Every day as humans we classify objects as coming from certain classes from observations obtained through our sensual inputs including the visual system, auditory system, olfactory system, body pressure and temperature system, taste system, etc. For example as we walk down the street we easily classify moving vehicles as sports cars, compact cars, 4-door sedans, trucks, limousines, vans, and buses. A one year old infant easily recognizes the differences between basic objects like balls, books, kitties, dogs, and favorite toys, as well as recognizing and classifying many spoken words. Even at three months, most infants are able to tell the difference between a mother's face and faces of other people. Although these problems are easily solved using the human brain and sensory systems the same problems, in many cases, can represent significant tasks for computer systems. Infants can also handle the subtle differences in voice and recognize family members and friends and even more complex problems like recognizing changes in tone and mood of their mother's voice. How humans perform these pattern classification problems is not thoroughly understood.

Current theories and research on anatomy and physiology have suggested certain approaches that are not necessarily easily imitated on current computers. Special computer structures have been suggested for imitating the human system. At times the results are placed in the framework of computers with various different types of processing and memory use. From analytical developments on pattern recognition systems, directions for understanding the human pattern recognition system have been presented. Also laboratory research work on simpler biological systems has been pursued to give clues to understanding
the human system and even suggest methods to solve the general pattern recognition problem. It is inherently seen in biological systems that learning is done at many different levels and that, perhaps, is the key to the pattern recognition problem.

The purpose of this book is not to try to imitate how a person makes decisions but present a basic framework including general methods for solving pattern recognition problems and present solutions to certain special types of pattern classification problems. Another main component of the book is to present various types of learning procedures to use for solving some special pattern recognition problems. Another important topic will be the presentation of various clustering methods for use in identifying and characterizing patterns.

In this chapter several different approaches to the pattern recognition problem will be introduced including the statistical approach, neural network approach and the syntactical approach. These approaches will then be studied carefully in successive chapters. The presentation of the statistical approach to the pattern recognition is first because the resulting structures found serve as motivation for adhoc discriminate function approaches and certain types of neural network structures that follow.

0.1 BASIC PATTERN RECOGNITION APPROACHES

0.1.1 Statistical Pattern Recognition

In the statistical approach probabilistic models for the classes are established along with apriori probabilities of the classes. The mathematical tools of hypothesis testing and estimation theory are used to establish optimum classification rules based on given types of performance indices. The basic advantages of this approach include the fact that the analytical development leads to the optimum structure. Also the mathematical solutions may give insight into new statistics or features that could be used to solve the problem. Structures for optimum systems may be suggested that can be used as basic systems for training pattern recognition systems to perform the pattern classification job. Disadvantages include that too much statistical information or unavailable statistical information may be required for the solution and even with all the statistical information the mathematics may be too complex to obtain an analytical solution (mathematically untractable). An example is presented indicating certain salient properties of the statistical pattern recognition approach.

EXAMPLE 0.1

The problem to be investigated is that of being able to examine a fruit (either an orange or a grapefruit) and determine whether it comes from the class of oranges or from the class of grapefruits by measuring its major diameter \(x\). After examining oranges and grapefruits for many years we have a good idea of the probability density functions for this diameter subject to each class. These probability density functions that may be estimated using estimation theory are illustrated in Figure 0.1. The densities show the fact that most grapefruits are larger than most oranges. It will be assumed that the apriori probabilities that a fruit under test is an orange and a grapefruit are known. Thus the problem of classifying the fruit can be posed as a hypothesis testing problem and a solution can be obtained in a Bayesian fashion. The mathematical framework and solution to this type of problems is the main emphasis of chapter 2.

0-2
0.1.2 Neural Pattern Recognition

The neural network approach to pattern recognition is based on the idea of training a specified structure to perform the given pattern classification job. The structure is assumed to be "rich enough" to contain the solution. Particular structures include feedforward and feedback nonlinear structures. The basic tools used include nonlinear systems theory and various types of training algorithms. A set of training samples representative of their respective classes is required in order to train the network to perform the pattern recognition on samples not necessarily present in the training set. Advantages of the neural network approach include being able to solve problems without statistical information or problems that with the statistical information given are too complex to obtain an analytical solution. A main disadvantage of the neural network approach is that it may take considerable computer time and memory, in some cases more than we might have available or be willing to use. Another disadvantage is that you may not have enough representative training samples that would allow the solution to provide the necessary generalization to non training patterns. Also the solution does not necessarily provide an understanding of why or how it works, simply gives you a solution. Furthermore changes of, or additions to the training set may require a retraining procedure thus essentially reworking the solution from the beginning. An example of a problem that could be solved using a neural network is given in Example 0.2.

EXAMPLE 0.2

The basic problem to be solved using the neural network approach is that of personal identification from utterances of a persons name. Let pattern class \(C_1\) be those name utterances coming from Tom, pattern class \(C_2\) be those coming from Dick and \(C_3\) be those coming from Harry. Assume that we have available a set of samples of these utterances from each individual as shown in Figure 0.2.

![Figure 0-1. Probability density functions for major diameters of oranges and grapefruit.](image-url)
As the spoken names are different depending upon the time of day, the physical
condition of the individual and the mood of the individual it would be very difficult to solve
the problem by a using a table lookup method. As the statistics of the signals may not be
known or difficult to determine, this problem is a candidate for using the neural network
approach. The neural net is trained using the representative signals given in the figure and
the resulting neural net will hopefully have the ability to generalize for signals that are not
in the training set.

**0.1.3 Syntactical Pattern Recognition**

The syntactical approach to a pattern recognition problem allows us to solve
problems that are not necessarily empirical in nature. The basic tools for the method involve
the use of formal language theory including different types of grammars and parsing
procedures. Required information to solve the problem is the specification of the grammars
by a set of production rules that characterize the languages associated with each class and
a way of resolving intersections in the languages. One of the main advantages of this
particular method is that non-quantitative information like structure and color can be used
and thus solutions can be obtained that could not be obtained using either the neural or
statistical approaches. However, disadvantages include the difficulty in formulating the class
grammars and the restrictiveness of the grammars that might be formulated. Problems with
highly quantitative data or where structural connections are not evident or present are unable
to be solved using the syntactical approach. Example 0.3 shows a problem that can be solved
using a syntactical pattern recognition approach.

**EXAMPLE 0.3**
Assume that there are two pattern classes, Class 1 and Class 2, which are sets of strings of symbols a, b, and c defined as follows: Strings in Class 1 are \{ ac, abc, abbc, abbbc, ..., \} and strings in Class 2 are \{ bc, abc, aabc, aaabc, ..., \}. The problem would be to classify an arbitrary string of symbols containing a, b, and c into one of the two classes mentioned above. Each of the classes given has a language generated by its grammar. To determine which class the given sequence belongs becomes a matter of parsing the sequence of symbols under test. It is easily seen that in this case sequences like aabbcc would not be said to come from either of the two classes.

0.1.4 Other Pattern Recognition Approaches

Many pattern recognition problems are solved not by any of the general approaches just discussed but approaches that are tailored for the specific problem at hand. They usually rely on intuition and other subjective information about the problem and in no sense represent an optimal solution.

In other problems the available information is not of a quantitative type but may be said to be "fuzzy" in nature. There is a growing and successful area of research and basics involving the use of fuzzy set theory to develop pattern recognition systems. In this text fuzzy concepts will be used primarily in the clustering area (Chapter 6).

Complex pattern recognition problems may be better solved by using combinations of the methods described and we should be hesitant to try to force a particular "favored" method on all problems. It is also imperative that we do not lock ourselves into a particular structure of implementation or realization.

0.2 BASIC PATTERN RECOGNITION STRUCTURE

The basic pattern recognition structure that will be used in this text, shown in Figure 0.3, consists of a source, transducer, processor, and classifier. Of these the only one not in our control is the source, and if we knew what the source produced there would be no reason to have a pattern recognizer.

![Figure 0-3. Basic Pattern Recognition Structure](image)

0.2.1 Source

The source produces patterns, thought of as vectors in an n-dimensional or infinite
dimensional space, that belong to a number of different classes. Control of the source could be by an unknown or known person, random physical phenomenon or act of nature. For mathematical purposes it is assumed that the set of all possible patterns is \( S \) and that the classes makeup subsets of \( S \) call them \( S_i \), \( i = 1, 2, \ldots, N \) such that

\[
\bigcup_{i=1}^{N} S_i ^\prime \quad \text{(exhaustive)}
\]

\[
S_i \cap S_j ^\prime \quad \text{null set} \quad \emptyset \quad \text{(mutually exclusive)}
\]

(give example and talk more about sources)

0.2.2 Sensor

The sensor or transducer takes the pattern produced by the source and obtains information about the pattern which essentially represents our “image” of the source pattern. The sensor could be as simple as a voltmeter for measuring a voltage, a sampler and digitizer for performing an analog to digital conversion of a continuous source pattern or as complex as a spectrum analyzer for determining the frequency content of the source pattern. The sensor gives a value or more usually a vector of values as its output. It is hoped that the transducer is able to extract components that contain all the information about the source pattern necessary to make a decision. The output of the transducer is sometimes called an observed pattern vector, observed signal, or observed measurement vector. Sensor or transducer selection is up to the user and may be based on physical limitations regarding what can and cannot be measured and what instruments are available.

0.2.3 Processor

The processor takes the measurement vector or signal \( x \) and transforms it to a new vector \( y \) in the processed or feature space \( Y \). Usually this functionally produces a vector of reduced size that will facilitate the classification process. Selection of the types of processing are up to the user and may be based on statistical analysis, a sufficient statistic for example, or intuition about what is minimal information necessary to make a decision. The resulting space is sometimes called a feature space to emphasize that it attempts to use only the distinguishing features or properties of the pattern tested for recognition.

For example if \( x \) represents the observed pattern vector, \( y \) could be the vector composed of the magnitudes of the first two Fourier coefficients associated with \( x \). Thus the output of the processor displays extracted frequency content information.

The processor can be a linear operation as is the case of finding the average value of the components of the observed vector or nonlinear as for the case of extraction of average power for the observed pattern vector.
0.2.4 Classifier

The last and final block of Figure 0.4 is the **classifier** or “heart” of the pattern recognition system. It takes the feature vector \( y \) and assigns to it a decision \( d \) where \( d \) is the number of the pattern class which the pattern recognizer has decided. The mapping, called a **decision rule**, is user selectable, and is usually based on maximizing or minimizing some performance measure, however, in some problems it may even be intuitively based. The classifier partitions the feature space into subsets \( Y_i \) such that if \( y \) is a member of \( Y_i \) the assigned decision is \( d_i \), that is the class assigned is class \( C_i \). Similarly if the classifier works directly on the vectors in the observation space without going through a processor block, the partitioning is \( X_i \) with \( d_i \) for class \( C_i \). If the classification takes place in the feature space the classification rule will induce decision regions in the observation space. (discuss mapping from observation or feature space and give example)

0.3 PATTERN RECOGNITION EXAMPLE

Consider the problem of determining whether a handwritten symbol selected at random from a written sequence of symbols is a zero or a one. The source produces a handwritten symbol either a zero or a one. A scanner/quantizer (the sensor) converts this continuous image to an \( N \) by \( N \) matrix of values between 0 and 1 as shown in Figure 0-4. This matrix is then changed to an \( N^2 \) squared by 1 vector representing the observed pattern vector. Two different forms of processing, each producing a one dimensional feature space, will be presented to show the effect that the feature transformation can have on performance. The first form of processing uses a linear transformation while the second uses a nonlinear transformation. The choice and selection of the processing to obtain the feature vector is a very important part of pattern recognition problems and is discussed in Chapter 7.

![Figure 0-4. Feature space and decision rule for linear processing of the observation vector for the handwritten number recognition example.](image)
0.3.1 Linear Processing

In the first approach the feature is described by a sum of all entries in the observed vector, the idea behind which is that the 0 images would have more values close to one than the "skinny" 1 image. The resulting feature space would be the real line from 1 to $N^2$. The value of $y$ in the feature space would have different probability density functions depending on whether the source pattern presented was a zero or a one. It is reasonable to expect that the peak in the density for a 0 image have a peak further along the x axis than that for a 1 as shown in Figure 0-5.

![Conditional probability densities for the statistic y for a "0" and a "1" image.](image)

The decision rule for this problem could be selected as a simple threshold test: if the value of $y$ is greater than $T$ classify it as an image of a 0, if $y$ is less than $T$ classify it as an image of a 1, and if $y = T$ flip an honest coin with a head classifying it as an image of a one and a tail classifying it as an image of a zero. The $T$ might be selected to minimize the sum of the error probabilities indicated by the shaded areas. As $T$ moves to the right the error conditioned on a "1" decreases, however, the error conditioned on a "0" increases. Because of the considerable overlap of the conditional density functions the probability of error for this decision rule could be quite large.

0.3.2 Nonlinear Processing

The second approach is based on the observation that the spread of the column projection for an image of a zero is much bigger than spread of an image of a one as a 0 is usually much wider than a 1. The processing is described as follows.

$$y = \sum_{j=1}^{N} (\text{Proj}(x_j) \& \text{Avg})^2$$

where

$$\text{Avg} = \sum_{j=1}^{N} x_j \text{Proj}(x_j)$$
Typical projections for representative "0" and "1" images and the conditional densities of the spread $y$, defined above, are shown in Figure 0.6. It is noticed that there is less overlap in the densities than for the case of linear processing and thus the threshold test will result in a smaller value of incorrect classification than the first method described. The block diagram for the classifier that uses the spread, a nonlinear processing of the original images, is shown in Figure 0-7.

In general a good choice of nonlinear processing will usually give better performance unless a linear processor is optimum and in that case the two processors will give equal performances. Thus the choice of selecting the proper processing to create the feature space is perhaps the most important part of the pattern recognition problem. We can all think of extra features that could be used for this particular example which are based on intuition and past experience. If we use both the linear and nonlinear features described we would expect with the proper choice of the classifier to do no poorer and more than likely do better than the classifier that just uses one of them.
Research over the last ten years has applied more complex processing than the simple linear and nonlinear processing to the problem of character recognition given above producing much better results and extending the techniques to handwritten letters as well. Our purpose in this example was to show the improvement that usually comes with nonlinear processing rather than provide the best overall solution to the handwritten number problem. Guidelines for finding the "best" or optimum method are presented in the chapters that follow.

0.4 PATTERN RECOGNITION PERFORMANCE

The performance measure for pattern recognition problem depends upon the objective of our pattern recognition system and usually on a subjective assignment of "goodness". Obvious measures of performance include the aposteriori probability (after the fact probability), the probability of error, a the probability of correct decisions, and a weighted probability of error(called risk).

Suppose that there are \( K \) pattern classes called \( C_j \) for \( j = 1, 2, \ldots, K \) and that the vector \( \mathbf{x} \) is observed. We would like to assign \( \mathbf{x} \) to one of the classes by using a decision rule determined by a given performance measure.

0.4.1 Aposterior Probability

If we are given a pattern vector \( \mathbf{x} \), the aposteriori probability of each class \( C_j \) given \( \mathbf{x} \) can be calculated using the following form of Bayes rule

\[
P(C_j|\mathbf{x}) \propto \frac{p(\mathbf{x}|C_j) P(C_j)}{p(\mathbf{x})}
\]

where \( p(\mathbf{x}|C_j) \) is the probability density function of the pattern vector conditioned on the class \( C_j \), \( P(C_j) \) is the apriori probability of the class \( C_j \) and \( p(\mathbf{x}) \) the probability density function for the observation vector or the averaged density over all \( K \) classes given by

\[
p(\mathbf{x}) = \sum_{j=1}^{K} p(\mathbf{x}|C_j) P(C_j)
\]

It is intuitively pleasing to use a decision rule that selects the Class \( C_{k_0} \) which has the highest aposteriori probability or after the fact probability, \( P(C_{k_0}|\mathbf{x}) \), which is the performance index for the maximum aposteriori decision rule.

0.4.2 Probability of Error

Perhaps the most universal performance measure for the pattern recognition problem is the total probability of error. If \( P(\text{error}|C_j) \) represents the probability of deciding a class other than \( C_j \) given that the observation vector \( \mathbf{x} \) is from Class \( C_j \), the total probability of error can be obtained from the conditional errors as follows
Thus it is natural to look for a decision rules that minimize the total probability of error.

0.4.3 Probability of Correct Classification
The probability of correct classification is directly related to the misclassification probability as follows

\[ P(\text{Correct}) = 1 - P(\text{error}) \]

Thus maximizing the probability of correct classification is the same as minimizing the probability of misclassification error.

0.4.4 Average Cost per Decision (Risk)
In certain problems misclassifying a given pattern as coming from class C_k when it is actually from class C_j is not the same for all k. Calling an apple good when it is bad would cost more in lawyers than calling a good apple bad which would just be the cost of throwing away one apple. The consequences of this type of misclassification is thus assigned a cost C_{jk}. The average cost per decision, sometimes called risk is defined as follows

\[ \text{Risk} = \sum_{j=1}^{K} \sum_{k=1}^{K} C_{jk} P(\text{deciding } C_k | C_j) P( C_j) \]

The Bayes decision rule selects the decision regions in the pattern space such that the Risk given above is minimized.

0.5 CLUSTERING OF PATTERN VECTORS
In the previous sections we have been concerned with deciding which class an arbitrary pattern vector is from using the statistical information of the conditional density functions and the apriori class probabilities. In opposition to this part of pattern recognition is the topic of trying to determine from a set of vectors how many pattern classes we have and also which samples go with which class. Thus it is not the decision rule that drives the classes but the structure of the data that determines the classes. This process of grouping data into classes is called clustering. Thus clustering is a partitioning of the data into different groups where the different groups are in some sense self consistent. Self consistency or similarity can be defined in terms of various metrics in the pattern vector space. Common clustering algorithms that use metrics or similarity measures that will be explored in this book include the K-means, ISODATA, and Hierarchical clustering algorithms. Another important area of clustering involves the use of fuzzy sets and instead of giving a hard clustering cluster groups are given in terms of membership functions.
Chapter 0

0.0 Introduction ................................................................. 0-1

0.1 Basic Pattern Recognition Approaches ............................... 0-2
   0.1.1 Statistical Pattern Recognition .................................. 0-2
   0.1.2 Neural Pattern Recognition ...................................... 0-3
   0.1.3 Syntactical Pattern Recognition ................................. 0-4
   0.1.4 Other Pattern Recognition Approaches ........................ 0-5

0.2 Basic Pattern Recognition Structure ................................ 0-5
   0.2.1 Source .......................................................... 0-6
   0.2.2 Sensor ......................................................... 0-6
   0.2.3 Processor ...................................................... 0-6
   0.2.4 Classifier ...................................................... 0-6

0.3 Pattern Recognition Example ........................................ 0-7
   0.3.1 Linear Processing .............................................. 0-7
   0.3.2 Nonlinear Processing ......................................... 0-8

0.4 Pattern Recognition Performance .................................. 0-10
   0.4.1 Aposterior Probability ......................................... 0-10
   0.4.2 Probability of Error ......................................... 0-10
   0.4.3 Probability of Correct Classification ........................ 0-11
   0.4.4 Average Cost per Decision(Risk) ............................ 0-11

0.5 Clustering of Pattern Vectors ........................................ 0-12