $\underline{z}_1, \ldots, \underline{z}_n$). For example, a condensing algorithm, such as the one discussed in chapter 3 can be used to derive compound templates based on a training set consisting of a sufficiently large set of nk-dimensional vectors each formed by concatenating n consecutive k-dimensional feature vectors. Then each of nk-dimensional templates are labeled with one of M^n possible combinations of M classes. In decision phase, each new compound pattern is assigned the label of the compound template closest to it.

In order to make this idea clear and also show the effectiveness of vector recognition, we work out a simple example. We consider the example of Section 3.2, i.e., two-class problem with class conditional densities $p(\underline{z} \mid x_i) = N(\underline{\mu}_i, \Sigma_i)$, $i=1,2$ where $\underline{\mu}_1=(1, 1, 1, 1)^T$ and $\underline{\mu}_2=(-1, -1, -1, -1)^T$ and Σ_1 and Σ_2 are 4×4 covariance matrices of first order Markov with correlation coefficients $\rho_1=0.3$ and $\rho_2=0.7$, respectively. However, here, we assume that the classes to which the patterns belong form a binary Markov sequence with a transition probability of $\frac{1}{4}$; that is, the probability that two consecutive patterns belong to two different classes is $\frac{1}{4}$. We take $n=2$, i.e., we make decisions based on the observation of two neighboring patterns. Using the condensing algorithm of chapter 3 and a sample set consisting of 20000, 8-dimensional compound feature vectors, we designed the five templates shown in

Table 6.1 together with their corresponding compound class labels. The probability of classification error using theses templates was found to be 7.52%. Comparing this value with the probability of error found for five templates in the case of single patterns (5th row of Table 3.1), a noticeable decrease in the probability of error is observed, while the complexity of the search is almost the same.

Template	Class
(0.974 , 1.032 , 1.115 , 1.071 , -1.602 , -1.701 , -1.754 , -1.598)	(1 , 2)
(1.558 , 1.028 , 0.517 , 0.664 , 1.714 , 1.807 , 1.191 , 0.624)	(1 , 1)
(-1.547 , -1.374 , -1.103 , -0.821 , -0.844 , -0.920 , -1.057 , -1.141)	(2 , 2)
(-0.319 , -0.385 , -0.523 , -0.631 , -1.605 , -1.673 , -1.717 , -1.642)	(2 , 2)
(-1.313 , -1.252 , -1.010 , -0.912 , 0.828 , 1.199 , 1.542 , 1.537)	(2 , 1)

Table 6.1 : Templates for the two-dimensional vector recognizer.

While the above example demonstrates the advantage of vector recognition over simple pattern-by-pattern classification, some problems remain which should be taken into consideration. The number of possible decisions M^n increases exponentially with n . Therefore, considering a large number of patterns at a time makes the search process demanding in terms of the number of operations required. Also, in order to design a reliable classifier for higher dimensions, we require a very large training set. The first problem can be solved for moderate values of n by using fast search algorithms such as those presented in Chapter 3. Also, it is possible to use a tree structure, similar to tree-searched vector quantization[3], where at each node of the tree only one decision is made based on the proximity of the given compound pattern to one of the two reference templates associated with that node. To solve the second problem, it is possible to first design a set of templates based on an initial sample set which is not extremely large and later modify the templates according to their performance on new patterns, for example, using the stochastic approximation method[56]-[57].

For large n, it is possible to reduce the number of templates by assigning templates only to those decision sequences which have more effect on the probability of error, and, therefore, discarding some of M^n sequences. Selection of sequences may be based either on the properties of good codes for the given source with memory which models the state of nature or can be done through the computation of the rate-distortion function for this source.

6.5 Discussion

In this chapter, first, we applied the rate-distortion theoretic model to a special case of correlated patterns where the state of nature was modeled as a binary symmetric Markov source. Based on the computation of the rate-distortion function for different number of patterns, it was observed that it is possible to reduce the probability of classification error by classifying several patterns concurrently. Based on this observation the idea of vector pattern recognition, i.e, treating the patterns in block instead of separately, was introduced and its effectiveness was demonstrated using a typical example. Also, by computing the rate-distortion function for a given source, but with different degrees of interference, it was shown that the gain achieved by increasing the number of patterns depends on the quality of the features, i.e., the less reliable the features, the greater the role played by the memory.

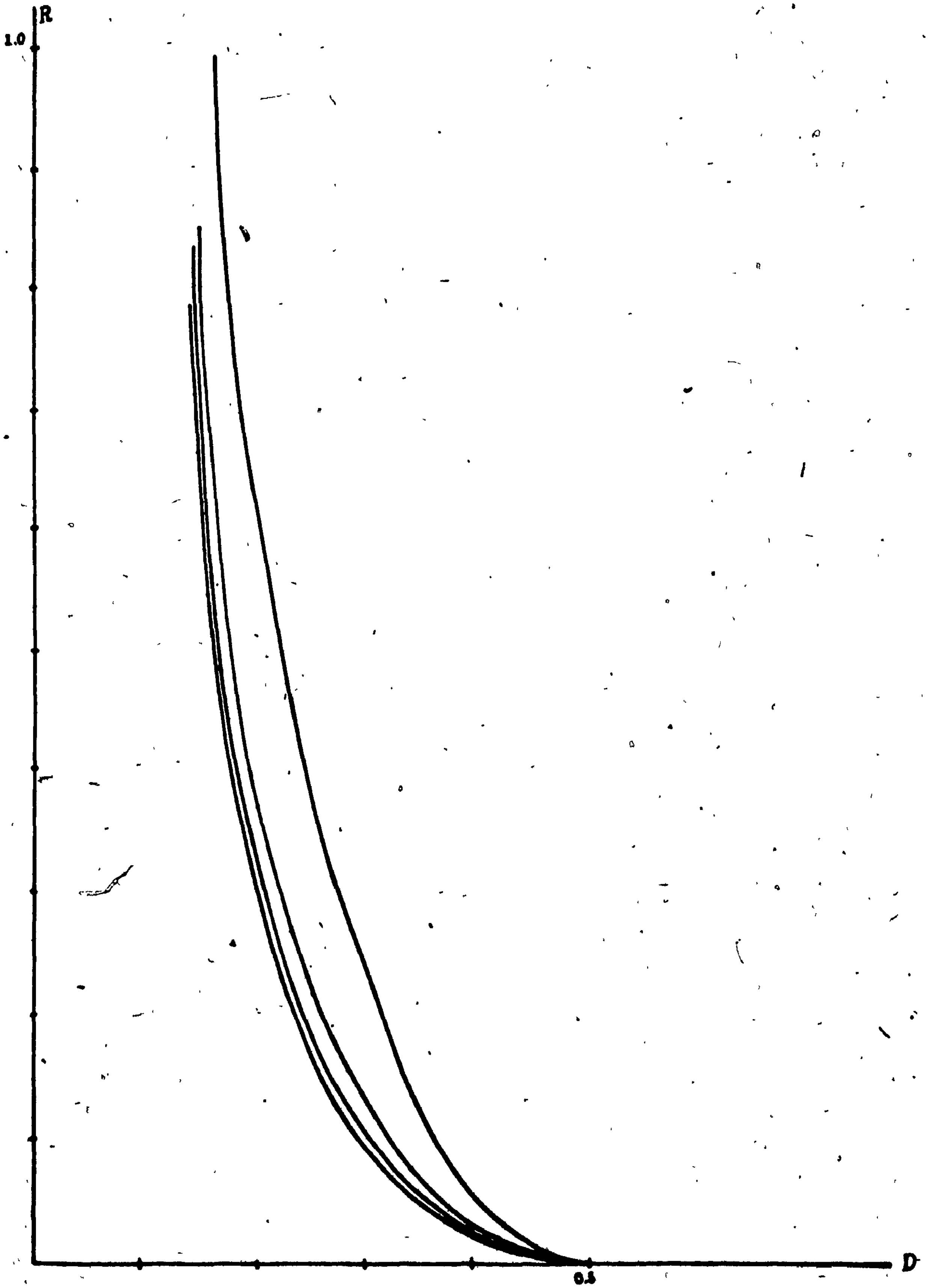

Figure 6.1 : $R_n(D)$, $n=1,2,3,4$ for a binary Markov source with transition probability 1/4 seen through three binary symmetric channels with crossover probabilities 1/3, 1/4, and 1/5.

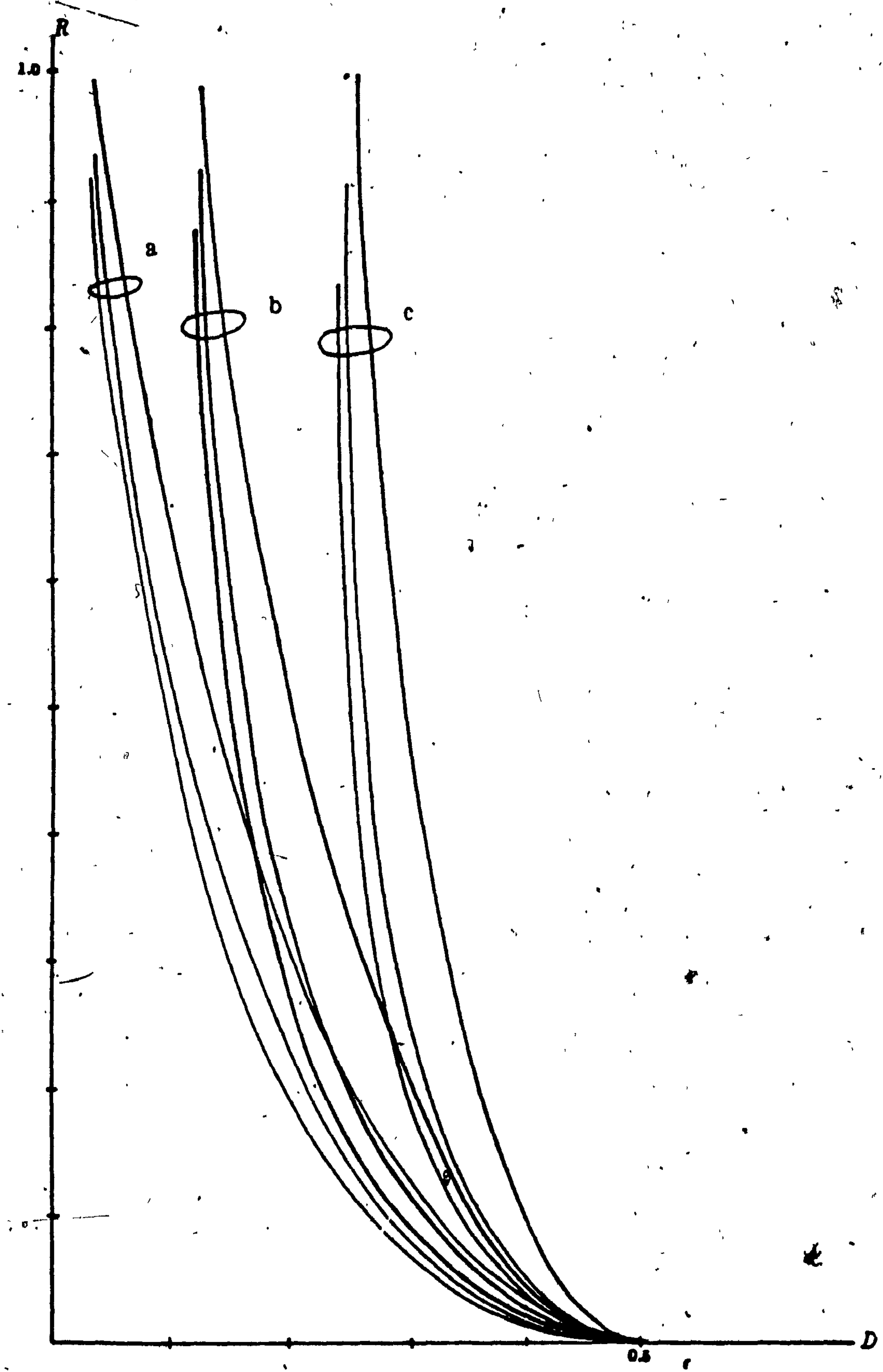

Figure 6.2 : $R_n(D)$, $n=1,2,3$ for a binary Markov source with transition probability 1/4 seen through three binary symmetric channels with crossover probabilities (a): 0.10, 0.11, 0.12, (b): 0.20, 0.22, 0.24, and (c): 0.30, 0.33, 0.36.

CHAPTER 7

Conclusions and Directions for Further Research

As mentioned earlier, a main goal of this work was to discuss some of the intricacies of rate-distortion theory within the framework of pattern recognition. We set out to plan this work after coming to the realization that the pattern recognition problem can be viewed in terms of a communications channel model similar to that employed for deriving the rate-distortion function. In this context, the notion of discriminant functions and the idea of feature selection and extraction according to various interclass distance measures fall into place. This is not to say that these problems are all solved if a rate-distortion view of the pattern recognition process is taken; on the contrary, as pointed out, few problems are solved and many still remain out of reach. A good example of the latter is the computation of the rate-distortion function for sources, or patterns, possessing correlation. This represents one of the open problems of information theory, the solution to which is crucial to the development and implementation of practical "Vector Recognizers", or pattern recognition systems designed according to the principles of rate-distortion theory. Just as vector quantization, based on rate-distortion theory, has led to innumerable advances and discussions within the speech processing research community in the last decade, we hope that the notion of vector recognition will spur thought and discussion within the pattern recognition research community. We dare to go further and suggest that rate-distortion theory coupled with advances for correlated sources may well explain and quantify recognition performance improvement for isolated versus strings of symbols in character and object recognition problems. If this is so, and only time and hard work will tell, the framework of rate-distortion theory may indeed impact the way in which we view problems in computer vision and artificial intelligence.

7.1 Directions for Further Research

There are several directions, related to this work, which can be the subject of further exploration in the future. Some of the most important ones are as follows:

1- Applying the proposed communication system model with different distortion measures in order to compare the suitability of different distance measures for nearest neighbor search. This can also lead to nonparametric feature selection criteria for a given distance measure.

2- Further examination of tree structures in order to reduce the complexity of vector recognition.

3- Use of stochastic approximation methods in the design of templates for vector recognition systems.

4- Study of the combinatorial properties of sequences suitable for coding sources with memory.

REFERENCES

[1] M. Bongard, *Pattern Recognition,* (English Translation by T. Cheron, Editor J.K. Hawkins), Spartan Books, 1970.

[2] Chi-hau Chen, *Statistical Pattern Recognition,* Spartan Books, 1973.

[3] S. Gulasu, *Information Theory with Applications,* McGraw-Hill Book Company, London, 1977.

[4] J. Pearl, "An Application of Rate-Distortion Theory to Pattern Recognition and Classification", *Pattern Recognition,* vol. 8, pp. 11-22, January 1976.

[5] A. Crolotte and J. Pearl, "Bounds on Memory Versus Error Trade-offs in Question-Answering Systems", *IEEE Transactions on Information Theory,* vol. IT-25, pp. 193-202, March 1979.

[6] P.A. Chou, and R.M. Gray, "On Decision Trees for Pattern Recognition", *1986 IEEE International Symposium on Information Theory,* Ann Arbor, MI, October 1986.

[7] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression,* Englewood Cliffs, N.J., Prentice-Hall, 1971.

[8] J.T. Tou, and R.C. Gonzalez, *Pattern Recognition Principles,* Addison-Wesley, 1981.

[9] C.K.Chow, "An Optimum Character Recognition System Using Decision Functions", *IRE Transactions on Electronic Computers,* vol. EC-6, pp. 247-254,

December 1957.

[10] R.O. Duda, and P.E. Hart, *Pattern Classification and Scene Analysis,* John Wiley and Sons, New York, 1973.

[11] T.Y. Young, and T.W. Calvert, *Classification, Estimation and Pattern Recognition,* American Elsevier Publishing Co., Inc., 1974.

[12] P.A. Devijver, and J. Kittler, *Pattern Recognition: A Statistical Approach,* Prentice-Hall International, Inc., London, 1982.

[13] R.L. Dobrushin, and B.S. Tsybakov, "Information Transmission with Additional Noise", *IRE Transactions on Information Theory,* vol. IT-8, pp. 293-304, September 1962.

[14] V.A. Kovalevsky, "Pattern Recognition: Heuristic or Science?", *Information Systems Science,* vol. 3, J.T. Tou, Ed., pp. 1-61, Plenum Press, New York, 1970.

[15] R.G. Gallager, *Information Theory and Reliable Communication,* John Wiley and Sons, Inc., New York, 1968.

[16] K. Fukunaga, and D.L. Kessel, "Application of Optimum Error-Reject Function", *IEEE Transactions on Information Theory,* vol. 18. pp. 814-817, Nov. 1972.

[17] A. Papoulis, *Probability, Random Variables, and Stochastic Processes,* McGraw-Hill Book Company, New York, 1984.

[18] C.R. Rao, *Linear Statistical Inference and its Applications,* John Wiley & Sons, New York, 1973.

[19] S.D. Morgera, and L. Datta, "Toward a Fundamental Theory of Optimal Feature Selection: Part I", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, pp. 601-616, September 1984.

[20] E. Parzen, "On Estimation of Probability Density Function and Mode", *The Annals of Mathematical Statistics*, vol. 33, pp. 1065-1076, September 1962.

[21] T.M. Cover, "A Hierarchy of Probability Density Function Estimates", *Frontiers of Pattern Recognition*, S. Watanabe, Ed., pp. 83-98, Academic Press, New York, 1972.

[22] E. Fix, and J.L. Hodges, Jr., "Discriminatory Analysis, Non-parametric Discrimination", USAF School of Aviation Medicine, Randolph Field, Tex., Project 21-49-004, Rept. 4, Feb. 1951, (Reprinted in *Machine Recognition of Patterns*, A.K. Agrawala Ed., IEEE Press, New York, 1977, pp. 261-279).

[23] E. Fix, and J.L. Hodges, Jr., "Discriminatory Analysis, Small Sample Performance", USAF School of Aviation Medicine, Randolph Field, Tex., Project 21-49-004, Rept. 11, Aug. 1952, (Reprinted in *Machine Recognition of Patterns*, A.K. Agrawala Ed., IEEE Press, New York, 1977, pp. 280-322).

[24] T.M. Cover, and P.E. Hart, "Nearest Neighbor Pattern Classification", *IEEE Transactions on Information Theory*, vol. IT-13, pp. 21-27, January 1967.

[25] P.A. Devijver, "New Error Bounds with the Nearest Neighbor Rule", *IEEE Transactions on Information Theory*, vol. IT-25, pp. 749-753, November 1979.

[26] J.H. Friedman, F. Baskett, and L.J. Shustek, "An Algorithm for Finding Nearest Neighbors", *IEEE Transactions on Computers*, vol. C-24, pp. 1000-1006,

October 1975.

[27] J.H. Friedman, J.L. Bentley, and R.A. Finkel, "An Algorithm for Finding Best Matches in Logarithmic Expected Time", *ACM Transactions on Mathematical Software,* vol. 3, pp. 209-224, September 1977.

[28] K. Fukunaga, and P.M. Narendra, "A Branch and Bound Algorithm for Computing k-Nearest Neighbors", *IEEE Transactions on Computers,* vol. C-24, pp. 750-753, July 1975.

[29] P.E. Hart, "The Condensed Nearest Neighbor Rule", *IEEE Transactions on Information Theory,* vol. IT-14, pp. 515-516, May 1968.

[30] D.L. Wilson, "Asymptotic Properties of Nearest Neighbor Rules Using Edited Data", *IEEE Transactions on Systems, Man, and Cybernetics,* vol. SMC-2, pp. 408-420, July 1972.

[31] M.R. Anderberg, *Cluster Analysis for Applications,* Academic Press, New York, 1973.

[32] G.H. Ball, and D.J. Hall, "A Clustering Technique for Summarizing Multivariate Data", *Behavioural Science,* vol. 12, pp. 153-155, March 1967.

[33] R.M. Gray, "Vector Quantization", *IEEE ASSP Magazine,* vol. 1, No. 2, pp. 4-29, April 1984.

[34] Y. Linde, A. Buzo, and R.M. Gray, "An Algorithm for Vector Quantizer Design", *IEEE Transactions on Communications,* vol. COM-28, No. 1, pp. 84-95, January 1980.

[35] D.T. Lee, and F.P. Preparata, "Computational Geometry - A Survey", *IEEE Transactions on Computers,* vol. C-33, No. 12, pp. 1072-1101, Dec. 1984.

[36] M.R. Soleymani, and S.D. Morgera, "An Efficient Nearest Neighbor Search Method", *IEEE Transactions on Communications,* vol. COM-35, pp. 677-679, June 1987.

[37] D.Y. Cheng, A. Gersho, B. Ramamurthi, and Y. Shoham, "Fast Search Algorithms for Vector Quantization and Pattern Matching", *Proc. IEEE Intl. Conference on Acoustics, Speech, and Signal Processing,* vol. 1, pp. 9.11.1-9.11.4, Mar. 1985.

[38] Chang-Da Bei, and R.M. Gray, "An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization", *IEEE Transactions on Communications,* vol. COM-33, No. 10, pp. 1132-1133, Oct. 1985.

[39] M.R. Soleymani, and S.D. Morgera, "A High-speed Search Algorithm for Vector Quantization", *Proc. IEEE Intl. Conference on Acoustics, Speech, and Signal Processing,* pp. 45.6.1-45.6.3, April 1987.

[40] T.R. Fishcer, "A Pyramid Vector Quantizer", *IEEE Transactions on Information Theory,* vol. IT-32, pp. 568-583, July 1986.

[41] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, *Data Structures and Algorithms,* Addison Wesley, 1983.

[42] M.R. Soleymani, and S.D. Morgera, "A Fast MMSE Encoding Technique for Vector Quantization", Submitted to *IEEE Transactions on Communications.*

[43] C.E. Shannon, "A Mathematical Theory of Communication", *Bell Syst.*

Tech. J., vol. 27, pp. 379-423, 623-656, 1948.

[44] C.E. Shannon, "Coding Theorems for a Discrete Source with a Fidelity Criterion", *IRE Nat'l. Conv. Rec.*, part 4, pp. 142-163, 1959.

[45] A.J. Viterbi and J.K. Omura, *Principles of Digital Communications and Coding,* McGraw-Hill, 1979.

[46] R.E. Blahut, "Computation of Channel Capacity and Rate-Distortion Functions", *IEEE Transactions on Information Theory,* vol. IT-18, pp. 460-473, July 1972.

[47] J.K. Wolf and J. Ziv, "Transmission of Noisy Information to A Noisy Receiver with Minimum Distortion", *IEEE Transactions on Information Theory,* vol. IT-16, pp. 406-411, July 1970.

[48] K. Knopp, *Theory and Application of Infinite Series,* English Edition, Blackie & Son, Ltd., Glasgow, 1961.

[49] R.M. Gray, "Information Rates of Autoregressive Processes", *IEEE Transactions on Information Theory,* vol. IT-16, pp. 412-421, July 1970.

[50] S.D. Morgera, "Toward a Fundamental Theory of Optimal Feature Selection: Part II-.Implementation and Computational Complexity", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-9, pp. 29-38, January 1987.

[51] M.R. Soleymani and S.D. Morgera, "A Direct Approach to Optimal Feature Selection in Pattern Recognition", under preparation.

[52] A.D. Wyner and J. Ziv, "Bounds on the Rate-Distortion Function for Sta-

tionary Sources with Memory", *IEEE Transactions on Information Theory,* vol. IT-17, pp. 508-513, September 1971.

[53] T. Berger, "Explicit Bounds to R(D) for a Binary Symmetric Markov Source", *IEEE Transactions on Information Theory,* vol. IT-23, pp. 52-59, January 1977.

[54] K.S. Fu, and T.S. Yu, *Statistical Pattern Classification using Contextual Information,* John Wiley and Sons, New York, 1980.

[55] P.A. Donde, and A. Rosenfeld, "Pixel Classification Based on Gray Level and Local "Busyness"", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-4, No. 1, pp. 79-84, January 1982.

[56] J.R. Blum, "Multidimensional Stochastic Approximation Methods", *The Annals of Mathematical Statistics,* vol. 25, pp. 737-744, 1954.

[57] D. Seret, and O. Macchi, "Automates Adaptatifs Optimaux", *Technique et Science Informatiques,* vol. 1, no. 2, pp. 143-153, 1982.