Feature Selection in Statistical Pattern Recognition

by

Naruetep Choakjarnwanit

Thesis submitted to the University of Surrey for the degree of Doctor of Philosophy

Department of Electronic and Electrical Engineering
University of Surrey
Guildford, Surrey
United Kingdom
October 1992
Abstract

This thesis addresses the problem of feature selection in pattern recognition. A detailed analysis and an experimental comparison of various search strategies for selecting a feature set of size $d$ from $D$ available measurements are presented. For a realistic problem, optimal search, even if performed using the branch and bound search method, is computationally prohibitive. The alternative is to use suboptimal search methods. Of these, there are four methods, namely the sequential forward selection (SFS), sequential backward selection (SBS), sequential forward floating selection (SFFS), and sequential backward floating selection (SBFS), which are relatively simple and require little computational time. It is suggested that the SFS method should be employed in the case of limited training sample size. Although the decision about including a particular measurement in the SFS method is made on the basis of statistical dependencies among features in spaces of monotonically increasing dimensionality, the approach has proved in practice to be more reliable. This is because the algorithm utilizes at the beginning only less complex mutual relations which using small sample sets are determined more reliably than the statistics required by the SBS method. Because both the SFS and SBS methods suffer from the nesting effect, if better solution is required then the SFFS and SBFS should be employed.

As the first of the two main issues of the thesis, the possibility of developing feature selection techniques which rely only on the merit of individual features as well as pairs of features is investigated. This issue is considered very important because the computational advantage of such an algorithm exploiting only at most pairwise interactions of measurements would be very useful for solving feature selection problems of very high dimensionality. For this reason, a potentially very promising search method known as the Max-Min method is investigated. By means of a detailed analysis of the heuristic reasoning behind the method its weaknesses are identified. The first weakness is due to the use of upper limit on the error bound as a measure of effectiveness of a candidate feature. This strategy does not guarantee that selecting a candidate feature with the highest upper bound will yield the highest actual amount of additional information.
The second weakness is that the method does not distinguish between a strong unconditional dependence and a poor performance of a feature which both manifest themselves by a near zero additional discriminatory information. Modifications aimed at overcoming the latter by favouring features which exhibit conditional dependence and on the other hand suppressing features which exhibit strong unconditional dependence have been proposed and tested but only with a limited success. For this reason the Max-Min method is subjected to a detailed theoretical analysis. It is found that the key assumption underlying the whole Max-Min algorithm is not justified and the algorithm itself is ill-founded, i.e. the actual increment of the criterion value (or decrease of the probability of error) can be bigger than the minimum of pairwise error probability reductions assumed by the Max-Min method. A necessary condition for invalidity of the key assumption of the Max-Min algorithm is derived, and a counter-example proving the lack of justification for the algorithm is presented.

The second main issue of the thesis is the development of a new feature selection method for non-normal class conditional densities. For a given dimensionality the subset of selected features minimizes the Kullback-Leibler distance between the true and postulated class conditional densities. The algorithm is based on approximating unknown class conditional densities by a finite mixture of densities of a special type using the maximum likelihood approach. After the optimization ends, the optimal feature subset of required dimensionality is obtained immediately without the necessity to employ any search procedure. Successful experiments with both simulated and real data are also carried out to validate the proposed method.
To my father,
and the memory of my mother
Acknowledgements

I would like to express my deepest appreciation and gratitude to my research supervisor, Professor Josef Kittler, who so generously provided me with the invaluable guidance, and also for the warmth and the friendship that he shared throughout the course. I am truly grateful to Dr. Pavel Pudil and Dr. Jana Novovičová from Institute of Information Theory and Automation, Czechoslovak Academy of Sciences, who contributed both directly and indirectly to my experience in this area.

The research was made possible due to the financial support from the Royal Thai Navy, the Vision, Speech and Signal Processing group within the Electrical and Electronic Engineering department of University of Surrey, and the ORS Awards Committee UK. Their financial assistance is greatly appreciated.

Finally, I am indebted to many friends and colleagues, especially to Isaac Ng, Graeme Jones, Tele Tan and James Song, who provided me with moral and technical support. Their friendship made life at Surrey a pleasant and enjoyable experience.
# Contents

1 Introduction  1
   1.1 Problem with Dimensionality  ....................................... 1
   1.2 The Dimensionality Reduction Process  .................................. 4
      1.2.1 Role of Dimensionality Reduction  .................................. 4
      1.2.2 Approaches to Mathematical Dimensionality Reduction  .......... 5
      1.2.3 Mathematical Preliminaries  ...................................... 10
   1.3 Comparison of Feature Selection and Extraction from Practical Viewpoint 11
   1.4 Aims  .......................................................... 13
   1.5 Achievements  .................................................... 14
   1.6 Overview of the Thesis  ............................................. 15

2 Feature Selection Criteria  17
   2.1 Basic Concepts of Feature Selection Criteria  .......................... 18
      2.1.1 Direct Calculation of Error Probability  .............................. 22
      2.1.2 Probabilistic Distance Measures  ..................................... 23
      2.1.3 Probabilistic Dependence Measures  ................................... 25
      2.1.4 Entropy Measures  ............................................... 26
      2.1.5 Interclass Distance Measures  ..................................... 26
   2.2 Most Often Used Distance Measures  .................................... 27
      2.2.1 Interpretation of The Bhattacharyya Distance and The Divergence 28
      2.2.2 Parametric Separability Measures  ................................... 30
      2.2.3 Recursive Calculation of Parametric Separability Measures  ...... 32
      2.2.4 Comparison of Error Bounds  ...................................... 35
   2.3 Feature Evaluation Criteria for Multiclass  ............................. 37
   2.4 Finite Sample Size Effects  .......................................... 38
      2.4.1 A Sample Size Dependent Error Bound  ............................. 39
      2.4.2 Predicting the Minimum Training Size  ............................. 39
   2.5 Summary  ......................................................... 40
3 A Review of Feature Set Search Procedures

3.1 Preliminaries

3.1.1 Earlier Work

3.1.2 Recent Search Strategies

3.2 Optimal Search Procedures

3.2.1 Exhaustive Search

3.2.2 Branch-and-Bound Algorithm

3.3 Suboptimal Search Procedures

3.3.1 Feature Selection on the Individual Merit Basis

3.3.2 Sequential Forward Selection

3.3.3 Sequential Backward Selection

3.3.4 Generalized Sequential Forward and Generalized Sequential Backward Selection

3.3.5 Plus $l$-Minus $r$ Algorithm

3.3.6 Generalized "Plus $l$-Minus $r$" Algorithm

3.3.7 Sequential Forward Floating and Sequential Backward Floating Selection

3.3.8 Max-Min and Min-Min Search Methods

3.4 Experimental Results and Discussion

3.5 Conclusion

4 The Max-Min Algorithm

4.1 Introduction

4.2 The Max-Min Method

4.3 Heuristic Reasoning Behind the Max-Min Method and Its Drawbacks

4.4 Can The Performance of The Max-Min Algorithm Be Improved?

4.4.1 A Measure of Dependence

4.4.2 Amplifying Dependence

4.4.3 Modifying the Max-Min Procedure

4.4.4 Experimental Results

4.4.5 Conclusion

4.5 Proof of Invalidity of The Key Premise of The Max-Min Algorithm

4.6 Conclusion

5 Feature Selection in Unknown Distributions

5.1 Introduction
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2</td>
<td>Approximating the Unknown Class Conditional Distributions</td>
<td>107</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Why Approximation?</td>
<td>107</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Constructing Finite Mixtures of the Same Subspace</td>
<td>108</td>
</tr>
<tr>
<td>5.3</td>
<td>Parameter Optimization</td>
<td>112</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Approximation Criterion</td>
<td>112</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Maximum Likelihood Estimation</td>
<td>113</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Optimization Using The EM Algorithm</td>
<td>114</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Application to a Particular Type of Mixtures</td>
<td>118</td>
</tr>
<tr>
<td>5.4</td>
<td>Approach to Feature Selection</td>
<td>121</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Feature Selection Criterion</td>
<td>121</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Characteristics of the Proposed Feature Selection Method</td>
<td>123</td>
</tr>
<tr>
<td>5.5</td>
<td>Experiments</td>
<td>126</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Simulated Data</td>
<td>127</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Texture Data</td>
<td>128</td>
</tr>
<tr>
<td>5.5.3</td>
<td>Speech Data</td>
<td>129</td>
</tr>
<tr>
<td>5.6</td>
<td>Conclusions</td>
<td>131</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions and Future Work</td>
<td>135</td>
</tr>
<tr>
<td>6.1</td>
<td>Conclusions</td>
<td>135</td>
</tr>
<tr>
<td>6.2</td>
<td>Future Work</td>
<td>137</td>
</tr>
</tbody>
</table>

Appendix A Test Images
# List of Figures

1.1 *The formation of an observation vector of a printed character ‘c’.* .......................... 2
1.2 *A simplified block diagram of a pattern recognition process.* ................................. 4
1.3 *Dimensionality reduction by feature selection.* .................................................... 8
1.4 *An example of feature selection.* ........................................................................... 8
1.5 *Dimensionality reduction by feature extraction.* ..................................................... 9
1.6 *An example of feature extraction.* ........................................................................ 9
2.1 *Plots of probability density functions of various features.* .................................... 19
2.2 *Plots of correlation ellipses of various pairs of features.* ....................................... 21
2.3 *Plots of Probability density functions* ................................................................. 24
2.4 *1-Dimensional example depicting the difference between the $i^{th}$ class conditional p.d.f. and the mixture p.d.f.* .......................................................... 25
2.5 *The interpretation of the Bhattacharyya distance.* ............................................. 29
2.6 *The interpretation of the divergence.* ................................................................. 29
3.1 *Solution tree for a branch-and-bound algorithm.* ............................................... 54
3.2 *Results of the optimal search methods* ............................................................... 71
3.3 *Results of the sequential forward search methods* ............................................. 72
3.4 *Results of the sequential backward search methods* ........................................... 73
3.5 *Results of the Plus l-Minus r search method when $l > r$* ....................................... 74
3.6 *Results of the Plus l-Minus r search method when $l < r$* ....................................... 75
3.7 *Results of the generalized Plus l-Minus r search method when $l > r$* ............... 76
3.8 *Results of the generalized Plus l-Minus r search method when $l < r$* ............... 77
3.9 *Results of the floating methods* ....................................................................... 78
4.1 *An example of possible ranges and actual values of Iy’s.* ...................................... 86
4.2 *Two examples of unconditional dependence cases* ............................................. 89
4.3 *An example of a class conditional stochastic dependence case.* ............................. 91
4.4 *Two contrasting examples of class conditional stochastic dependence cases.* ....... 92
4.5 1-Dimensional projection of Case 1 in Figure 4.4 on to $x_1$ and $x_2$ axes. 93
4.6 1-Dimensional projection of Case 2 in Figure 4.4 on to $x_1$ and $x_2$ axes. 94
4.7 Results of the modified Max-Min methods 96

5.1 Data sets having identical first- and second-order statistics. 106
5.2 Approximation of an unknown distribution by three mixture components. 111
5.3 A counter-example for the Karhunen-Loeve expansion. 124
5.4 A schematic illustration corresponding to equiprobability contours of 2-D distributions in which our approach fails. 125
5.5 A schematic illustration corresponding to equiprobability contours of 2-D distributions in which feature selection using distance measure approach fails. 126

A.1 The two colour images used in the experiment. 139
### List of Tables

5.1 *Parameters used in simulating the data.* ................................. 127
5.2 *Classifier performance on different feature subset size as a function of the size of the training set.* .................................................. 128
5.3 *Classifier performance on different feature subset size of image data.* ........ 129
5.4 *Classifier performance on different feature subset size of speech data.* ........ 130
Chapter 1
Introduction

Statistical pattern recognition is basically concerned with the problem of automatic classification of elements belonging to one of \( m \) possible classes, \( \omega_i, i = 1, 2, \ldots, m \). The decision regarding the class membership of an element is made on the basis of a set of \( D \) measurements taken on the element forming a pattern which, presumably, contains enough information to distinguish among the classes. That is, the statistical variability (which is not due to a random noise) between patterns from different classes is high enough to ensure satisfactory performance in terms of a correct classification rate.

The field of pattern recognition has evolved as a setting for studying general statistical-based classification methods. In recent years, considerable advances have been achieved in the practical application of statistical classification techniques, thanks to the rapid development in the digital computers. These techniques have now penetrated diverse areas of science such as character recognition, medical diagnosis, remote sensing, speech recognition, image classification and radar or sonar signature analysis, etc. A detailed discussion of various approaches to pattern classification techniques can be found in, e.g. [4], [5], [7], [10] and [13].

1.1 Problem with Dimensionality

For any given classification problem there is often an unlimited number of primitive measurements which could be made on the object of interest. For example, consider the problem of optical character recognition where the objective is to identify the membership of specimens of a given alphabet; the primitive measurements might consist of a digitized character as shown in Figure 1.1. In this case, suppose that the grid has \( D \) cells and the electrical output of each cell increases as more of the character appears in the cell. Then a \( D \)-dimensional vector \( Y \) comprised of the cell outputs is formed; the \( i^{th} \)
Chapter 1: Introduction

Figure 1.1: The formation of an observation vector of a printed character 'c'.
(a) Digitized Character. (b) Character as data vector. (c) Character as point in observation space.

The component of $Y$ is proportional to the fraction of the $i^{th}$ cell covered by the character. The vector $Y$ is called the observation vector and the $D$-dimensional space in which it resides is called the observation space or measurement space. A cost is associated with each measurement. This cost may involve processing time, memory requirements and ultimately classification success. Since each of these measurements carries a very small amount of information about the object, the number of measurements $D$ usually becomes high, perhaps hundreds. Moreover, it is extremely difficult to make sure that the measurements are independent. Therefore, many of these measurements may be redundant or irrelevant.
Problem with Dimensionality

It is known that the performance of a pattern recognition system is closely related to the measurements taken by the classifier. As the number of the measurements $D$ increases, the number of parameters defining the decision surface increases, and a more flexible decision surface classifies the training set with a lower classification rate. However, as $D$ increases the estimate of the probability associated with each class becomes less and less reliable (a consequence of the curse of dimensionality [1]). This means that the training set, which is of limited size in practice, becomes more and more sparsely distributed, and the training set elements become less and less representative of the shape of the class conditional density functions. The consequence is that the decision surface may fit the training set with increasing $D$, but that this decision surface generalizes less well to new samples from the test set, i.e. the true error rate increases. Therefore, a lower misclassification rate can sometimes be achieved by using fewer measurements. Obviously, as the number of inputs to a classifier becomes smaller, the design of the classifier will become simpler.

There are, of course, ad hoc procedures for reducing the number of measurements based upon specific problem knowledge and empirical studies. However, even if good results of research in one area of pattern recognition are obtained, the same principle could not possibly give similar results in other areas. Therefore, more formal statistical approaches must be adopted. This problem is known as dimensionality reduction and is probably the most important subject of pattern recognition. This is because with a proper and efficient dimensionality reduction process, both simple and sophisticated classification algorithms can be implemented owing to the large dimensionality reduction provided by such a process. On the other hand, even the highly sophisticated classification algorithms will not be able to compensate for any information loss incurred by an improper or inefficient dimensionality reduction process. As a result, the classification algorithms will necessarily be less efficient and classification errors will increase. In addition, there is another equally important function of the dimensionality reduction stage apart from the physical reduction of dimensionality as required by feasibility limitations of either a technical or economical nature. That is to ensure the reliability of the decision-making processor by removing the redundant and irrelevant information which has a derogatory effect on the classification process.
Chapter 1: Introduction

1.2 The Dimensionality Reduction Process

A pattern recognition problem begins with class definitions and labeled samples of those classes in some workable representation. The problem is solved when a decision rule is derived which assigns a unique label to a new pattern. Therefore, the process necessary in deriving the decision rule in a pattern recognition problem can be indicated diagrammatically in Figure 1.2. Both the dimensionality reduction process which has the function of reducing the dimensionality of the representation vector, and the classifier, which carries out the actual decision process, work with a vector of measurements which can be considered as an abstract pattern. As a result, the dimensionality reduction and classification stages can be implemented using mathematical methods irrespective of the original application. However, this orientation does not imply that the choice of methods is independent of the application, but that the methods themselves can be discussed as independent tools. Note that the ideal classification stage should be applied in a feature space which (a) is finite-dimensional (b) is of relatively low dimension, and (c) contains sufficient information to satisfactorily perform the classification.

Figure 1.2: A simplified block diagram of a pattern recognition process.

The inclusion of the dimensionality reduction stage in Figure 1.2 effectively partitions the pattern recognition problem into two subproblems: dimensionality reduction process and classifier design. Coincidently, this model of the machine pattern recognition appears to be consistent with the mechanisms involved in human perceptual processes. Ideally, the problem of feature selection and extraction should never be considered independently from the classifier design. In practice, however, the simplifying assumption that they are mutually independent is often assumed.

1.2.1 Role of Dimensionality Reduction

The primary purpose of reducing the dimensionality of the original measurements stems from engineering considerations. First of all, by minimizing the number of pattern descriptors the cost of hardware implementation of the data acquisition system (sensors,
The Dimensionality Reduction Process

A/D converters, data processing system) can be substantially reduced. Furthermore, the complexity of hardware implementation of a classifier grows rapidly with the number of dimensions of the pattern space. It is therefore important to base decisions only on the most essential, so-called discriminatory information, which is conveyed by features. This, in turn, will effectively minimize the storage capacity, cost and time needed for a pattern recognition system.

The need of dimensionality reduction may also be motivated by other factors, such as the compression of information to facilitate the transmission of pattern vectors over a long distance communication lines. In such a case the identification of redundant or irrelevant measurements, with regard to a particular classification tasks, may be important for reducing the large communication bandwidth required and the overall cost of measurement extraction.

Apart from such engineering constraints, dimensionality reduction may prove beneficial from the classification performance point of view. This claim may seem somewhat controversial, since from the standpoint of Bayesian decision rules there are no bad features. One cannot improve the performance of a Bayes classifier by eliminating a feature (this property is called monotonicity). However, the use of Bayesian classifiers in practice is limited by the fact that, in general, a priori densities are not known and must be estimated from a set of finite (usually small) number of training samples. If the number of training samples is small, problems are commonly manifested due to the so-called peaking phenomenon [3], [14], [9]. This phenomenon concerns the dependence of the probability of correct recognition of patterns outside the training set and the number of features used. Initially the performance improves as new features are added, but at some point inclusion of further features may result in an actual degradation in performance. As a consequence, it is possible to improve the accuracy of the classifier’s performance by deleting a feature [6], [12].

Finally, a useful by-product of feature evaluation is that it provides the means for assessing the potential of a given pattern representation-space for discriminating between elements of different classes. If the class overlap is too high, even in representation space, then new sources of information will need to be sought to enhance class separability.

1.2.2 Approaches to Mathematical Dimensionality Reduction

In the context of pattern recognition, the underlying philosophy of any dimensionality reduction mechanism should be the retention of class discriminatory and the reduction of class commonality information. Lower-dimensional pattern descriptors are commonly
referred to as features. Features play a fundamental role in characterizing the distinguishing properties of pattern classes. Thus, discovering the properties of patterns of one class which discriminate those from another class should maintain recognition capability but in a reduced dimensionality feature space. Dimensionality reduction is usually presented in the context of supervised pattern recognition because a priori knowledge of prototype classification is often used for defining intra versus inter class considerations as well as for comparing the dimensionality reduction results. However, for non-supervised tasks the similar principles can just as well be employed (for example see Chapter 9 of [4]).

Before proceeding further, it should be mentioned that dimensionality reduction techniques can be categorized from two different standpoints –

- according to their purpose (for representation or discrimination).
- according to the way of deriving features (selecting or extracting).

The first category is actually related to a number of classes which are considered. In this case, we consider all the data as one group rather than a mixture of different classes. Then, it is the problem of finding optimal features for one distribution. This problem is often referred to as the problem of optimal representation of the original data vector (or a signal) in a lower-dimensional subspace. The primary goal in this case is not to select features which provide the best discrimination among the classes, but rather to select features according to quite a different criterion, like e.g. minimization of the mean-square error between the original data and their representation in the lower-dimensional subspace (Karhunen-Loeve expansion in discrete form which is sometimes known as principal component analysis).

On the contrary, the methods having the purpose of finding features which carry most of discriminative information obviously concern of a multiclass problem. The data are no longer considered as being just one group and the problem is referred to as the problem of finding optimal features for multidistribution.

In certain cases of high-dimensional signals, the problem of their optimal representation in a feature subspace is of great importance. This is because a direct application of the methods of selecting directly the most discriminative features may not be computationally feasible. In these cases, finding features for one distribution can serve as the first stage of the overall feature selection or extraction process.

In our research we have restricted ourselves to the problem of finding features which discriminate well among different classes, thus the problem of optimal representation is not discussed. As far as the dimensionality reduction techniques in the second category
are concerned, all techniques can be classified into two distinct approaches. The first approach is to identify measurements which give insignificant contribution or none at all to class separability, as illustrated in Figure 1.3. The problem is then one of selecting a small subset of \( d \) features \( x_j, j = 1, 2, \ldots, d \), out of the available \( D \) measurements \( y_k, k = 1, 2, \ldots, D \). This dimensionality reduction process is known as *feature selection*. Note that no computation is required during routine pattern processing. The redundant and irrelevant sensor outputs are simply ignored.

An example of 2-dimensional feature selection is illustrated in Figure 1.4. The distribution of each pattern class in Figure 1.4(a) is a tilted ellipsoidal which indicates that the two measurements are highly correlated. If the measurement \( y_1 \) is individually considered as in Figure 1.4(b), the class conditional marginal density functions are heavily overlapped. On the other hand, as shown in Figure 1.4(c), if the measurement \( y_2 \) is individually considered, the marginal density functions are well separated with little overlap. Therefore, as far as the contribution to the class separability is concerned, the measurement \( y_2 \) is obviously much better than \( y_1 \), and the measurement \( y_1 \) may be ignored.

In contrast, the second approach utilizes all the sensor outputs and maps the useful information content in the measurement space into a lower-dimensional feature space, as shown in Figure 1.5. This method is referred to as *feature extraction*. The problem can be viewed as projecting the original \( D \)-dimensional measurement space on a \( d \)-dimensional subspace, and finding the orientation of the subspace which best preserves the information available in the complete subset. Though, a linear mapping is usually considered, it can also be non-linear. Figure 1.6 shows an example of 2-dimensional feature extraction. Figure 1.6(a) exhibits two principle axes, \( e_1 \) and \( e_2 \), of the density functions that the measurements can be projected on to. If the two distributions are mapped on to \( e_1 \), the marginal density functions are heavily overlapped as in Figure 1.6(b). On the other hand, if they are mapped to \( e_2 \), the marginal density functions are well separated with little overlap as illustrated in Figure 1.6(c). It is clear that, for classification purposes, the projection of measurements on to \( e_2 \) is preferred to \( e_1 \) since it preserves more of the discriminatory information. Therefore, feature extraction can be regarded as feature selection in transformed space.

It is worth of mentioning that in some books on pattern recognition a little discrimination is made between the two approaches. It can be regarded as their drawback since both the approaches differ quite considerably in their practical implications. The point which will be discussed later.
Figure 1.3: Dimensionality reduction by feature selection.

Figure 1.4: An example of feature selection. (a) Correlation ellipses (density contours) of both measurements $y_1$ and $y_2$. (b) Class conditional probability density of measurement $y_1$ alone. (c) Class conditional probability density of measurement $y_2$ alone.
Figure 1.5: Dimensionality reduction by feature extraction.

Figure 1.6: An example of feature extraction. (a) Correlation ellipses (density contours) of both measurements $y_1$ and $y_2$. (b) Projection of $y_1$ and $y_2$ on to $e_1$ axis. (c) Projection of $y_1$ and $y_2$ on to $e_2$ axis.
Chapter 1: Introduction

1.2.3 Mathematical Preliminaries

It must be emphasized that in this thesis the subject of feature selection and extraction in pattern recognition is concerned with mathematical tools for reducing the dimensionality of pattern representation, not with the definition of distinguishing characteristics scenes. In order to solve the feature selection problem, we need to specify three main ingredients: the feature evaluation criterion, the dimensionality of the feature space, and the optimization procedure. In addition, in feature extraction we need to specify the form of the mapping $A$.

Before we proceed any further, it will be very useful to introduce some notation. We shall denote a set of $D$ possible measurements providing adequate representation of any element to be classified by $\{y_k\}, k = 1, 2, \cdots, D$, or in a vector form by $Y$, that is,

$$Y = [y_1, \cdots, y_D]^T \quad (1.1)$$

Each representation vector $Y$ belongs to one of $m$ possible pattern classes $\omega_i, i = 1, 2, \cdots, m$. It will be assumed that patterns are generated by a random process and that the model of the process can be characterized by class conditional density functions $p(Y | \omega_i)$ and a priori class probabilities $P(\omega_i), i = 1, 2, \cdots, m$.

A set of candidate features $x_j, j = 1, 2, \cdots, d$ where $d < D$, will be denoted by $\chi$, while the set of optimal features in the sense of maximum or minimum of some criterion functions $J$ will be designated by $X = \{x_j | j = 1, 2, \cdots, d\}$. It will be seen in the following chapter that the criterion functions are defined in terms of the model characteristics $P(\omega_i)$ and $p(Y | \omega_i)$ or the conditional density function marginals $p(x | \omega_i)$. In the case of feature selection, optimization is carried out over all possible candidate feature sets, that is,

$$J(X) = \max_Z J(Z) \quad (1.2)$$

In feature extraction, on the other hand, the optimization is performed over all admissible mappings, that is,

$$J(A) = \max_A J(A(Y)) \quad (1.3)$$

where $A$ is an optimal feature extractor which can be either linear or nonlinear [2]. Once $A$ is determined, the feature vector

$$x = [x_1, \cdots, x_d]^T \quad (1.4)$$
Comparison of Feature Selection and Extraction from Practical Viewpoint

is then given by

\[ X = A(Y) \]  \hspace{1cm} (1.5)

Note that the class of linear mappings includes the feature selection since the selection of any \(d\) features out of \(D\) measurements can be similarly accomplished by selecting the appropriate \(d \times D\) matrix \(A\) consisting only of 0's and 1's, one 1 per row.

1.3 Comparison of Feature Selection and Extraction from Practical Viewpoint

The factors governing the choice of dimensionality of the feature space are common to both feature selection and extraction approaches. They include hardware or computational constraints, the peaking phenomenon, or permissible information loss. When deciding between them, one has to be aware of their respective priorities and drawbacks so as to be able to choose the right approach from the viewpoint of ultimate goals and requirements, concerning the task to be solved. In the following, the priorities and drawbacks of both the approaches from a practical point of view are briefly discussed.

Feature Selection

• Priorities:
  
  (a). Selected features do not lose their original physical interpretation. They have an exactly defined meaning which is in many application fields an essential fact, e.g. in medical diagnostics.

  (b). Measurements which are not selected as significant features need not be measured or collected at all during the application phase. This fact may result in a considerable saving of time and costs in the phase of data collection where the ratio of unavoidable tedious human work is very often a restrictive "bottleneck" factor.

• Drawbacks:

  (a). Preservation of the physical interpret-ability is unfortunately paid by impossibility to achieve generally the optimum in the framework of selected criterion of feature significance as compared to general feature extraction (optimally extracted subset of \(d\) transformed features will have generally a better discriminative ability than the best subset of \(d\) original data components).
Chapter 1: Introduction

Feature Extraction

• Priorities:

(a). The feature space $\mathcal{X}$ in some applications is physically more meaningful than the measurement space $\mathcal{Y}$. For example it is well known that in speech analysis, the frequency spectrum is more meaningful than the actual speech waveform.

(b). Discriminative power achievable with optimally transformed features is generally higher than in the case of restricting ourselves only to selection without any transformation.

• Drawbacks:

(a). Since new features are the functions of all the original data components (e.g. usually their linear combination), they lack a straightforward and clear physical meaning. They may be looked upon as a certain abstraction and cannot be practically reasonably interpreted.

(b). The form or criterion of the mapping is usually restricted, consequently an oversimplistic model of the pattern recognition system is then implicitly assumed. This may cause the degradation of the system’s performance. For example, the classes are not linearly separable but the mapping is restricted to a linear form. This may result in a great loss of discriminatory information.

(c). None of $D$ original data components can be saved in the application phase. Quite contrary, all of them must be measured and collected since the resultant $d$-dimensional feature vector is derived only from the complete original $D$-dimensional data vector by applying a suitable transformation. From the standpoint of hardware, nothing is gained, and perhaps, even, something is lost in terms of performance.

From the outlined priorities and drawbacks of both the approaches, it can be seen that their properties are somewhat to a certain extent rather contradictory. Feature selection methods will be more suitable in cases when the potential user puts emphasis on preserving the interpret-ability of original data and prefers decision-making on the basis of meaningful features. Furthermore, they will be suitable when one of the goals is to reduce tediousness and costs of data collecting by finding the data components which can be completely excluded from further collecting process. Most problems of medical differential diagnostics, for example, belong to this class of tasks.
Aims

On the other hand there exist applications where the requirements are quite opposite. The emphasis is laid on optimum reduction of the problem dimensionality and neither any transformation of features nor the necessity of measuring all the original data components represent any problem at all. A typical example of such a diagnostic task is the VCG (vectorcardiogram) classification where the primary data vector is represented by a time series (sampled VCG signal), having several hundred elements. Since the physical interpretability of respective sampled points of the VCG curve is discussable anyhow and, moreover, since the signal is recorded automatically and thus the possibility to reduce the number of measured components plays obviously a less important role, feature extraction methods are preferable in this case.

1.4 Aims

As mentioned earlier, statistical pattern recognition techniques have penetrated diverse areas of science due to the rapid development in the digital computers. Recently, in some areas the problem of very high dimensionality data has been encountered. A good example regarding this problem is from texture classification where the original measurements are usually derived from a bank of convolution masks [11] or from the first- and second-order statistics [8]. Apart from resulting in a large number of measurements, it is also a time consuming process. High dimensionality makes pattern recognition problems difficult. Hence there is a need to reduce the dimensionality of original measurements. From the practical point of view, feature selection approach is chosen as the tool for dimensionality reduction process. This is because in feature selection the redundant and irrelevant measurements are simply ignored. As a consequence, these measurements need not be measured for the classification stage. On the contrary, feature extraction achieves no dimensionality reduction in the space of the original measurements. That is, all the measurements must be generated in order to map into a lower dimensional feature space. From the standpoint of hardware, nothing is gained, and perhaps, even, something is lost in terms of performance.

In the past, various methods have been developed for selecting subset of features. Some methods give very good results but at the expense of large computational time therefore they are unlikely to be suitable for high dimensionality problem. On the other hand, there are some methods which produce the results not as good as the former methods but they require less computational time. Motivated by these problems, the primary aim is, therefore, to develop a fast algorithm which can select a feature subset of size \( d \) from \( D \) available measurements when \( D \) is very large without significantly degrading
the performance of the recognition system.

Another important issue in feature selection is that when evaluating the feature set effectiveness, one usually uses various analytical criterion functions that bound the probability of error instead of the actual error rate from the classifier which is a time consuming process. However, there is a danger in this procedure. The criterion functions usually depend upon the probability density function, which may not be known. It is tempting, therefore, to make specific assumptions about the form of the density (e.g. that it is multivariate Gaussian). This can give misleading results. Therefore, a development of an algorithm that can select a good feature set when the form of underlying probability distribution of the pattern is unknown is proposed.

1.5 Achievements

In this thesis only the feature selection problem is studied in detail. It is not at all concerned with the problem of extracting measurements from the object of interest. The main contributions of the work reported in the thesis can be summarized as follows:

- A large number of feature subset search strategies have been reviewed, analyzed and compared. Following the analysis of their respective advantages and shortcomings, the conditions under which certain strategies are more pertinent than others have been established. As a result some of the strategies have been recommended as suitable methods for solving large-scale feature selection problems.

- The reasons for a rather poor performance of one interesting approach known as the Max-Min method have been discovered. The main one is the lack of theoretical justification for the original algorithm. A necessary condition for a counter-example to the key assumption of the Max-Min to hold has been found. The results, though somewhat negative, are of a great importance to other potential users of the Max-Min method, since unrealistic expectations might otherwise be associated with it.

- A new feature selection algorithm based on approximating the unknown class conditional densities by finite mixtures of the densities of a special type has been developed and successfully employed. The approximation is best in the sense of minimizing the Kullback-Leibler distance between the true and postulated class conditional densities. It is especially suitable for multimodal distributions.
1.6 Overview of the Thesis

The problem of feature selection investigated in this thesis is that of selecting the best subset of $d$ features from the given set of $D$ measurements in accordance with a given feature selection criterion. In the following chapters the fundamental problems in feature selection which confront designers of pattern recognition systems are studied in detail.

Chapter 2 begins with an introduction of basic concepts of feature selection. This is followed by a summary of the most widely used criteria and their error bounds. The parametric forms and recursive forms of these criteria are also presented. Furthermore, the effect of limited training size on the overall performance in relation to these criteria is discussed and a general guideline relating to the number of training samples and the number of features is given.

The major developments over the past 30 years in the area of feature selection, as far as search strategies are concerned, are reviewed in Chapter 3. The advantages and disadvantages of various methods are compared both theoretically and experimentally. Owing to the really appealing characteristics in terms of speed of computation, the Max-Min method is singled out for a further study. A detailed analysis of the Max-Min method and various attempts at improving its performance are discussed in Chapter 4. It will be shown, however, that conceptual problems with the Max-Min approach prevent its improvement to make it a sufficiently reliable tool for feature selection.

In Chapter 5, a new feature selection algorithm based on approximating unknown class conditional distributions by a finite mixture of the densities of a special type is presented. Experiments performed on simulated and real problems are described and their results discussed.

A short summary concludes each chapter, but a more detailed discussion is presented in Chapter 6. Finally, some unsolved problems and ideas for future research are presented in Chapter 6.

References


Chapter 2

Feature Selection Criteria

Feature selection is an important task in the design of automatic pattern classifiers since the success of a classifier depends both on the kind of classifier used (linear, quadratic, piecewise linear, etc.) and on the features the classifier operates on. The main goal of feature selection is to select a subset of \( d \) features from the given set of \( D \) measurements, \( d < D \), without significantly degrading the performance of the recognition system [16], [20],[26]. The achievement of this goal requires

- a capability for evaluating the effectiveness of feature subsets.
- an effective strategy for searching for the best \( d \) features from the given \( D \) measurements.

The procedures for searching for the best subset, in general, may be discussed in the context of e.g. sequential and nonsequential pattern recognition. The common need, however, for all these procedures is an evaluation function by which the effectiveness of a feature, or a subset of features, in distinguishing among the classes is assessed.

A traditional method for selecting the best feature subset, in the case when a large training set is available, is to estimate the classification error rate by splitting the training set into two subsets; one subset is used for constructing the classifier, and the other subset is used for testing the classifier, thus measuring the error rate. Repeating this process using various feature subsets, the best feature subset is chosen as the one that yields the lowest error rate. This approach is known as the hold-out method. The main disadvantage of this method is that it does not use the data efficiently, and it takes a large training set (let alone the heavy computational burden) to achieve good results. Other approaches to use the given training set as the design set and as the test set, such as the resubstitution, the cross-validation and bootstrap methods, have their main disadvantage of either computational time consuming or producing biased results.
Chapter 2: Feature Selection Criteria

It follows that a lot of computational time is required to evaluate the effectiveness of each subset when the actual error rate, which is known to be biased anyway [14], [40], [44], [49], is used as the feature selection criterion. However, it is generally accepted that the computational time can be substantially reduced especially in the parametric case, if alternative feature selection criteria are employed instead. If the computational resources dedicated to feature set effectiveness evaluation are reduced, more time can be spent on the search for the best subset. Nevertheless, these criteria are useful only if they can be related to the error probability. Another important advantage of using alternative feature selection criteria is due to the fact that the optimum feature set depends on the classifier, therefore it is difficult to compare the effectiveness of different feature sets in general. In order to avoid this additional complexity, it is strongly suggested to seek the optimum feature set with reference to the Bayes classifier [18]. This is because the error rate of the Bayes Classifier is equivalent to the probability of error in which most of the alternative feature selection criteria can be related to.

2.1 Basic Concepts of Feature Selection Criteria

The idea underlying feature selection is to obtain features which maximize the similarity of objects in the same class while maximizing the dissimilarity of objects in different classes. It is desirable to perform feature selection on statistically independent (uncorrelated) features [34]. However, it should be noted that uncorrelated features do not necessarily mean that the features are statistically independent. Unfortunately, features are seldom independent in practice and hence the statistical dependencies among features have to be taken into account when selecting features. If the dimensionality of the feature set is low, it is easy intuitively to select features that give high discriminatory power. As an example, consider the case when we want to select the best feature from four available features. Figure 2.1 shows the plots of individual class conditional density functions of all four features. Now, the amount of discriminatory power (information) can be related to the amount of non-overlapping area of the class conditional density functions [28], [37]. Obviously, according to the discriminatory information content of individual features, they would be ordered as follows: $x_2, x_3, x_1$ and $x_4$ respectively. Therefore, feature $x_2$ is chosen as the best feature since it has the smallest non-overlapping area, i.e. the highest class discriminatory power.
Figure 2.1: Plots of probability density functions of various features.
However, if we want to select the best two features we cannot simply choose the two best features. This is because the statistical dependence among the features can cause the best two-feature subset not to be composed of the individually best features. Surprisingly, even conditionally independent features can exhibit such an anomalous behaviour [11], [47]. These relationships are demonstrated in Figure 2.2. It depicts 2-dimensional density contours in which the major and minor axes are equal to the standard deviations (correlation ellipses) of various pairs of features and the amount of overlap is inversely proportional to the discriminatory power. Figure 2.2(a) illustrates the combined contribution to the class separability when features $x_1$ and $x_2$ are taken together. Their combined discriminatory power is slightly higher than that of features $x_2$ and $x_4$ as shown in Figure 2.2(b). However, as illustrated in Figure 2.2(c) when the two best features $x_2$ and $x_3$ are combined together they disappointingly do not give the highest discriminatory information. In contrast, the two features $x_1$ and $x_4$ which, if taken separately, are useless for discriminating between the two classes, give a perfect separation of the two classes when taken together, as shown in Figure 2.2(d). This effect is compounded in the case of multivariate relationships. However, as the dimensionality of the feature set increases, the study of the effect of statistical dependencies among features is not quite straightforward and is beyond human visualization abilities. Therefore, some criterion functions are needed to evaluate the discriminatory information of various feature sets. There exist many feature evaluation criteria some of which will be discussed in the following subsections.
Figure 2.2: Plots of correlation ellipses of various pairs of features.
2.1.1 Direct Calculation of Error Probability

The ultimate goal for designing a classification system is to achieve the lowest possible probability of error. Since the probability of error is the design criterion for the whole recognition system, it is naturally also the ideal objective function for designing the feature selection subsystem.

In an m-class case, the Bayes error probability, $P_e$, for a d-dimensional candidate feature set $\mathbf{x} = \{\xi_1, \xi_2, \cdots, \xi_d\}$ is given as

$$P_e(\mathbf{x}) = \int_{-\infty}^{\infty} \left[ 1 - \max_{\forall \xi \in \mathbb{R}^d} P(\omega_k | \xi) \right] p(\xi) d\xi$$

(2.1)

where $\xi$ is the feature vector composed of candidate features $\xi_j$, that is,

$$\xi = [\xi_1, \xi_2, \cdots, \xi_d]^T.$$  

(2.2)

$P(\omega_i | \xi)$ is the a posteriori probability of the $i^{th}$ class and $p(\xi)$ denotes the mixture density function satisfying

$$p(\xi) = \sum_{i=1}^{m} p(\xi | \omega_i)P(\omega_i).$$

(2.3)

The a posteriori probability function $P(\omega_i | \xi)$ is related to the $i^{th}$ class conditional density as

$$P(\omega_i | \xi) = \frac{p(\xi | \omega_i)P(\omega_i)}{p(\xi)}.$$  

(2.4)

A direct calculation of the probability of error, in general, is often impossible or impractical partially due to the following reasons:

- If the probability density functions $p(\xi | \omega_i)$ of the classes are assumed perfectly known, then the expression for the probability of error of any classification rule can be computed. However, its actual evaluation involves usually numerical integration in a high-dimensional space.

- If the density functions are not completely known, an additional disadvantage is that a small inaccuracy in the density function may induce a substantial error in the expression for the probability of error.
Therefore, it is useful to search for alternative feature evaluation criteria that may be weaker than the error probability, but easier to evaluate and manipulate. Upon examining the work in the area of feature selection, the extent to which other feature evaluation criteria are used so that the probability of error $P_e$ can be avoided is quite surprising. It is apparent that an ideal alternative criterion is one which does not contradict the $P_e$ rule. That is, if $\chi'$ is not preferred to $\chi''$ by the $P_e$ rule, then it is not preferred to $\chi''$ by that alternative criterion either.

Ben-Bassat [4] has introduced the concept of equivalence groups, i.e. according to which all the potential features are partitioned into the groups, in which the features exhibit the same $P_e$, when considered individually on their own. Thus the features within the same group are indifferent from the point of view of $P_e$ criterion. However, among the features which are individually indifferent by the $P_e$ rule, it is possible, and moreover desirable, to have an internal ordering which differentiates between good and bad features. The reason is the insensitivity of $P_e$ criterion under certain circumstances, caused by the fact that the $P_e$ function depends only on the $a$ posteriori most probable class. In this connection Ben-Bassat states that an ideal rule, which would serve as an alternative to $P_e$ criterion, should result an in equivalence groups partition either coinciding with that induced by $P_e$ rule, or preferably even refining it. Furthermore, he questioned the widely spread hope of finding a "magic functional" which, for arbitrary class distributions, induces the same ordering as does the $P_e$ rule. His results [5] lead to a conjecture that such a magic functional probably does not exist at all for the general case.

Since ideal rules could not be found, it was suggested to assess a feature evaluation criterion by considering the tightness of both their lower and upper bounds on the probability of error by means of the evaluation function. The major research work in the past was, therefore, directed towards utilizing various information measures for the evaluation of features, and the determination of their bounds on the error probability of the Bayes classifier [12], [21], [25]. In the following, a brief introduction to various concepts of class separability measures on which alternative feature selection criteria are based is presented. For a complete list of distance measures see Devijver and Kittler [13].

### 2.1.2 Probabilistic Distance Measures

The concept of probabilistic distance can be developed by considering the probability of error $P_e$ in the two-class case. From equation (2.1), it is easy to show that for $m = 2$,

\[
P_e(\chi) = 0.5 \left[ 1 - \int_{-\infty}^{\infty} |p(\xi | \omega_1)P(\omega_1) - p(\xi | \omega_2)P(\omega_2)| d\xi \right].
\]  

(2.5)
Figure 2.3: Probability density functions of (a) two completely overlapping classes and (b) two well separated classes.

From equation (2.5), it can be easily seen that $P_e$ will be maximum when the integrand is zero, that is, when density functions $p(\xi | \omega_i), i = 1, 2$ weighted by the respective class a priori probabilities are completely overlapping as illustrated in Figure 2.3(a). On the other hand, $P_e$ will be zero when the functions $p(\xi | \omega_i)$ do not overlap as in Figure 2.3(b). Furthermore, the integral in equation (2.5) expresses a "probabilistic distance" between two weighted density functions which is referred to as the Kolmogorov variational distance. The greater the distance, the smaller the error and vice versa. Thus by maximizing the Kolmogorov distance for all the possible feature sets it is possible to find the optimal feature space which is associated with the minimum error.

Unfortunately, using the Kolmogorov distance is not a very convenient way to measure the class separability due to the complexity in evaluation. However, by analogy to the Kolmogorov distance it is possible to introduce other distance measures between two density functions which are related to error probability and can therefore serve as criterion for feature selection. In general any measure that is,
Basic Concepts of Feature Selection Criteria

\[ J(z) = \int_{-\infty}^{\infty} f[p(\xi | \omega_i), P(\omega_i), i = 1, 2] d\xi \]  \hspace{1cm} (2.6)

satisfying

- \( J(\chi) \geq 0 \) and \( J(\chi) = 0 \) when \( p(\xi | \omega_i), i = 1, 2 \) are completely overlapping, i.e. identical.

- \( J(\chi) \) is maximum when \( p(\xi | \omega_i), i = 1, 2 \) are nonoverlapping, i.e. disjoint.

can be used as a probabilistic distance measure of class separability. Therefore, features selected according to the magnitude of \( J(\chi) \) will imply their corresponding discriminatory power between the two pattern classes. In other words, feature set \( \chi' \) is considered more effective than the feature set \( \chi'' \) if \( J(\chi') > J(\chi'') \).

### 2.1.3 Probabilistic Dependence Measures

The pattern recognition process can be considered to involve two random variables: the pattern vector \( \xi \) and the class \( \omega \). The observation of an outcome of the former enables us to make a decision about the latter. The dependence of the two variables is embodied in the class conditional density functions \( p(\xi | \omega_i), i = 1, 2, \ldots, m \). If \( \xi \) and, say, \( \omega_i \) are statistically independent, then \( p(\xi | \omega_i) = p(\xi) \), that is the \( i^{th} \) class conditional density function will be identical to the mixture density. In such a situation, by observing the pattern vector \( \xi \) we do not gain any information about its class membership. On the other hand, if \( \xi \) is statistically dependent on \( \omega_i \), the \( i^{th} \) class conditional density function \( p(\xi | \omega_i) \) will be different from the mixture distribution as shown in Figure 2.4.

![Figure 2.4](image)

**Figure 2.4:** 1-Dimensional example depicting the difference between the \( i^{th} \) class conditional p.d.f. and the mixture p.d.f.
Chapter 2: Feature Selection Criteria

It is apparent that the degree of dependence between the variable $\xi$ and a particular realization of $\omega_i$ can be measured by the "distance" between the class conditional density $p(\xi \mid \omega_i)$ and the mixture density $p(\xi)$. It is noticed that this distance constitutes a natural concept of class separability since the greater the distance, the greater the dependence between $\xi$ and $\omega_i$. In general, any criterion function

$$J(\chi) = \int_{-\infty}^{\infty} g[p(\xi), p(\xi \mid \omega_i), P(\omega_i), i = 1, 2]d\xi$$

(2.7)

satisfying the conditions of equation (2.6) can be used as a probability dependence measure.

2.1.4 Entropy Measures

The concept on which entropy measures are based is similar to that of probabilistic dependence. By observing $\xi$, the a posteriori probabilities $P(\omega_i \mid \xi)$ can be used to determine how much information has been gained from the experiment. If all classes become equally probable, then the information gain is minimal or uncertainty (entropy) is maximum. By implication, the classes will be overlapped. On the other hand, if the entropy is low then the overlap will be small. Therefore, it is obvious that the entropy value can be used as an indicator of class separability especially for multiclass problems.

2.1.5 Interclass Distance Measures

Given a set of patterns which is a representative of the mixture distribution (training set), it is reasonable to assume that the pattern vectors of each class occupy a distinct region in the observation space. The average pairwise distance (interclass distance) between the patterns in the set is then a measure of class separability in the space. The greater the distance between patterns of different classes, the better the separability of the two classes. Since information about class conditional density functions $p(\xi \mid \omega_i)$ is not used in defining the interclass distance, separability measures based on this concept cannot serve as true indicators of mutual class overlap. However, because of its computational simplicity a number of popular feature evaluation criteria are based on the heuristic notion of interclass distance, even though their relationship to error probability, in general, is very loose.
2.2 Most Often Used Distance Measures

Most of the works in which experiments with various feature evaluation criteria have been reported conclude that the feature rankings induced by the various criteria are very similar [2], [6], [48], [50]. This finding suggests that if we decide to avoid the use of the probability of error as a criterion, then computational efficiency should be the key factor in determining the alternative feature evaluation criterion to be used. Among the alternative feature evaluation criteria discussed, the most commonly adopted ones are the probabilistic distance measures [18], [26], [28]. This is because of ease of computation and largely due to their close relationship to the probability of error especially in the two-class case. Furthermore, a number of these measures can be explicitly calculated in the case when the class conditional density functions \( p(\xi \mid \omega_i) \) for a family of distributions such as Gaussian, multinomial and Poisson distributions, etc., are parametric. The most widely used probabilistic distance measures and their expression in two-class case are given in equations (2.8) – (2.11). A full list of such measures may be found in [30].

**Chernoff**

\[
J_C = -\ln \int_{-\infty}^{\infty} p'(\xi \mid \omega_1) p^{1-s}(\xi \mid \omega_2) d\xi, \quad s \in [0, 1]. \tag{2.8}
\]

**Bhattacharyya**

\[
J_B = -\ln \int_{-\infty}^{\infty} [p(\xi \mid \omega_1) p(\xi \mid \omega_2)]^{1/2} d\xi. \tag{2.9}
\]

**Divergence**

\[
J_D = \int_{-\infty}^{\infty} [p(\xi \mid \omega_1) - p(\xi \mid \omega_2)] \ln \left[ \frac{p(\xi \mid \omega_1)}{p(\xi \mid \omega_2)} \right] d\xi. \tag{2.10}
\]

**Patrick-Fisher**

\[
J_P = \left( \int_{-\infty}^{\infty} [p(\xi \mid \omega_1) - p(\xi \mid \omega_2)]^2 d\xi \right)^{1/2}. \tag{2.11}
\]

The best feature subset is selected in the sense of maximizing a prespecified distance measure. As far as the computational difficulty is concerned, the Bhattacharyya distance and the divergence are easier to compute than the other distance measures. Consequently, only these two distance measures will be discussed in more detail. However, it should be noted that none of the distance measures take into account the particular classifier structure for which the features are intended. It is well known, for example,
Chapter 2: Feature Selection Criteria

that the best features for a quadratic classifier may be worthless for a linear classifier; yet a distance measure would have both classifiers using the same feature sets.

2.2.1 Interpretation of The Bhattacharyya Distance and The Divergence

Bhattacharyya

The integral in equation (2.9) is sometimes referred to as the Bhattacharyya coefficient $\rho_B$ and if it is rewritten as

$$
\rho_B = \int_{-\infty}^{\infty} \sqrt{p(\xi | \omega_1)p(\xi | \omega_2)} d\xi
= \mathbb{E}[\sqrt{p(\xi | \omega_1)p(\xi | \omega_2)} | \omega_1]
$$

(2.12)

then the Bhattacharyya distance can be graphically demonstrated in Figure 2.5. Note that if the original class conditional density functions are well separated, then the expectation of the $\sqrt{p(\xi | \omega_1)/p(\xi | \omega_2)}$ with respect to $\omega_2$ will give a low value ($\ll 1$). Thus the Bhattacharyya distance, which is the negative logarithm of this quantity, will be high. On the other hand, if the class conditional density functions overlap, then the expectation will tend to give a high value ($> 1$) and the Bhattacharyya distance will correspondingly be low.

Divergence

The divergence in equation (2.10) can be rearranged as

$$
J_D = \int_{-\infty}^{\infty} \ln \left[ \frac{p(\xi | \omega_1)}{p(\xi | \omega_2)} \right] p(\xi | \omega_1) d\xi - \int_{-\infty}^{\infty} \ln \left[ \frac{p(\xi | \omega_1)}{p(\xi | \omega_2)} \right] p(\xi | \omega_2) d\xi.
$$

(2.13)

In addition, if we denote the log-likelihood ratio between two class conditional density function by $\Lambda(\xi)$, that is

$$
\Lambda(\xi) = \ln \left[ \frac{p(\xi | \omega_1)}{p(\xi | \omega_2)} \right].
$$

(2.14)

Then equation (2.13) can be rewritten as

$$
J_D = \mathbb{E}[\Lambda(\xi) | \omega_1] - \mathbb{E}[\Lambda(\xi) | \omega_2]
$$

(2.15)

which can be interpreted as a measure of the difference between the means of the two log-likelihood ratios, $\eta_2 - \eta_1$, as shown in Figure 2.6. More explicitly, if we denote the
Most Often Used Distance Measures

Figure 2.5: The interpretation of the Bhattacharyya distance (a) a typical pair of 1-D density functions $p(\xi | \omega_1)$ and $p(\xi | \omega_2)$ and (b) the corresponding plot of $\sqrt{p(\xi | \omega_1)/p(\xi | \omega_2)}$.

Figure 2.6: The interpretation of the divergence.
averaged relative information of class \( i \) with respect to class \( j \) by \( H(i,j) \) which is defined as

\[
H(i,j) = \int_{-\infty}^{\infty} \ln\left( \frac{p(\xi | \omega_i)}{p(\xi | \omega_j)} \right) p(\xi | \omega_j) d\xi
\]

\[
= \mathbb{E} \left[ \ln\left( \frac{p(\xi | \omega_i)}{p(\xi | \omega_j)} \right) | \omega_j \right]
\]

then from equations (2.13) and (2.16) it can be seen that

\[
J_D = H(1,2) + H(2,1).
\]

Therefore if the relative information is high, the divergence and the separation of the classes are also high. Furthermore, it is interesting to note that when the two classes are separable, i.e. \( p(\xi | \omega_1) = 0 \) if \( p(\xi | \omega_2) > 0 \) and vice versa, the patterns may be classified without error and \( J_D = \infty \). On the other hand, when \( p(\xi | \omega_1) = p(\xi | \omega_2) \), the two classes are indistinguishable and \( J_D = 0 \).

Note that the divergence seems to have the intuitive qualities of a distance measure, but the use of the word “distance” in connection with the Bhattacharyya distance seems more difficult to accept. However, both the Bhattacharyya distance and the divergence are additive when the components of \( \xi \) are conditionally statistically independent, i.e. they can be expressed as a sum of similar terms with each term involving only one of the components of \( \xi \). Moreover, in case of feature extraction, both are invariant under a one-to-one transformation of the vector \( \xi \).

### 2.2.2 Parametric Separability Measures

It is well known that the above distance measures can be analytically simplified if the class conditional probability density functions are parametric and in particular normal (Gaussian), that is when the class conditional density function \( p(\xi | \omega_i) \) is defined as

\[
p(\xi | \omega_i) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_i|}} \exp\left\{ -\frac{1}{2} (\xi - \mu_i)^T \Sigma_i^{-1} (\xi - \mu_i) \right\}
\]

where the class \( i^{th} \) \( D \)-component mean vector \( \mu_i \) is expressed as

\[
\mu_i = \mathbb{E} [\xi] = [\mu_{i1}, \mu_{i2}, \cdots, \mu_{id}]^T
\]
and the class \(i^{th}\) D-by-D symmetric covariance matrix \(\Sigma_i\) is defined by

\[
\Sigma_i = E \left[ (\xi - \mu_i)(\xi - \mu_i)^T \right] = \begin{bmatrix}
\sigma_{i1} & \sigma_{i2} & \cdots & \sigma_{iD} \\
\sigma_{i2} & \sigma_{i3} & \cdots & \sigma_{iD} \\
\vdots & \vdots & \ddots & \vdots \\
\sigma_{iD} & \sigma_{iD} & \cdots & \sigma_{iD}
\end{bmatrix}
\]  

(2.20)

where \((\xi - \mu_i)^T\) is the transpose of \((\xi - \mu_i)\), \(\Sigma_i^{-1}\) is the inverse of \(\Sigma_i\) and \(|\Sigma_i|\) is the determinant of \(\Sigma_i\). Equation (2.18) is normally abbreviated as \(p(\xi|\omega_i) \sim N_D(\mu_i, \Sigma_i)\). It should be noted that \(\Sigma_i\) is always positive semidefinite. The diagonal elements of \(\Sigma_i\) are the variances of the class \(i^{th}\) vector components. If all the off-diagonal elements are zero, \(p(\xi|\omega_i)\) reduces to the product of the univariate normal densities for the components of \(\xi\). For this special case the diagonal variance terms \(\sigma_{im}\) may be conventionally redefined as the standard deviations \(\sigma_i\) such that \(\sigma_{im} = \sigma_i^2\). Therefore, the class conditional density function \(p(\xi|\omega_i)\) can be expressed as

\[
p(\xi|\omega_i) = \frac{1}{\sqrt{(2\pi)^D \sigma_1^2 \sigma_2^2 \cdots \sigma_D^2}} \exp\left\{ -\frac{1}{2} \sum_{j=1}^{D} \frac{(\xi_j - \mu_j)^2}{\sigma_j^2} \right\}.
\]  

(2.21)

In feature selection the assumption of Gaussian distribution of the data is generally made even though it may not have a multivariate Gaussian distribution. This is because the assumption of normality is a reasonable approximation for many real data sets, and is particularly true for cases where random variables are sums of many variables and the central limit theorem can be applied. Moreover, there are many important properties of normal distributions, such as the expected vector \(\mu\) (equation (2.19)) and covariance matrix \(\Sigma\) (equation (2.20)) are the only two parameters required to sufficiently characterize a normal distribution uniquely. For the complete listing of important properties of normal distributions see [18]. However, it should be noted that the assumption should not be used without good justification. The resulting two-class probabilistic distance measures in their parametric form are given in equations (2.22) and (2.23) (the derivations can be found in Chapter 9 of [46]).

**Bhattacharyya**

\[
J_B = \frac{1}{4} \Delta \mu^T (\Sigma_1 + \Sigma_2)^{-1} \Delta \mu + \frac{1}{2} \ln \left[ \frac{1}{\sqrt{|\Sigma_1||\Sigma_2|}} \right] \left\{ \frac{1}{\sqrt{\left( \Sigma_1 + \Sigma_2 \right)}} \right\}
\]  

(2.22)

where \(\Delta \mu = \mu_2 - \mu_1\).
Divergence

\[ J_D = \frac{1}{2} \Delta \mu^T (\Sigma_1^{-1} + \Sigma_2^{-1}) \Delta \mu + \frac{1}{2} \text{tr} \{ \Sigma_1^{-1} \Sigma_2 + \Sigma_2^{-1} \Sigma_1 - 2I \}. \]  

(2.23)

Note that these expressions do not involve any integration in multidimensional space and can, therefore, be implemented more easily. In the very special case of two normally distributed classes with identical covariance matrices, i.e. \( \Sigma_1 = \Sigma_2 = \Sigma \), both the Bhattacharyya distance and the divergence can then be simplified even further to a simple criterion function known as Mahalanobis distance. Furthermore, the Mahalanobis distance can be used in general case even if \( \Sigma_1 \neq \Sigma_2 \) by letting \( \Sigma = P (\omega_1) \Sigma_1 + P (\omega_2) \Sigma_2 \) and the resulting distance is known as generalized Mahalanobis distance.

Mahalanobis

\[ J_M = \Delta \mu^T \Sigma^{-1} \Delta \mu. \]  

(2.24)

As seen in equation (2.22), the Bhattacharyya distance consists of two terms. The first or second terms disappear when \( \mu_1 = \mu_2 \) or \( \Sigma_1 = \Sigma_2 \), respectively. Therefore, the first term gives the class separability due to the mean-difference, while the second term gives the class separability due to covariance-difference. It is important to know which terms is dominant because that determines what type of classifier must be adopted for a given distribution [18].

The divergence measure is similar to the Bhattacharyya distance but easier to compute. Most properties can be discussed in terms similar to those used for the Bhattacharyya distance. Therefore, it is sometimes quite useful to use the divergence for theoretical discussions. However, it will be seen later that its weaker link to the Bayes probability of error which affects its wider application.

2.2.3 Recursive Calculation of Parametric Separability Measures

The most common characteristics of all search procedures which will be discussed in the following chapter is that the best feature set is constructed by adding (or removing) a smaller number of features to the current \( k \)-feature set \( \chi_k \). In the case of parametric probabilistic distance measures, the evaluation of the criterion function normally involves the computation of simple matrix functions such as the inverse, trace and the determinant. It is then highly desirable to evaluate the updated distance \( J(\chi_{k+1}) \) or \( J(\chi_{k-1}) \) by modifying the distance \( J(\chi_k) \) obtained from the previous stage. This operation is known as recursive
calculation of parametric separability measures which proves to be highly beneficial in saving computer time. A fundamental identity in matrix algebra [43] establishes such relationships which can be defined as follows:

Let $\Sigma_i$ and $\Psi$ be real $n$-by-$n$ symmetric definite matrices, i.e.

$$
\Sigma_i = \begin{bmatrix}
\sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\
\sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn}
\end{bmatrix},
$$

$$
\Psi = \begin{bmatrix}
\psi_{11} & \psi_{12} & \cdots & \psi_{1n} \\
\psi_{21} & \psi_{22} & \cdots & \psi_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\psi_{n1} & \psi_{n2} & \cdots & \psi_{nn}
\end{bmatrix}.
$$

In addition, denote by $\hat{\Sigma}_i$ and $\hat{\Psi}$ the submatrices of $\Sigma_i$ and $\Psi$ obtained by deleting the $k^{th}$ row and column of $\Sigma_i$ and $\Psi$ respectively. Further let $\Sigma_i^{-1}$ be represented as

$$
\Sigma_i^{-1} = \begin{bmatrix}
\Theta_{11} & \Theta_{12} & \cdots & \Theta_{1n} \\
\Theta_{21} & \Theta_{22} & \cdots & \Theta_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\Theta_{n1} & \Theta_{n2} & \cdots & \Theta_{nn}
\end{bmatrix}.
$$

Then the following relationships hold:

$$
|\hat{\Sigma}_i| = \theta_{ik} |\Sigma_i|
$$

$$
\hat{\Sigma}_i^{-1} = S - \frac{\hat{\Theta}_k \hat{\Theta}_k^T}{\theta_{ik}}
$$

$$
\text{tr}\{\hat{\Psi} \hat{\Sigma}_i^{-1}\} = \text{tr}\{\Psi \Sigma_i^{-1}\} - \frac{1}{\theta_{ik}} (\Theta_k^T \Psi \Theta_k)
$$

$$
\theta_{ik} = [\sigma_{ik} - \sigma_{ik}^T \hat{\Sigma}_i^{-1} \sigma_{ik}]^{-1}
$$

$$
\Sigma_i^{-1} = \begin{bmatrix}
\hat{\Sigma}_i^{-1} + \theta_{ik} \hat{\Sigma}_i^{-1} \sigma_{ik} \sigma_{ik}^T \hat{\Sigma}_i^{-1} & -\theta_{ik} \hat{\Sigma}_i^{-1} \sigma_{ik} \\
-\theta_{ik} \sigma_{ik}^T \hat{\Sigma}_i^{-1} & \theta_{ik}
\end{bmatrix}
$$

where $\hat{\Theta}_k$ is the $k^{th}$ column of $\Sigma_i^{-1}$ with $k^{th}$ element $\theta_{ik}$ omitted, i.e.

$$
\hat{\Theta}_k = [\theta_{ik}, \cdots, \theta_{i(k-1)k}, \theta_{i(k+1)k}, \cdots \theta_{ik}]^T
$$
and $S$ is the submatrix of $\Sigma_i^{-1}$ defined as

$$S = \begin{bmatrix}
\hat{\theta}_{i_1} & \cdots & \hat{\theta}_{i_{n-1}} & \hat{\theta}_{i_{k+1}} & \cdots & \hat{\theta}_{i_n} \\
\theta_{n1} & \cdots & \theta_{n(k-1)} & \theta_{n(k+1)} & \cdots & \theta_{n1n} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\theta_{n(k-1)} & \cdots & \theta_{n(k-1)(k-1)} & \theta_{n(k-1)(k+1)} & \cdots & \theta_{n(k-1)2n} \\
\theta_{n(k+1)} & \cdots & \theta_{n(k+1)(k-1)} & \theta_{n(k+1)(k+1)} & \cdots & \theta_{n(k+1)2n} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\theta_{n1} & \cdots & \theta_{n(k-1)} & \theta_{n(k+1)} & \cdots & \theta_{n1n}
\end{bmatrix}.$$  

(2.34)

Kittler [29], in addition, has derived the expressions allowing for a fast evaluation of separability measures in an $(k - 1)$-dimensional space from the values of these measures in the $k$-dimensional space using the relationships in equations (2.28) - (2.30). The summary of the various recursive forms corresponding to the $k^{th}$ measurement omitted are as follows:

**Bhattacharyya**

Let $\Sigma = (\Sigma_1 + \Sigma_2)/2$ and $\nu = \Delta \mu \Delta \mu^T$ then equation (2.22) can be rewritten as

$$J_B = \frac{1}{8} \text{tr}\{\nu \Sigma^{-1}\} + \frac{1}{2} \ln|\Sigma_1| - \frac{1}{4} \ln(|\Sigma_1||\Sigma_2|).$$  

(2.35)

If $k^{th}$ measurement is omitted, then the new distance $\bar{J}_B$ can be recursively computed using

$$\bar{J}_B = J_B - \frac{1}{8 \theta_{kk}} (\Delta \mu^T \hat{\Theta}_k)^2 + \frac{1}{2} \ln \theta_{kk} - \frac{1}{4} \ln(\theta_{1ak} + \theta_{2ak}).$$  

(2.36)

**Divergence**

Equation (2.23) can alternatively be rewritten as

$$J_D = \frac{1}{2} \text{tr}\{\nu (\Sigma_1^{-1} + \Sigma_2^{-1}) + \Sigma_1^{-1} \Sigma_2 + \Sigma_2^{-1} \Sigma_1 - 2I\}.$$  

(2.37)

Then an expression for divergence $\bar{J}_D$ in the subspace corresponding to the omission of $k^{th}$ measurement is

$$\bar{J}_D = J_D - \frac{1}{2 \theta_{1ak}} [\Theta^T_1 (\nu + \Sigma_2) \Theta_1] - \frac{1}{2 \theta_{2ak}} [\Theta^T_2 (\nu + \Sigma_1) \Theta_2].$$  

(2.38)
Most Often Used Distance Measures

Mahalanobis

$J_M$ in equation (2.24) can be rewritten as

$$J_M = \text{tr}(\nu \Sigma^{-1}).$$  \hfill (2.39)

If the $k^{th}$ measurement is omitted, then the Mahalanobis distance between classes in the reduced space, $\hat{J}_M$, is given as

$$\hat{J}_M = J_M - \frac{1}{\theta_{kk}} (\Delta \mu^T \Theta_k)^2.$$  \hfill (2.40)

The expressions for a fast evaluation of separability measures in an $(k+1)$-dimensional space from the values of these measures in the $k$-dimensional space can be derived in a similar way using equations (2.30) – (2.32).

2.2.4 Comparison of Error Bounds

Some experimental results from the comparison of the effectiveness of several feature criteria have been reported in [17]. While experimental results are important in evaluating the feature effectiveness, one possible drawback with the experimental comparison is that incorrect assumptions of the underlying distributions of the patterns may lead to inconsistent or erroneous conclusions. In contrast, theoretical comparison of the relative merits of various distance measures by evaluating the error bounds should provide unique results. Note that both the upper and lower bounds to the error probability of the Bayes classifier $P_e$ in terms of a separability measure are indicative of how closely the measure approximates $P_e$ [8], [33]. Consequently, the performance of feature sets selected by various feature selection criteria can be computed theoretically on the basis of such bounds. The reliability of a separability measure depends, of course, on how tightly it bounds the error probability. Normally the tighter the error bounds a distance measure can provide, the better features it can select.

Kailaith [25] and especially Chen [8] provide theoretical comparison of feature selection criteria using error bounds. They can be summarized as follows.
Chapter 2: Feature Selection Criteria

Bhattacharyya

There exist both the upper and lower bounds in terms of the Bhattacharyya distance $J_B$ which can be expressed as

$$
\frac{1}{2} \left[ 1 - \sqrt{1 - 4P(\omega_1)P(\omega_2)\exp(-2J_B)} \right] \leq P_e \leq \sqrt{P(\omega_1)P(\omega_2)\exp(-J_B)}. \tag{2.41}
$$

If $P(\omega_1) = P(\omega_2)$ and as $J_B \to \infty$ (low $P_e$) the difference between the upper and lower bounds tends to zero, i.e. the bounds are tighter. It should be noted that the Bhattacharyya distance is actually a special case of a more general distance measure (equation (2.8)) introduced by Chernoff [9]. This measure is in general closer to the probability of error than the Bhattacharyya distance, but it is usually not as easy to evaluate as the Bhattacharyya distance.

Divergence

No similar bounds in terms of the divergence $J_D$ appears to be generally true. However, by assuming $P(\omega_1) = P(\omega_2)$ and using an inequality between $J_B$ and $J_D$ a crude lower bound on $P_e$ in terms of $J_D$ can be obtained

$$
P_e \geq \frac{1}{8} \exp(-2J_B) \geq \frac{1}{8} \exp(-J_D/2). \tag{2.42}
$$

In addition, it is possible to find the tighter relations between $J_B$ and $J_D$ in particular cases, e.g., when $p(\xi|\omega_1)$ and $p(\xi|\omega_2)$ are Gaussian. The bounds become

$$
P_e \leq \frac{1}{2} \exp(-2J_B) < \frac{1}{2} \exp(-J_D/4)^{-1/2}. \tag{2.43}
$$

It should be noted that tighter upper and lower error bounds are available with $J_B$. This is an advantage of $J_B$ over $J_D$. Moreover, the divergence also suffers from another drawback since the divergence is defined by the difference in the mean values of the log-likelihood ratio, and a discontinuity can occur when one of probability densities becomes equal to zero. This problem does not arise in the Bhattacharyya distance.
Mahalanobis

The upper bound on $P_e$ in terms of the Mahalanobis distance $J_M$ is expressed as

$$\frac{2P(\omega_1)P(\omega_2)}{1 + P(\omega_1)P(\omega_2)J_M} \geq P_e.$$  \hspace{1cm} (2.44)

Unfortunately, this bound is not very tight except when $P_e$ is large. However, this bound is very easy to calculate.

It follows from the above relations and especially when the family of probability density functions is known (parametric case) that it is probably advantageous to use distance measures for feature selection [35]. However, it must be pointed out that there are many more feature evaluation criteria that are relatively simple, e.g., discriminant analysis methods [38], [41] but one disadvantage is the fact that these criteria do not have a direct relationship to the Bayes probability of error. In addition to providing general guidance on the effectiveness of feature selection criteria, the ability to estimate statistical bounds on the probability of error may prove beneficial in reducing the development cost. Note that the upper bounds indicate when an automatic pattern classification system is satisfactory, and lower bounds signal when it is inadequate. The former is valuable when it is cheap to overdesign, by using a large number of measurements to train the system. The latter is valuable when overdesigning is costly; the lower bound would indicate, for instance, that more measurements were needed. By taking advantage of both types of bounds, the necessity of test samples, above and beyond the training samples is minimized.

2.3 Feature Evaluation Criteria for Multiclass

So far, only the distance measures between two classes have successfully been established. Unfortunately, there is no obvious way of extending the distance measures into the $m$-class problem when $m > 2$. However, it is known that any $m$-class problem can be treated as a series of 2-class problems, therefore by generalizing the pairwise distance measures it is possible to obtain a feature selection criterion for $m$-class problems [31], [32]. For example, a reasonable indication of the class separability for $m$ classes can be obtained by evaluating a weighted average of the pairwise distances $J_{ij}(\chi)$, that is,

$$J(\chi) = \sum_{i=1}^{m} \sum_{j=i+1}^{m} P(\omega_i)P(\omega_j)J_{ij}(\chi).$$  \hspace{1cm} (2.45)
Chapter 2: Feature Selection Criteria

This extension for the multiclass case was used by Fu et al [17] for Kullback-Leibler divergence measure, by Lainiotis [31] for the Bhattacharyya distance, and by Toussaint [47] for the Kolmogorov variational distance. Alternatively, the generalized distance measure can be defined as the average pairwise distance, i.e.

\[ J(z) = \frac{2}{m(m-1)} \sum_{i=1}^{m} \sum_{j=i+1}^{m} J_{ij}(z). \]  

(2.46)

The major disadvantage of these two approaches is that one large value of \( J_{ij}(z) \) may dominate the value for \( J(z) \) and impose ranking which reflects only the distance between the two most separable classes.

In order to avoid this disadvantage Grettenberg [22] proposed another approach, also based on the \( J_{ij}(z) \) values, which suggests to prefer \( X' \) to \( X'' \) if \( X' \) discriminates better between the two most confusing pair of classes, i.e. if

\[ \min_{i,j} J_{ij}(X') > \min_{i,j} J_{ij}(X''). \]  

(2.47)

By its definition, the drawback of this approach is that it takes into account the distance between the closest pair only.

It is apparent that these criteria are rather complex and their implementation would, therefore, be computationally very demanding. More importantly, it should be noted that the criteria based on pairwise distance measures are no longer retaining their close relationship with the error probability. However, their lower bound cannot be smaller than the lower bound at \( m=2 \). Finally, although there are some natural multiclass criteria, such as entropy measures and dependence measures etc., which do not require any generalization, they are not widely used since they are considered to be more costly to implement than the generalized ones even when \( p(\xi|\omega_i), i = 1, 2, \cdots, m \) are Gaussian [28].

2.4 Finite Sample Size Effects

The main purpose of the studies of error bounds is the construction of easily computable error bounds in the case of known class distributions. These error bounds are therefore sample size independent. However, in many practical pattern recognition problems, the underlying class conditional probability densities are either partially or completely unknown. Consequently, they have to be estimated from information based on a finite,
usually small, number of training samples which are available for each class. The estimated parameters are often obtained with low accuracy unless an infinite number of training samples are used. Experimental results indicate the statistical trap involved in using these error bounds as the true error bounds when the ratio of sample size to feature size is small. Therefore, these error bounds are almost useless since they do not take into account the error made by the estimation of the distribution.

2.4.1 A Sample Size Dependent Error Bound

In an effort to isolate the effect of finite sample size on expected probability of error, Duin [14] derived an upper bound on the two-class classification problem with finite learning sets. This error bound is expressed as

\[ U(P_e) \leq P_e^* + U(e_1) + U(e_2) \]  

(2.48)

where \( P_e^* \) is the optimal Bayes error and \( e_i, i = 1, 2 \), are the errors made in estimating the corresponding class conditional densities which are defined such that \( 0 \leq U(e_i), i = 1, 2, \leq 1 \). It is obvious that \( U(e_i) \) is a function of the sample size, dimensionality and the true underlying density. However, it is not clear how tight these bounds are.

2.4.2 Predicting the Minimum Training Size

Significant research efforts, e.g. [1], [10], [15], [23], [39], [44], [45], have been made to find the relationship between the probability of error, the dimensionality of features, the number of training samples, and the complexity of classification procedures, usually for multivariate Gaussian distributions. Although no explicit relationship between these quantities are given, the following guidelines for choosing the number of samples that are required to reasonably estimate parameters of a multivariate Gaussian distribution have been suggested.

Intuitively, it is generally known that in order for the estimate of the covariance matrix to be nonsingular, the number of samples must be greater than or equal to the dimensionality of the features. Cover [10] derived some results, which may be interpreted as follows: regardless of the true performance of a two-class classifier, if the total number of samples is less than twice the number of features, there exists a linear hyperplane such that the probability of error on the design set is always zero. Foley [15] considered this problem when the class conditional densities are multivariate Gaussian with unknown mean vectors and known common covariance matrix. He demonstrated that the design-
set error is an extremely biased estimate of the Bayes error rate if the ratio of samples per class $N$ to feature size $d$ is less than three. In addition, the variance of the design-set error rate is shown to be approximated by a function that is bounded by $1/8N$. This indicates that even if the number of feature is small, enough samples must still be used in order to minimize the variance of the design-set error rate.

Mehrotra [36] extended Foley’s results to situations where the common covariance matrix is also unknown and concluded that the ratio $(N/d)$ must be larger than five before the bias in the design-set error rate is sufficiently small. Fukunaga and Kessell [19] experimented with unlabelled test samples and recommended that the ratio $(N/d)$ should be at least ten. These findings confirm with that of Kanal and Chandrasekaran [27] who concluded that if less is known about the underlying probability structure, an even greater ratio of sample size to feature size is needed. This is also in accordance with the so-called Bellman’s “curse of dimensionality” [3], showing that the density of data in multidimensional space decreases dramatically with the increase of dimensionality. As a result, $2^d$ sample points may be theoretically evenly spread in $d$-dimensional space, i.e. one point for each half-unit of $d$-dimensional cube, which represents a very sparse set of samples. However, a reasonable engineering rule of thumb is to have five to ten times as many samples as features. Furthermore, as far as the number of training samples for each class is concerned, it is proved advantageous to have equal numbers of samples [7], [24], [42].

2.5 Summary

The important issues in feature selection criteria so far can be summarized as follows.

In order to select the best subset of $d$ features from a set of $D$ measurements, it is required to evaluate the chosen class separability measure $\binom{D}{d}$ times. This number is apparently astronomically high for a large number of measurements $D$. Consequently, the method ceases to be feasible. It will be seen in the following chapter that even for relatively small numbers of $D$ and $d$, the method is computationally demanding. It is, therefore, essential to choose the simplest criterion of all and thus reduce the number of computations as much as possible.

As pointed out earlier, the probability of classification error is an ideal criterion of feature set effectiveness. However, this criterion function cannot often be used in practical applications because of its computational complexity due to its form which precludes analytical simplification. Therefore, it is essential to resort to alternative evaluation criteria that may be weaker than the error probability, but are easier to evaluate and manipulate. The alternative evaluation criteria are usually defined as probabilistic distance measures,
probabilistic dependence measures, entropy measures or interclass distance measures, and the best feature subset is selected in the sense of maximizing a prespecified class separability measure.

As far as the reliability of a class separability measure is concerned, it depends, of course, on how tightly it bounds the error probability. Additionally, the calculation of the bounds must be practical since there is no point to calculate the bounds which are not simpler than the calculation of $P_e$. Ideally, the designer should consider this merit when choosing a suitable feature selection criterion. However, in practice the choice of a criterion function is based mainly on computational considerations. Taking these demands into account, the most commonly used class separability measures are the probabilistic distance measures. In addition to the ease of computation and their close relationship with the probability of error, another very important aspect of these criteria is that a number of them can be analytically simplified if the classes have a parametric distribution. More importantly, in a special case when the class conditional density functions are Gaussian, two distance measures become particularly useful, the Bhattacharyya distance and the divergence. The computational complexity of a parametric criterion functions is given by the number of matrix inversions and determinant evaluations for each set of $d$ features.

As far as the computational complexity is concerned, the divergence measure is easier to compute than the Bhattacharyya distance, but due to weaker link to the Bayes probability of error which precludes the divergence from wider applications. Additionally, if the classes are distributed normally with identical covariance matrices, then the implementation of the Mahalanobis distance is even simpler than the computation of the divergence. However, this situation seldom happens hence the practical use of this criterion is rather limited.

In practice, the effectiveness of a feature set must be estimated from the available samples. If the number of available samples is small compared to dimensionality of features, these estimates will not be very reliable. Realizing the deteriorating effects, the designer of a pattern recognition system should make every possible effort to obtain as many samples as possible. However, there are many pattern recognition problems where either the number of samples is limited or obtaining a large number of samples is extremely expensive. It is this small sample size problems where the designer of a recognition system has to be extremely careful. However, the general guideline for having five to ten times as many samples as features seems to be a good practice to follow.

Finally, when solving the problem of feature selection in a multiclass case, the de-
Signer should be aware of the fact that a particular subset of features may be useful for discriminating some pairs of classes, whereas that subset of features may not be pertinent for separating other classes. Thus the globally best feature subset may not be the best subset for separating a specific pair of classes in a local feature space. Even with the worst features in the overall sense may be useful for discriminating a specific pair of classes. This means that it may be more effective to split the multiclass problem into a series of dichotomic problems (if it is feasible) and to solve a feature selection for a series of dichotomic classification problems.

References


Chapter 3

A Review of Feature Set Search Procedures

If one can find an appropriate probabilistic distance measure for feature subsets, feature selection is reduced to a search problem that detects an optimal feature subset based on the selected measure. Cover [7] has shown that in order to guarantee the finding of an optimal subset of \( d \) features from the given \( D \) measurements, exhaustive search is a necessary procedure. This is trivially true because exhaustive search examines all \( \binom{D}{d} \) subsets of size \( d \). However, in many practical cases the values of \( d \) and \( D \) result in the number of possible subsets that are too large, i.e. the search is too long, the required memory is too large, or both. For example, in a typical texture classification problem, we might be trying to select 10 features out of say 60 available measurements which would require evaluation of more than \( 7.54 \times 10^{10} \) feature sets. Obviously, exhaustive search seems to be prohibitive in such an application. Therefore, some computationally feasible procedures to avoid the exhaustive search are essential even though the feature set obtained may be suboptimal.

For the above reason, the question of the trade-off between the optimality and efficiency of algorithms for feature selection is recognized, and the mainstream of research on feature selection has thus been directed toward suboptimal search methods. Cover [7] has also pointed out the potential of any non-exhaustive procedure to select the worst possible set of features. This fact prevents any claim of optimality for any non-exhaustive method. All that can be claimed is that certain methods are better than others. It should be noted that in contrast to feature extraction the search strategy for feature selection is independent of the criterion function used.
3.1 Preliminaries

In all the feature set search algorithms to be discussed in the sequel, the best feature set is constructed by adding to and/or removing from the current feature set, a small number of measurements at a time until the required feature set, \( X_d \), of cardinality \( d \) is obtained. More specifically, to form the best \( d \)-feature set the starting point of the search can be either an empty set, \( X_0 \), which is then successively built up or, alternatively, the starting point can be the complete set of measurements, \( Y \), in which superfluous measurements are successively eliminated. The former approach is referred to as the "bottom up" search while the latter is known as the "top down" method.

In order to describe the various algorithms, let \( \chi_k \) be a set containing \( k \) elements, \( \xi_1, \xi_2, \cdots, \xi_k \), from the complete set of available measurements, \( Y \), i.e.

\[
\chi_k = \{ \xi_i | i = 1, 2, \cdots, k; \xi_i \in Y \} \tag{3.1}
\]

Further denote by \( \bar{\chi}_k \) the set of \( D-k \) features obtained by removing \( k \) measurements, \( \xi_1, \xi_2, \cdots, \xi_k \), from the complete set of measurements, \( Y \), i.e.

\[
\bar{\chi}_k = \{ y_i | y_i \in Y, 1 \leq i \leq D; y_i \neq \xi_j, \forall j \} \tag{3.2}
\]

Obviously, different sets \( \chi_k \) and \( \bar{\chi}_k \) are obtained by choosing different measurements \( \xi_i \). In addition, the following notations are defined:

\[
X_0 = \chi_0 \equiv \emptyset
\]

\[
\bar{X}_0 = \bar{\chi}_0 \equiv Y
\]

All bottom up and top down search algorithms are based on the assumption of monotonicity of the feature selection criterion function [18]. The monotonicity condition requires that for nested feature sets \( \chi_1, \chi_2, \cdots, \chi_k \), i.e.

\[
\chi_1 \subset \chi_2 \subset \cdots \subset \chi_k
\]

the criterion function \( J \) satisfies

\[
J(\chi_1) \leq J(\chi_2) \leq \cdots \leq J(\chi_k)
\]

In fact, all the probabilistic distance measures mentioned in the previous chapter are known to possess this property.
3.1.1 Earlier Work

Most of the published techniques for automatic feature selection are parametric in the sense that the criterion functions used in these techniques are based on the assumption that the probability densities of the training sets are known or that some a priori information about the data is available. Examples of these criterion functions are Bayes error probability, Bhattacharyya distance, and divergence. In some cases, however, little is known about the underlying probability distributions, and performance must be estimated directly. The following is a brief review of some of the major work on feature selection during the past three decades.

The pioneering work in the area of feature selection is associated with the names of Sebestyen [29], Lewis [22] and Marill and Green [23] who made their contribution in the early sixties. Since at that time the theoretical framework for evaluating the error rate of classifiers was in its preliminary stage of development, the original approaches to feature selection were based on interclass distance [29], on entropies [22] and on probabilistic measures of class separability [23]. In some cases the independence of features was assumed and the features were selected on the basis of their individual merit. An example of this method is given by Mucciardi and Gose [25]. The major problem with this method is that it does not take the statistical dependency among features into account. Consequently, even such a simplified model did not guarantee the optimality of a selected feature subset (for instance, the best two independent features don’t have to be the two best if they are binary-valued features, as was pointed out by Cover [6]). The need to consider the statistical dependency among features is discussed by Elashoff et al. [9], and by Michael [24].

In 1963 Marill and Green [23] introduced a feature selection technique using the divergence distance as the criterion function and the sequential backward selection (SBS) method as the search algorithm. The shortcomings of this feature selection technique are that the divergence [20] is used as the criterion function which requires a priori knowledge of the forms of distribution (usually Gaussian) and selected features are nested, i.e., discarded features are not re-selected. The nesting constraint prevents the method from finding optimal subsets. Another feature selection technique known as the sequential forward selection (SFS) was introduced in 1971 by Whitney [34]. It is the counterpart of the SBS technique. Even though direct error estimate was the criterion function used in this technique, it has the shortcomings that the search is generally suboptimal and suffers from the nesting property.

In 1973 a method of preventing nesting of feature sets, but still suboptimal, was pro-
posed by Chang [4]. It is based on the concepts of dynamic programming which offers one means of constructing very powerful search procedures. Chang's results are impressive, despite the fact that his experiments were carried out with the divergence instead of the error probability as the evaluation criterion. However, this method has not been heavily pursued by other researchers due to numerous restrictive requirements, e.g. the monotonicity condition and statistical independence of features. Additional discussion of how dynamic programming can be applied to feature selection for sequential recognition systems, a subject beyond the scope of this thesis, can be found in [11].

There is, however, a way to prevent nested subsets which is much simpler and computationally more attractive than the methods of dynamic programming. This method was first put forward by Michael and Lin [24] in the context of Whitney's sequential forward selection. The idea was later refined and developed into the Plus-l-Minus r (l-r) search method (also suboptimal) by Stearns [33] in 1976. The main drawback of this method is that there is no theoretical way of predicting the values of l and r to achieve the best feature set. The search in this direction was concluded by introducing the generalization of SBS, SFS, and (l-r) algorithms proposed by Kittler [18] in 1978. In addition, a couple of more sophisticated heuristic procedures based on discriminant power [27] aimed at overcoming the drawbacks of SBS, SFS and (l-r) methods have been successfully employed in the PREDITAS system [3], [28]. They are known as sequential backward floating selection (SBFS) and sequential forward floating selection (SFFS) since the resulting dimensionality in respective stages of the algorithm is not changing monotonously but is actually "floating" up and down. The shortcoming of these methods is that the search schemes are generally suboptimal.

A real breakthrough in optimal set search came in 1977 with the introduction of the branch-and-bound algorithm which was proposed by Narendra and Fukunaga [26]. The optimality of the results in this method, however, is constrained by the fact that monotonic parametric distance measures (e.g. Bhattacharyya distance, divergence) must be used as the criterion function, i.e. the monotonicity condition must be satisfied. The branch-and-bound algorithm often makes practicable problems for which the exhaustive search would be totally out of the question. A shortcoming of this approach is that the criterion is not necessarily related to the performance of the classifier, particularly when the data is far from Gaussian and the class regions overlap. It is well known that the error rate of a classifier (if it is not a Bayes classifier) does not satisfy the monotonicity condition and the lack of monotonicity in the classifier's error rate makes it useless for the branch-and-bound procedure. However, Hamamoto et al [14] show that the proce-
Preliminaries

dure works well in terms of the classifier's error rate, even if the monotonicity condition is not satisfied.

As far as the discussed techniques for feature selection are concerned, in order to assess the effectiveness of a feature set of size $k$, all the measures of separability have to be evaluated in a $k$-dimensional space. Unless the distance measure is parametric, the evaluation involves probability density function estimate and then the computation of some distance between two or more densities over the whole feature space. It is clear that for feature spaces of reasonable size the amount of computation involved will be excessive. Motivated by this problem Backer and De Schipper [2] proposed a new method for evaluating the additional effectiveness of a new candidate feature when it is combined with the already selected feature set. The method involves only the computation of individual and pairwise merits, and when it is incorporated with the search in sequential forward selection manner it is known as the Max-Min algorithm. Although this algorithm has a computational advantage, since there is no possibility of evaluating the absolute quality of the selected feature set, the performance is not so impressive when comparing with other well known methods [5], [18].

3.1.2 Recent Search Strategies

Since the introduction of the branch-and-bound procedure, the major work on feature selection has been directed toward graph search procedures. Ichino and Sklansky [16] present an optimal feature selection method for a box classifier by reducing the branch-and-bound procedure to zero-one integer programming. This procedure is nonparametric, hence the effectiveness of a feature subset is evaluated directly with respect to the chosen classifier and the training sets. It is shown to be much more efficient than general branch-and-bound search in the use of heuristic information about solutions. However, the usefulness of this approach is still dependent on the computation speed and computer memory.

Foroutan and Sklansky [10] point out that in branch-and-bound search method the assumption of monotonicity is a major obstacle in many practical situations where the sample size is small and no a priori information about the data is available. They then introduced the concept of approximate monotonicity and demonstrated that a locally trained piecewise linear classifier of Sklansky and Michelotti [32] yields error rate functions that are only mildly nonmonotonic. Consequently, they successfully used a modified zero-one integer programming in [16] with the classifier's rate to search for the optimal feature subset. Although, the supporting tests were done only for one data set the
idea of approximate monotonicity constitutes another breakthrough in understanding and applying methods for optimal feature selection for certain classifiers.  

Recently, Siedlecki and Sklansky [31] have successfully applied genetic algorithms, a technique that simulates natural process in biology, to solve the problem of large-scale feature selection. It is a parallel test-and-go technique, in which a predefined number of possible solutions is modified, tested and stored at the same time. The idea is based on the assumption that large domains of data are organised and evolve in a manner similar to processes occurring in nature (for an overview see [13] and [15]). Alternatively, Siedlecki and Sklansky [30] have suggested the possibility of applying simulated annealing technique to solve large-scale feature selection problem. The simulated annealing takes its inspiration from statistical mechanics and the metallurgical technique of “annealing” – hardening a metal by slow cooling (see [1], [17] or [21]). The method attempts to produce a quality solution through nondeterministic hill-climbing. However, the optimality of the selected feature set from either the genetic algorithm or simulated annealing cannot be guaranteed.

According to the way that possible candidate feature sets are searched, it is obvious that the feature search procedures can be classified into 2 categories; one is the optimal and the other is the suboptimal search procedures. In the next section, various well known feature search procedures are discussed in detail.

### 3.2 Optimal Search Procedures

#### 3.2.1 Exhaustive Search

Let a set of candidate features of size $d$ be denoted by $\mathcal{X}_d$, and the set of “optimal features” in the sense of maximizing some criterion function $J$ by $X = \{x_j | j = 1, \ldots, d\}$. If an optimization is carried out over all possible candidate feature sets, i.e. performing an exhaustive search in which $J(X) = \max J(\mathcal{X}_d) \ \forall \mathcal{X}_d$, then it is guaranteed that the best subset of $d$ features from a complete set of $D$ measurements, $Y$, is chosen. However, this approach involves the evaluation of all the possible candidate feature sets $\mathcal{X}$ of size $d$ that can be constructed from measurements $y_j$. The search for the optimum is, therefore, a combinatorial problem, that is, the number of sets that need to be considered equals $\frac{D!}{(D-d)!d!}$. Consequently, the difficulty in finding the optimal feature set lies in the amount of computation which increases exponentially with the feature size. Hence, the disadvantage of this approach is that the number of sets to be searched is excessive even for moderate values of $D$ and $d$ which means that this option can very quickly cease to be computationally feasible.
3.2.2 Branch-and-Bound Algorithm

Feature set selection by exhaustive search can become computationally prohibitive. However, the branch-and-bound algorithm guarantees to select an optimal feature subset of size $d$ without involving explicit evaluation of all the possible combinations of $d$ measurements. It is basically a "top-down" search algorithm with a backtracking facility which allows all the possible combinations of features to be examined. The algorithm is applicable under the assumption that a feature selection criterion satisfies the monotonicity condition. Denoting by $\bar{x}_j$ the set of $D-j$ features obtained by removing $j$ measurements from the complete set of measurements, $Y$. The monotonicity condition implies that for nested feature subsets, $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_j$, related as

$$\bar{x}_1 \supset \bar{x}_2 \supset \cdots \supset \bar{x}_j,$$

the criterion function $J(\bar{x}_j)$ satisfies

$$J(\bar{x}_1) \geq J(\bar{x}_2) \geq \cdots \geq J(\bar{x}_j)$$

By a straightforward application of this property, many combinations of features can be rejected from the set of candidate feature sets.

Illustration is found to be the easiest way to introduce the basic idea behind the branch-and-bound algorithm. A problem of selecting the two best features out of five measurements is proposed as an example. The list of all the possible triplets of measurements, which include the ones that have to be discarded to obtain the optimal set of two features is as follows

$$\begin{align*}
\{1, 2, 3\} & \quad \{1, 2, 4\} & \quad \{1, 2, 5\} & \quad \{1, 3, 4\} & \quad \{1, 3, 5\} & \quad \{1, 4, 5\} \\
\{2, 3, 4\} & \quad \{2, 3, 5\} & \quad \{2, 4, 5\} & \quad \{3, 4, 5\}
\end{align*}$$

These triplets can be represented by a solution tree in which each node designates an eliminated measurement as shown in Figure 3.1. However, given a pair of numbers $(D,d)$ a large number of different solution trees could be constructed. Hence, the computational efficiency of the branch-and-bound method lies in an effective organization of the solution tree. It should be noted that the solution tree in Figure 3.1 is proved to be very efficient since it is not constructed level by level but from the least dense part to the part with most branches (from right to left).
Figure 3.1: Solution tree for a branch-and-bound algorithm.

To perform the branch-and-bound algorithm, firstly the right-most branch is generated and the magnitude of the feature criterion function at the terminal node is taken as the current best criterion value $J_0$, and this node defines the current best feature set. We then return to the nearest branching node and generate the next right-most branch of the tree. If the value of the criterion function at some node is less than $J_0$, then the branches originating from that node cannot possibly lead to the optimal feature set and need not be explored. This is because by virtue of the monotonicity property the elimination of additional measurements will only result in a further decrease of the criterion function value. Therefore, the algorithm backtracks to the nearest branching point in the lower level and the next right-most branch is then generated. The path illustrating this construction process for the solution tree in Figure 3.1 is marked in dashed line.

If, on the other hand, at any node the criterion function value exceeds $J_0$, then there is still a chance that a better feature set will be discovered and the search must, therefore, continue along the right most unexplored branch. If the bottom of the tree is reached and the corresponding criterion function value is greater than $J_0$, then this node defines the new best feature set and $J_0$ is updated accordingly. The algorithm then backtracks to the nearest branching point and the next section of the tree is then generated. This process is continued until the whole tree is constructed. Upon termination of the algorithm the current best feature set becomes the optimal feature set of the required cardinality, and the current best criterion value $J_0$ gives the optimal value of the criterion function.
In comparison with the exhaustive search, the branch-and-bound algorithm affords substantial computational saving. As a rule only a fraction of all the possible candidate feature sets need be explicitly enumerated to find the optimal feature set. This reduction factor of computation costs being most dramatic for \( d \approx D/2 \). Note, however, that in addition to the sets, \( \mathcal{X}_{D-d} \), which correspond to the terminal nodes of the fully generated branches, their supersets, \( \mathcal{X}_{D-d-j}, j = 1, 2, \ldots, D - d \), must also be evaluated. Thus, the actual number of inspected sets is somewhat higher. This implies that the branch-and-bound algorithm performs an exhaustive search in the entire feasible region. Therefore, its usefulness depends not only on fulfilling the requirement of the monotonicity condition, but also on the computational speed and computer memory. This should be borne in mind when deciding whether to use the branch-and-bound algorithm or opt for the exhaustive search which for very small values of \( d \) or values of \( d \) approaching \( D \) the calculation may be less involved as shown by Hamamoto et al [14]. The branch-and-bound search will be particularly efficient if the measurements \( y_j \) for the successor nodes to each node of the solution tree are selected from right to left in descending order of magnitude of the criterion function [8], [12], [18]. Additionally, as mentioned earlier in the previous chapter the time required to compute the criterion values of nested sets can be substantially reduced if they are computed recursively.

3.3 Suboptimal Search Procedures

In large-scale feature selection problem, even the powerful branch and bound search procedure may not be computationally feasible. Therefore, we have to employ various suboptimal search methods by considering the trade off between the optimality and computational efficiency. In the following, various search procedures which are relatively simple to implement including their reliability and drawbacks are discussed. It should be noted that the feature set yielded by a sophisticated suboptimal algorithm is likely to be better than the one that would be obtained by using the simplest suboptimal procedure, but this cannot always be guaranteed. Nevertheless, the likelihood of obtaining a better feature set when more complicated search schemes are employed is higher. This is because the total number of candidate feature sets explored by more sophisticated search methods is greater than that of simplistic procedures.
3.3.1 Feature Selection on the Individual Merit Basis

As mentioned before, it is well known that the set of \( d \) individually best features is not necessarily the best feature set of size \( d \) even for the case of statistical independent features [6]. Surprisingly, the only situation where this situation is true, i.e. the best feature set of size \( d \) comprises of \( d \) individually best features, is when the criterion function for that set can be expressed either as

\[
J(X_d) = \sum_{i=1}^{d} J(x_i) \quad (3.3)
\]

or

\[
J(X_d) = \prod_{i=1}^{d} J(x_i). \quad (3.4)
\]

Such a situation would arise, for example, if the Mahalanobis distance \( J_M \) or Bhattacharyya distance \( J_D \) were to be used in a problem concerning two normally distributed classes with diagonal covariance matrices. For this very special case, the optimal set of \( d \) features out of \( D \) measurements can then be determined by selecting \( d \) individually best measurements in \( Y \). This algorithm is known as the best \( d \)-feature algorithm which can be easily implemented as follows:

---

The Best \( d \)-feature Algorithm

- **Step 1**: Compute and rank the measurements \( y_j \) in the order of decreasing magnitude of the criterion function \( J(y_j) \) so that

\[
J(y_1) \geq J(y_2) \geq \cdots \geq J(y_d) \geq \cdots J(y_D).
\]

- **Step 2**: The best feature set \( X_d \) is then defined by the first \( d \) measurements \( y_j \), that is, \( X_d = \{y_j | j = 1, 2, \ldots, d\} \).

---

Unfortunately, the case of data sets naturally having diagonal covariance matrices seldom occurs in practice unless the independence of features is assumed. As a result of this assumption, this method of selecting features is the most unreliable but, nevertheless, the simplest one. Consequently, it should be used only if no alternative search method is feasible. However, it should be noted that this method will yield an optimal subset if the features are statistically independent and the criterion function satisfying equations (3.3) or (3.4) is used.
3.3.2 Sequential Forward Selection

The sequential forward selection (SFS) algorithm is a simple bottom up search procedure where one feature at a time is added to the current feature set. The algorithm starts from an empty set, and the individually best measurement is selected as the first feature. At each subsequent stage the candidate feature to be included in the set is selected from among the remaining available measurements, so that in combination with the features already selected it yields the best value of the criterion function.

The Sequential Forward Selection Algorithm

Suppose $k$ features have already been selected from the complete set of measurements $Y = \{y_j| j = 1, 2, \ldots, D\}$ to form feature set $X_k$. The $(k + 1)^{\text{st}}$ feature is then chosen from the set of available measurements, $Y - X_k$, so that

$$J(X_{k+1}) = \max_{y_j \in Y - X_k} J(X_k \cup y_j), \quad y_j \in Y - X_k$$

(3.5)

Initialization: $X_0 \equiv \emptyset$

It is known that the best pair of features does not necessarily contain the individual best feature selected in the first step of the algorithm [7]. As a result, the main source of suboptimality of the SFS method is that once a feature is included in the selected feature set, there is no mechanism for removing it from the feature set even if at a later stage, when more features have been added, this feature becomes superfluous.

As pointed out by Kittler [18], another drawback of the SFS method is that although it takes into account the statistical dependence between candidate features and those already selected, the number of candidate features to be added at each step of the algorithm is restricted to one. Hence, due to this restriction it is impossible to take into consideration statistical dependence between elements of the set of available measurements.
3.3.3 Sequential Backward Selection

The SFS algorithm has a counterpart called the sequential backward selection (SBS) algorithm which is a top-down process. Starting from the complete set of available measurements $Y$, one measurement at a time is discarded until $D-d$ measurements have been deleted. At each stage the measurement to be discarded from the current feature set is selected so that the newly reduced set of features gives the maximum possible value of the criterion function.

The Sequential Backward Selection Algorithm

Suppose $k$ features have already been removed from the complete set of measurements, $Y = \{y_j| j = 1, 2, \cdots, D\}$, to form feature set $X_{D-k}$. The $(k+1)^{st}$ feature to be eliminated is then chosen from the set $X_{D-k}$ so that

\[
J(X_{D-k-1}) = \max_{y_j} J(X_{D-k} - y_j), \quad y_j \in X_{D-k}
\]

Initialization : $X_D = Y$

The drawbacks of the SBS method are analogous to those of the SFS method. However, the main difference between these two methods is that the SBS procedure provides as a by-product a measure of maximum achievable class separability with the given set of features which can be used to assess the amount of information loss in the feature selection process. As far as the computational complexity is concerned, the SFS method is simpler than the SBS method since it requires that the criterion function be evaluated at most in $d$-dimensional spaces. In contrast, in the SBS method the criterion function must be computed in spaces of dimensionality ranging from $D$ down to $d$. 
3.3.4 Generalized Sequential Forward and Generalized Sequential Backward Selection

The statistical dependence of features to be added to or discarded from the current feature set, depending on the type of search procedure employed, can be taken into consideration by adding to or subtracting from the current feature set more than one measurements at a time. The following methods have been designed by Kittler [18] to accommodate this approach.

The generalized sequential forward selection GSFS(\(r\)) algorithm is similar to the basic SFS algorithm but with \(r\) measurements being considered at any one time instead of just one measurement. Starting from an empty set, we perform exhaustive search to find the best \(r\) measurements from the complete set of available measurements \(Y\). At each subsequent stage the next best \(r\) features are picked from the remaining available measurements so that in combination with the features already selected they yield the best value of the criterion function.

The Generalized Sequential Forward Selection Algorithm

Suppose \(k\) features have already been selected to form feature set \(X_k\). Generate all the possible sets of size \(r\), \(X_r\), from the set of available measurements \(Y - X_k\). Then select the next \(r\) features so that in combination with \(X_k\), the overall criterion function is maximum, i.e.

\[
J(X_{k+1}) = \max_{X_r} J(X_k \cup X_r), \quad X_r \in Y - X_k
\]  

(3.7)

Initialization : \(X_0 = \emptyset\)

The generalized sequential backward selection GSBS(\(r\)) algorithm is, by analogy, essentially the same as the SBS algorithm with the exception that more than one features are discarded at a time. Starting from the complete set of available measurements \(Y\), the worst \(r\) measurements are discarded. At each subsequent stage the worst \(r\) measurements from the remaining available set of measurements are deleted until \(d\) features are left in the remaining set.
The Generalized Sequential Backward Selection Algorithm

Suppose \( k \) features have been discarded from the set of available measurements \( Y \) to form feature set \( X_k \). Now form all the possible sets \( \tilde{X}_{k+r} \) by removing various combinations of \( r \) measurements from \( X_k \). Then select \( \tilde{X}_{k+r} \), the candidate feature set \( \tilde{X}_{k+r} \) that maximizes the criterion function, i.e.

\[
J(\tilde{X}_{k+r}) = \max_{\forall \tilde{X}_{k+r}} J(\tilde{X}_{k+r}), \quad \tilde{X}_{k+r} \in X_{D-k}
\]  

(3.8)

Initialization : \( \tilde{X}_0 \equiv Y \)

A few comments are in order at this point. Although the GSFS\((r)\) algorithm is more reliable than the basic SFS method, it is also more costly in computational terms; for at each stage \( (D-k) \) feature sets, \( X_k \cup \chi_r \), have to be inspected to find the feature set \( X_{k+r} \). Similarly, the GSBS\((r)\) algorithm requires a substantial increase in computation since at each stage \( (D-k) \) sets must be evaluated in comparison with the \( D-k \) candidate sets in the case of the SBS algorithm. However, the GSBS\((r)\) algorithm takes into consideration not only statistical dependence among features in the current feature set, \( X_k \), but also the relationship between the discarded measurements. Note that both the GSFS\((r)\) and GSBS\((r)\) algorithms still suffer from the nesting of successive feature sets.
3.3.5 Plus $l$-Minus $r$ Algorithm

The nesting of feature sets, which may rapidly result in suboptimality of both the SFS and SBS algorithms, can be partially overcome by alternating the process of augmentation and depletion of the feature set. This process can be viewed as the search method using the dynamic principle of optimization. After adding $l$ measurements to the current feature set, $r$ features are removed. Thus the net change in the size of feature set is equivalent to $l-r$, hence the name “Plus $l$-Minus $r$ ($l, r$)” selection algorithm. This process is continued until the feature set reaches the required size. There are many different ways to achieve the net change in size of the current feature set by $l-r$ features, however the easiest is to apply the basic SFS and SBS algorithms alternately.

The Plus $l$-Minus $r$ Algorithm

Let $X_k$ be the current feature set and if $l > r$ then

- **Step 1**: Apply SFS $l$ times to generate feature set $X_{k+l}$
- **Step 2**: Apply SBS $r$ times to obtain feature set $X_{k+l-r}$
- **Step 3**: Stop if $k + l - r = d$ else return to Step 1

The procedure for $l < r$ is the same as above with Step 1 and Step 2 interchanged. Note that the ($l, r$) algorithm with $r = 0$ is the SFS algorithm and with $l = 0$ it is the SBS algorithm.

Although the nesting in the ($l, r$) algorithm is avoided, it should be noted that eliminating nesting does not eliminate the Cover paradox [7]. Also, the ($l, r$) algorithm still suffers from other drawbacks of the SFS and SBS algorithms namely that groups of features are added and removed from the current feature set irrespective of their mutual relationship, i.e. only one candidate feature is being considered at a time. Furthermore, it is in principle impossible to order features according to their significance since the features in subsets of cardinalities differing by one, e.g. $X_k$ and $X_{k+1}$ or $X_k$ and $X_{k-1}$, may differ by more than one features. Finally, another drawback of the ($l, r$) algorithm is that no theoretical indication for what values of $l$ and $r$ will yield the best result, consequently multiple runs are usually needed.
3.3.6 Generalized "Plus $l$-Minus $r$" Algorithm

Another approach to achieve the net change in size of the current feature set set by $l-r$ features is to employ the GSFS($l$) and GSBS($r$) algorithms in alternation. After adding the best $l$ features with respect to the already selected features, the worst $r$ features are removed from the newly enlarged feature set. This process is continued until the required size of features is reached.

The Generalized "Plus $l$-Minus $r$" Algorithm

Suppose $k$ features have been selected to form set $X_k$ and if $l > r$ then

- **Step 1:** Enlarge $X_k$ by applying GSFS($l$) to generate feature set $X_{k+l}$
- **Step 2:** Reduce $X_{k+l}$ by applying GSBS($r$) to obtain feature set $X_{k+l-r}$
- **Step 3:** Stop if $k + l - r = d$ else return to Step 1

Initialization of the generalized $(l,r)$ algorithm is identical to that of the $(l,r)$ algorithm discussed earlier. Note that the procedure for $l < r$ is the same as above with Step 1 and Step 2 interchanged.

This greater sophistication of the feature set selection procedure is achieved once again at the expense of extra computations which is, in turn, preventing $l$ and $r$ from being too large. However, it is possible to curb computational complexity of the algorithm if the GSFS($l$) and GSBS($r$) algorithms are split into a number of substeps (see [18]).
3.3.7 Sequential Forward Floating and Sequential Backward Floating Selection

So far, the simple way to avoid nesting of feature sets is to employ either the \((l, r)\) or generalized \((l, r)\) algorithm which involves successive augmentation and depletion process. Consequently, the resulting dimensionality in respective stages of both algorithms is fixed depending on the prespecified values of \(l\) and \(r\). Unfortunately, there is no theoretical way of predicting the values of \(l\) and \(r\) to achieve the best feature set. Alternatively, instead of fixing these values, there is a couple of more sophisticated procedures of which these values are flexibly changing so as to approximate the optimal solution as much as possible. Although both of these methods switch between including and excluding features, they are recognized as two different algorithms according to the dominant direction of the search. The search in the forward direction is known as the *sequential forward floating selection* (SFFS), while in the opposite direction is known as the *sequential backward floating selection* (SBFS) [3], [28]. They are known as floating methods because the resulting dimensionality in respective stages of the algorithm is not changing monotonously but is actually “floating” up and down.

The SFFS is basically a bottom up search procedure which includes new features by means of applying the basic SFS procedure to the current feature set, followed by a series of successive conditional exclusion of the worst feature in the newly updated set if a further improvement can be made to the previous sets.

**The Sequential Forward Floating Selection Algorithm**

Suppose \(k\) features have already been selected from the complete set of measurements \(Y = \{y_j | j = 1, 2, \cdots, D\}\) to form set \(X_k\), with the corresponding criterion function \(J(X_k)\). In addition, the values of \(J(X_i)\) for all preceding subsets of size \(i = 1, 2, \cdots, k - 1\), are known and stored.

- **Step 1:** (Inclusion). Using the basic SFS method, select feature \(x_{k+1}\) from the set of available measurements, \(Y - X_k\), to form feature set \(X_{k+1}\), i.e. the most significant feature \(x_{k+1}\) with respect to the set \(X_k\) is added to \(X_k\).
• **Step 2**: (Conditional Exclusion). Find the least significant feature in the set $X_{k+1}$, i.e.

\[
J(X_{k+1} - x) = \max_{x_i} J(X_{k+1} - x_i), \quad x_i \in X_{k+1}
\]  

(3.9)

If $x_{k+1}$ is the least significant feature in the set $X_{k+1}$, i.e.

\[
J(X_{k+1} - x_{k+1}) \geq J(X_{k+1} - x_j), \quad \forall x_j \in X_k
\]

then set $k = k+1$ and return to Step 1, but if $x_r, 1 \leq r \leq k$, is the least significant feature in the set $X_{k+1}$, i.e.

\[
J(X_{k+1} - x_r) > J(X_k)
\]

then exclude $x_r$ from $X_{k+1}$ to form a new feature set $X'_k$, i.e.

\[
X'_k = X_{k+1} - x_r.
\]

Note that now $J(X'_k) > J(X_k)$. If $k = 2$, set $X_k = X'_k$ and $J(X_k) = J(X'_k)$ then return to Step 1 else go to Step 3.

• **Step 3**: (Continuation of conditional exclusion). Find the least significant feature $x_r$ in the set $X'_k$. If $J(X'_k - x_r) > J(X'_{k-1})$ then exclude $x_r$ from $X'_k$ to form a newly reduced set $X'_{k-1}$, i.e.

\[
X'_{k-1} = X'_k - x_r.
\]

Set $k = k-1$. Now if $k = 2$ then set $X_k = X'_k$ and $J(X_k) = J(X'_k)$ then return to Step 1 else repeat Step 3.

Initially, the procedure starts from an empty set, $X_0 \equiv \emptyset$, and the first two features are selected by the SFS method.

---

The SBFS is a top down search procedure which excludes features by means of applying the basic SBS procedure to the current feature set and followed by a series of successive conditional inclusions of the most significant feature from the available features if an improvement can be made to the previous sets.

---

**The Sequential Backward Floating Selection Algorithm**

Suppose $k$ features have already been removed from the complete set of measurements $X_0 = Y$ to form feature set $X_k$ with the corresponding criterion function $J(X_k)$. Furthermore, the values of all supersets $X_i, i = 1, 2, \ldots, k - 1$, are known and stored.

• **Step 1**: (Exclusion). Use the basic SBS method to remove feature $x_{k+1}$ from the current set $X_k$ to form a reduced feature set $X_{k+1}$, i.e. the least significant feature $x_{k+1}$ is deleted from the set $X_k$. 
Suboptimal Search Procedures

- **Step 2**: (Conditional inclusion). Find among the excluded features the most significant feature with respect to the set \( \bar{X}_{k+1} \), i.e.

\[
J(\bar{X}_{k+1} + x_j) = \max_{x_j} J(\bar{X}_{k+1} + x_j), \quad x_j \in Y - \bar{X}_{k+1} \tag{3.10}
\]

If \( x_{k+1} \) is the most significant feature with respect to \( \bar{X}_{k+1} \), i.e.

\[
J(\bar{X}_{k+1} + x_{k+1}) \geq J(\bar{X}_{k+1} + x_j), \quad \forall x_j \in Y - \bar{X}_{k+1}
\]

then set \( k = k+1 \) and return to Step 1. If \( x_r, 1 \leq r \leq k \), is the most significant feature with respect to the set \( \bar{X}_{k+1} \), i.e.

\[
J(\bar{X}_{k+1} + x_r) > J(\bar{X}_k)
\]

then include \( x_r \) to the set \( \bar{X}_{k+1} \) to form a new feature set \( \bar{X}_k' \), i.e.

\[
\bar{X}_k' = \bar{X}_{k+1} = x_r.
\]

Note that now \( J(\bar{X}_k') > J(\bar{X}_k) \). If \( k = 2 \), set \( \bar{X}_k = \bar{X}_1' \) and \( J(\bar{X}_k) = J(\bar{X}_k') \) then return to Step 1 else go to Step 3.

- **Step 3**: (Continuation of conditional inclusion). Find among the excluded features the most significant feature \( x_s \) with respect to the set \( \bar{X}_{k-1}' \). If \( J(\bar{X}_k' + x_s) > J(\bar{X}_{k-1}) \) then include \( x_s \) to the set \( \bar{X}_k' \) to form the new enlarged set \( \bar{X}_{k-1}' \), i.e.

\[
\bar{X}_{k-1}' = \bar{X}_k' + x_s.
\]

Set \( k = k-1 \). Now if \( k = 2 \) then set \( \bar{X}_k = \bar{X}_1' \) and \( J(\bar{X}_k) = J(\bar{X}_k') \) and return to Step 1 else repeat Step 3.

The procedure begins by deleting the first two features using the basic SBS method.

Unlike the \((l, r)\) and generalized \((l, r)\) algorithms in which factors such as the net change in the size of the current feature set, and especially the amount of computational time, are governed by the values of \( l \) and \( r \), the SFFS and SBFS methods are not restricted to these factors. This is because both methods are freely allowed to correct wrong decisions made in the previous steps so as to approximate the optimal solution as much as possible. As a result, there is no net change in the size of the current feature set in both methods. Moreover, the computational time depends mainly on the number of corrections made in the previous sets. Therefore, it is possible that a lot of computational time has to be spent before the required feature set is obtained. This problem is accentuated in the case of data of greater complexity and dimensionality.
3.3.8 Max-Min and Min-Min Search Methods

As far as the whole feature set $X_k = \{x_1, x_2, \cdots, x_k\}$ is concerned, the additional discriminatory information provided by a measurement $y_j$ to the set $X_k$ is expressed as

$$\Delta J(y_j, X_k) = J(y_j, X_k) - J(X_k)$$  \hspace{1cm} (3.11)

which involves calculation in at least $k+1$-dimensional space. However, in [2] Backer and De Shipper suggest that the amount of $\Delta J(y_j, X_k)$ in equation (3.11) can be determined by the minimum of the incremental information $J(y_j, x_i) - J(x_i)$ of measurement $y_j$ with respect to feature $x_i$, i.e.

$$\Delta J(y_j, X_k) = \min_{x_i} \Delta J(y_j, x_i), \quad x_i \in X_k$$  \hspace{1cm} (3.12)

where $\Delta J(y_j, x_i) = J(y_j, x_i) - J(x_i)$.

This relation simply implies that measurement $y_j$ which is similar or equivalent to some feature $x_i$ and therefore the value of $\Delta J(y_j, x_i)$ is small or zero, will not add any new information to the feature set $X_k$ even if in comparison with other features in the set it appears to convey a lot of relevant information.

When the criterion function in equation (3.12) is used in conjunction with the sequential forward selection, the search is known as the Max-Min algorithm. By analogy, it is essentially the same as the SFS procedure with the exception that measurement $y_j$ is selected as the $(k+1)^{st}$ feature if it satisfies

$$\Delta J(x_{k+1}, X_k) = \max_{y_j} \min_{x_i} \Delta J(y_j, x_i), \quad x_i \in X_k, \ y_j \in Y - X_k$$  \hspace{1cm} (3.13)

The Max-Min Algorithm

- **Step 1**: Starting from an empty set, the best pair of features is picked from the available measurements $Y$ using the GSFS(2) method.
- **Step 2**: For each $y_j \in Y - X_k$, the minimum of $\Delta J(y_j, x_i)$ for all $x_i \in X_k$ is stored.
- **Step 3**: The candidate $y_j$ corresponding to the maximum of the stored values in Step 2 is selected as the next feature.
- **Step 3**: Repeat Step 2 until $d$ features have been selected.
Similarly, the criterion function in equation (3.12) can also be used in conjunction with the SBS, and the search is called the Min-Min algorithm. In this algorithm the \((k + 1)\)th feature to be eliminated is chosen from the set \(X_{D-k}\) so that

\[
\Delta J(x_{k+1}, X_{D-(k+1)}) = \min_{y_j} \min_{x_i \neq y_j} \Delta J(y_j, x_i), \quad x_i, y_j \in X_{D-k}
\]  

(3.14)

The Min-Min Algorithm

- **Step 1**: The worst pair of features is deleted from a full set of measurements using the GSBS(2) method.
- **Step 2**: For each feature \(y_j \in X_{D-k}\), the minimum of \(\Delta J(y_j, x_i)\) where \(x_i \in X_{D-k}\) and \(x_i \neq y_j\) is stored.
- **Step 3**: The feature \(y_j\) that corresponds to the minimum of the stored values in Step 2 is deleted.
- **Step 4**: Repeat Step 1 until \(D - d\) features have been deleted.

Both the Max-Min and Min-Min algorithms require very little computational time since the evaluation of the criterion function is done in one- and two-dimensional spaces only. However, their performance, in general, is very poor. This is because the bound in equation (3.12), which is very loose, is used to evaluate the effectiveness of a candidate feature with respect to the already selected feature set. In addition, this bound cannot distinguish certain types of statistical conditions between the selected features and the candidate feature which may be useful in the search process [5], [19].
3.4 Experimental Results and Discussion

The described search methods have been evaluated by experiments on various types of data. The results confirm that the relative performance of the various methods is similar in all cases [5]. It is apparent that in order to clearly demonstrate the effectiveness of each method, the selection of a data set showing high statistical dependencies among features is preferential. As a result, the emphasis is now put on of the experiment based on the data used in [18] with strong interactions among features. This data consists of two normally distributed classes in a 20-dimensional space with means $\mu_i, i = 1, 2$ and an equal covariance matrix $\Sigma$. Consequently, it is pertinent to use the Mahalanobis distance $J_M$ as a criterion of feature set effectiveness. Unfortunately, the comparison of the effectiveness of all described search methods together is rather difficult, therefore it will be done separately in smaller groups. Note that all the experiments were performed on SUN SPARC station 1.

Exhaustive and Branch-and-Bound Search Methods

The results of the exhaustive search and the branch-and-bound methods are shown in Figure 3.2. From Figure 3.2(a) it can be clearly seen that the optimal Mahalanobis distance obtained from the exhaustive search increases monotonically as the size of feature subsets increases. This, in turn, explicitly shows that the Mahalanobis distance possesses the monotonicity property. Owing to the satisfaction of the monotonicity condition, the branch-and-bound similarly achieves the optimum solution but with much less computational time than that of the exhaustive search as shown in Figure 3.2(b). This is due to the fact that only a fraction of all the possible candidate feature sets was enumerated to find the optimal set of features. However, apart from having to satisfy the monotonicity condition, another drawback of this method is that, in the case of high dimensionality of the feature space, the number of nodes visited may become very large especially when $d$ is approaching $D/2$. Consequently, the computational time may be too large or prohibitive. Figure 3.2(b) also shows that it is advantageous to use the exhaustive search when $d$ is small or approaching $D$ since it requires less computational time than the branch-and-bound method due to smaller number of inspected sets.

Sequential Forward Search Methods

Figure 3.3 shows the results of various sequential forward search methods. The main source of suboptimality of these methods is that once a feature is included in the selected
feature set in an early stage, there is no mechanism for removing it from the already selected feature set even if at later stage, this feature becomes irrelevant or redundant. Figure 3.3(a) shows that, as expected, the performance of the GSFS(r) is better than that of the basic SFS. This is because in the GSFS(r) method, the statistical dependence between more than one candidate features and those already selected is taken into account instead of just one candidate feature as in the SFS method. Unfortunately, the better performance is obtained at the expense of a considerable increase in computational time. As far as the computational time is concerned, it is the Max-Min that requires the least as shown in Figure 3.3(b), however its performance is the worst due to the limited information used in the search process and other drawbacks mentioned earlier.

**Sequential Backward Search Methods**

The results of various sequential backward search methods are shown in Figure 3.4. Since these methods are top down search procedures, it is conventional to look at their results from a high dimensional feature set downwards. The cause of suboptimality of these methods are analogous to that of the sequential forward search methods discussed above. Generally, the performance of all the methods shown in Figure 3.4(a) is very similar except that of the Min-Min which is rapidly degrading due to limited information used in the search process. Furthermore, Figure 3.4(a) also clearly shows that all of these methods suffer heavily from the nesting effect when the size of feature set $d$ is less than 9, except the Min-Min which badly suffers at the very early stage. However, when $d > 9$ the performances of the SBS and GSBS(r) methods are surprisingly close to and sometimes even optimal. As expected, the more sophisticated the search method becomes the longer the computational time required (see Figure 3.4(b)) but not always the better the performance (see Figure 3.4(a)). Although the computational time required to find a feature set by any top down search procedure is usually longer than that of bottom up counterpart, this is compensated by the ability to assess the information loss due to feature selection.

**Plus l-Minus r Method**

Figure 3.5 shows the results of the Plus l-Minus r search method $(l, r)$ when $l > r$ (bottom up search). The performance of the $(l, r)$ method with various combinations of $l$ and $r$ is relatively close to the optimal solution as shown in Figure 3.5 (a) This is because the $(l, r)$ algorithm has a mechanism to allow some irrelevant or redundant features to be removed after they had been selected, i.e. no nesting. Nevertheless, as mentioned
before, eliminating nesting does not guarantee optimal solution. As far as the values of \( l \) and \( r \) are concerned, increasing both of them always increases computational time but does not necessarily mean that better feature subsets will be obtained. For example, when the size of the selected feature subset \( d \) is 8, the performance of \((l, r)\) when \( l = 3 \) and \( r = 2 \) is actually better than when \( l = 4 \) and \( r = 3 \). Additionally, it should be noted that the computational time required to find any feature subset can be very small if the values of \( l \) and \( r \) are chosen to be largely different. However, this may cause the algorithm to be incapable of preventing nesting properly. Hence the degraded performance such as the case when \( d = 4, l = 4 \) and \( r = 2 \).

The drawbacks of the Plus \( l \)-Minus \( r \) \((l, r)\) search method when \( l < r \) are analogous to those of \((l, r)\) when \( l > r \). Furthermore, the performance of the Plus \( l \)-Minus \( r \) search method when \( l < r \) is similar to that when \( l > r \) as shown in Figure 3.6 (a). However, since it is a top down process which starts computation from the full feature set \( D \) downwards, it requires a much longer time than the \((l, r)\) method with \( l > r \), particularly when \( d \) is close to \( D/2 \).

**Generalized "Plus \( l \)-Minus \( r \)" Method**

The drawbacks of the generalized Plus \( l \)-Minus \( r \) search method are similar to those of the \((l, r)\) with the exception that more than one candidate feature is considered at a time. From Figure 3.7(a) and 3.8(a) the generalized \((l, r)\) search method gives optimal feature sets almost everywhere. However, the computational time shown in Figure 3.7(b) and 3.8(b) is much higher than that of the basic \((l, r)\) algorithm and sometimes even as long as that of the branch-and-bound. Consequently, high computational time requirements prevent \( l \) and \( r \) from being too large.

**Sequential Forward Floating and Sequential Backward Floating Methods**

Although both the SFFS and SBFS methods can partially overcome nesting, they are different from \((l, r)\) and even generalized \((l, r)\) methods. This is because at each step of either the SFFS or SBFS method the number of features being included and excluded varies, though the inclusion or exclusion is only conditional. In other words, the values of \( l \) and \( r \) are not fixed but are flexibly changing so as to approximate the optimal solution as much as possible. Consequently, both the SFFS and SBFS methods give, as expected, optimal feature sets almost everywhere as shown in Figure 3.9(a). Figure 3.9(b) indicates that the computational time required by either of these methods compares very favourably with the branch-and-bound and the generalized \((l, r)\) algorithms.
Optimal Search Methods

20-dimensional data

(a)

(b)

Figure 3.2: Results of the optimal search methods
Sequential Forward Search Methods

20-dimensional data

(a)

(b)

Figure 3.3: Results of the sequential forward search methods
Sequential Backward Search Methods

20-dimensional data

![Graph showing results of sequential backward search methods](image)

(a)

Figure 3.4: Results of the sequential backward search methods
Chapter 3: A Review of Feature Set Search Procedures

Plus l-Minus r method (l>r)

20-dimensional data

(a)

Figure 3.5: Results of the Plus l-Minus r search method when l > r
Figure 3.6: Results of the Plus l-Minus r search method when $l < r$
Chapter 3: A Review of Feature Set Search Procedures

Generalized Plus l-Take Away r Method (l>r)

20-dimensional data

![Graph showing Mahalanobis distance and time vs. number of selected features/set](image)

(a)

![Graph showing Mahalanobis distance and time vs. number of selected features/set](image)

(b)

Figure 3.7: Results of the generalized Plus l-Minus r search method when l > r
Experimental Results and Discussion

Generalized Plus l-Take Away r method (l<r)

20-dimensional data

(a) 

Figure 3.8: Results of the generalized Plus l-Minus r search method when l < r
Chapter 3: A Review of Feature Set Search Procedures

Floating Search Methods

20-dimensional data

Mahalanobis distance $J_M$

(a)

(b)

Figure 3.9: Results of the floating methods
3.5 Conclusion

The major developments over the past three decades in the area of feature selection, as far as search strategies are concerned, have been reviewed in this chapter. The advantages and disadvantages of various methods have been compared both theoretically and experimentally. It can be seen that the feature set yielded by a more sophisticated method, in general, is likely to be better than the one that is obtained by using the less sophisticated method, but this cannot always be guaranteed as the experimental results have shown. Nevertheless, the likelihood of obtaining a better feature set when more complicated search schemes are employed is higher, but at the expense of an increase in computational time.

It must be emphasized that the objective of this chapter has been to analyse the performance of different feature selection methods. By taking into the account the advantages and disadvantages of these various methods, it is hoped to develop a new feature selection algorithm suitable for high dimensional feature selection problems. Out of these methods, it is noticed that the computational time required by the Max-Min and the Min-Min algorithms is very little and hardly increases even though the size of the required feature set increases. Unfortunately, both methods give rather disappointing results. However, it is obvious that these methods will be very favourable with the high dimensionality problems, which is the aim of our research, provided that an improvement on results can be made. Therefore, our attention turns towards the modification of these two methods.

Since both the Max-Min and Min-Min algorithms are based on the same principle which is the evaluation of the feature set effectiveness in 2-dimensional space, only the Max-Min algorithm will be studied in more detail with the aim of improving its performance. Then any modification made to the Max-Min algorithm can be similarly applied to the Min-Min algorithm. Note that the recent search strategies have not been implemented and tested because their criterion function needs to be evaluated in higher dimensional spaces than that of the Max-Min algorithm. As a result, they are unlikely to use less computational time than the Max-Min algorithm. Moreover, the optimality of the selected feature set from these recent search strategies cannot be guaranteed.

In order to improve the performance of the Max-Min algorithm, its weaknesses have to be identified and subsequently overcome. These requirements will be discussed in the following chapter.
References


References


Chapter 4

The Max-Min Algorithm

4.1 Introduction

The Max-Min feature selection method is a sequential method proposed by Backer and De Schipper [1]. The method has a computational advantage over the other well known methods mentioned in Chapter 3. This advantage lies in the fact that instead of computationally time consuming calculations in a multidimensional space, typical for the other methods, the Max-Min method requires all the calculations in two-dimensional space only. Thus, it is very attractive for feature selection problems in the case when original dimensionality is quite high.

Owing to this really appealing characteristics, it has been included into a number of monographs on pattern recognition or feature selection particularly, e.g. Hand [5], Kittler [7]. However, the results achieved with this method are invariably rather unsatisfactory. The results of comparative study of various feature selection algorithms reported in Chapter 3 indicates that the Max-Min method gives the poorest performance reflecting the limited information used in the search process. The result evidently confirms that it is not possible to select a set of features in a high-dimensional space based on two-dimensional information measures without a substantial information loss [4]. In addition to this deficiency, there are other detrimental factors which are inherent to the Max-Min method itself. They will be presented after the discussion of the original algorithm in the next section. Finally, the question whether it is possible to improve the performance of the Max-Min method is answered.
4.2 The Max-Min Method

As known the problem of feature selection lies in selecting the best subset $X_d$ of the features, $X_d = \{x_i \mid i = 1, 2, \ldots, d; x_i \in Y\}$ from the set $Y$, $Y = \{y_j \mid j = 1, 2, \ldots, D\}$, of $D$ original measurements representing the pattern, $D > d$. By the best subset we understand the combination of $d$ features which optimizes a feature evaluation criterion $J$ with respect to any other combination of $d$ measurements taken from $Y$.

Let us denote

- $X_k$: feature from the selected feature set $X_k = \{x_1, x_2, \ldots, x_k\}$ acquired in the $i^{th}$ step of the selection procedure;
- $Y_j$: $j^{th}$ feature from the set $Y - X_k$ of candidate features;
- $\Delta J(y_j, x_i)$: the absolute value of the difference between $J(y_j, x_i)$ and $J(x_i)$.

In the Max-Min method such a candidate feature $y_j$ is chosen as the $(k + 1)^{st}$ feature $x_{k+1}$ which yields

$$\max_{x_i \in X_k, y_j \in Y - X_k} \min \Delta J(y_j, x_i).$$

(4.1)

The derivation of equation (4.1) is based on the following. In accordance with the original Backer's paper, let us adopt the probability of error $P_e$ as the feature evaluation criterion.

Then, from among the set $Y$ of candidate features, the feature that gives the smallest $P_e$ is selected as the feature $x_1$, i.e.

$$P_e(x_1) = \min_{y_j} P_e(y_j).$$

(4.2)

The second feature $x_2$ is selected from the candidates $y_j$ such that

$$\Delta P_e(x_2, x_1) = \max_{y_j} \Delta P_e(y_j, x_1),$$

(4.3)

where $\Delta P_e(y_j, x_1) := P_e(x_1) - P_e(y_j, x_1)$ for any $x_i \in X_k, y_j \in Y - X_k$.

If the probability of error is used as the feature evaluation criterion, the value of $\Delta P_e(y_j, x_1)$ can serve as a measure of additional discriminatory information when the candidate $y_j$ is added to the already selected feature $x_1$. In the original paper of Backer and De Schipper [1] it is assumed, or rather stated without a proof, that if a new feature is added to the first two already selected features, the decrease of $P_e$ is always less than or equal to the decrease of $P_e$ obtained by fusing that feature with one of the already selected features, i.e.
Heuristic Reasoning Behind the Max-Min Method and Its Drawbacks

\[ \Delta P_e(y_j, (x_1, x_2)) \leq \Delta P_e(y_j, x_1), \quad \] (4.4)
\[ \Delta P_e(y_j, (x_1, x_2)) \leq \Delta P_e(y_j, x_2). \quad \] (4.5)

Extension of inequalities (4.4) and (4.5) leads to

\[ \Delta P_e(y_j, X_k) \leq \min_{x_l \in X_k} \Delta P_e(y_j, x_l), \quad x_l \in X_k. \quad \] (4.6)

The maximization of inequality (4.6) over all the candidate features forms equation (4.1) and gives the \((k+1)\)st selected feature. Inequalities (4.4) and (4.5), together with their extension (4.6), constitute a basic premise from which the whole Max-Min method is developed. In the following the original algorithm is discussed from the viewpoint of its heuristic reasoning and potential drawbacks are outlined.

4.3 Heuristic Reasoning Behind the Max-Min Method and Its Drawbacks

Since the Max-Min method is a sequential method, the cause of its deficiency is the error propagating from the selection of wrong features in the initial stages. The reasons for selecting wrong features can be elicited as follows.

Firstly, in the Max-Min procedure the amount of additional discriminatory information \(I_{y_j}\), provided by each candidate feature \(y_j\), is according to relation (4.6) bounded by the minimum of \(\Delta P_e(y_j, x_l)\) for any \(x_l \in X_k\), i.e.

\[ I_{y_j} \leq \min_{x_l \in X_k} \Delta P_e(y_j, x_l), \quad x_l \in X_k. \quad \] (4.7)

However, in the Max-Min method the upper limit on the error bound is used instead, i.e.

\[ I_{y_j} = \min_{x_l \in X_k} \Delta P_e(y_j, x_l), \quad x_l \in X_k. \quad \] (4.8)

The maximization of equation (4.8) over all the available features produces the next \((k+1)\)st feature. In this way the actual amount of additional information is implicitly identified with its upper bound, which will obviously not be true in general.

However, though not explicitly specified in the original paper of Backer and De Schipper [1], there is some heuristic reasoning hidden behind the use of the upper bound (equation (4.8)) as a measure of feature effectiveness. It lies in the conjecture that the higher the upper bound of \(I_{y_j}\), the higher the actual value of \(I_{y_j}\). Though this hope is far from justified, it is still better than picking up candidate features at random. Nevertheless, it
is important always to bear in mind that there is absolutely no guarantee that selecting a candidate feature with the highest upper bound will yield the highest actual amount of additional information. This may result in incorrect selection of new features which is one source of errors. We shall illustrate this problem more explicitly by the following example.

Suppose 2 features \( x_1 \) and \( x_2 \), have already been selected, and there are 3 available candidate features \( y_1 \), \( y_2 \) and \( y_3 \). Then from inequality (4.6) it follows that the actual values of \( I_{y_j} \) lie within the respective regions denoted \( R_{y_j} \), as illustrated in Figure 4.1. According to the Max-Min algorithm, clearly \( y_2 \) will be selected as the third feature since its upper bound is the maximum one. However, it is possible that the actual value of \( I_{y_1} \) or \( I_{y_3} \) will be greater than \( I_{y_2} \), hence resulting in the selection of wrong features.

![Figure 4.1: An example of possible ranges and actual values of \( I_{y_j} \)'s.](image)

Another inherent weakness of the Max-Min method is that it does not distinguish between strong unconditional dependence and poor performance. The latter is due to independence of a new candidate feature which has little discriminatory power on its own (see [2] and [6]). Both of these cases manifest themselves in near zero \( \Delta f(y_j,x_i) \) values. The point is that the new candidate feature, though containing little information value on its own, can exhibit the amplifying dependence effect on some of the already selected features. The effect results in a far bigger value of joint information than the sum of single information values. However, as simple analysis can show, if among the currently selected features there exists a feature on which the candidate feature is independent, then it will not make much impact on the joint performance. This will result in small or
Can The Performance of The Max-Min Algorithm Be Improved?

The main problem with the Max-Min algorithm is that the criterion in equation (4.8) cannot distinguish between two important cases which both manifest themselves in near zero $\Delta J(y_j, x_i)$ values. More specifically, these two cases may be summarized as follows: when $y_j$ exhibits a strong unconditional dependence on one of the existing features in $X_k$; and when $y_j$ is an individually poor feature (of little discriminatory power) which is simultaneously unconditionally independent on one of the already selected features. Although this may seem counterintuitive, the candidate $y_j$ in the latter case can still be extremely valuable if it exhibits the amplifying dependence effect with one or more of the features in $X_k$. In order to differentiate between these two situations, it has been suggested that some measure of dependence of a candidate feature on the other already selected features must be evaluated and used as an aid to feature selection [2], [8].

4.4.1 A Measure of Dependence

A measure of dependence is defined as a numerical value that can be used to determine the statistical condition under which a feature pair is characterized. In [2] and [8], a useful measure of dependence $F_i$ between the already selected feature $x_i$ and the candidate feature $y_j$ is expressed as

$$F_i = \frac{J(x_i, y_j)}{J(x_i) + J(y_j)}.$$  (4.9)

Generally, any class separability measure $J$ depends actually on the difference of class conditional densities. For example the Kolmogorov variational distance for the already selected feature $x_i$, the candidate feature $y_j$ and their combination can be expressed as follows:
Chapter 4: The Max-Min Algorithm

\[ J(x_i) = \int_{-\infty}^{\infty} |[p(x_i|\omega_1) - p(x_i|\omega_2)]| dx_i, \quad (4.10) \]

\[ J(y_j) = \int_{-\infty}^{\infty} |[p(y_j|\omega_1) - p(y_j|\omega_2)]| dy_j, \quad (4.11) \]

\[ J(x_i, y_j) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |[p(x_i, y_j|\omega_1) - p(x_i, y_j|\omega_2)]| dx_i dy_j. \quad (4.12) \]

Therefore, the measure of dependence in equation (4.9) can be interpreted as the ratio between the integral of the absolute value of the difference of class conditional joint density functions and the sum of the integral of the absolute value of the differences of class conditional marginal density functions.

In the following, a number of examples of various statistical conditions with the corresponding values of the dependence factor are given as an aid to understanding these conditions of which some are rather complex.

Class Unconditional Stochastic Independence and Dependence

Two features \( x_1 \) and \( x_2 \) form an unconditionally stochastically independent pair if for their joint density the following relation holds:

\[ p(x_1, x_2) = p_1(x_1)p_2(x_2), \quad (4.13) \]

where \( p_1(x_1) \) and \( p_2(x_2) \) are, respectively, the marginal density functions for \( x_1 \) and \( x_2 \) which are drawn from the mixture population, i.e. mixture density functions. In the opposite case when

\[ p(x_1, x_2) \neq p_1(x_1)p_2(x_2) \quad (4.14) \]

that is

\[ p(x_1, x_2) = p_1(x_1)p_2(x_2|x_1), \quad p_2(x_2|x_1) \neq p_2(x_2), \quad (4.15) \]

the two features \( x_1 \) and \( x_2 \) are said to be an unconditionally stochastically dependent feature pair. Thus in this case a range of values of feature \( x_2 \) with the corresponding probabilities can be determined based upon the knowledge of \( p_2(x_2|x_1) \) (see Figure 4.2(a)). At the extreme of unconditional stochastic dependence we have the case of deterministic functional dependence, when \( x_2 \) is a function of \( x_1 \), i.e.

\[ x_2 = g(x_1). \quad (4.16) \]
In the case of linear function, this case can be expressed in a standard straight line equation, i.e.

\[ x_2 = ax_1 + b \]  \hspace{1cm} (4.17)

where \( a \) and \( b \) are constant, and \( a \neq 0 \). Consequently, the points corresponding to the samples \((x_1, x_2)\) form a straight line (see Figure 4.2(b)).

![Diagram](image)

Figure 4.2: Two examples of unconditional dependence cases (a) unconditional stochastic dependence (b) deterministic functional dependence.

Note that in the case of a strong unconditional stochastic dependence, there will be very little increase in the amount of additional discriminatory information when combining \( x_2 \) with \( x_1 \). Thus, in terms of the criterion function \( J \), this situation can be expressed
as

\[ J(x_1, x_2) < J(x_1) + J(x_2) \]

or

\[ \Delta J(x_2, x_1) < J(x_2), \]

\[ \implies F_1 < 1. \] \hspace{1cm} (4.18)

In the extreme case of deterministic functional dependence, \(x_2\) becomes redundant since it does not contribute any additional discriminatory information at all. In terms of the criterion function this can be expressed as

\[ J(x_1, x_2) = J(x_1) \]

or

\[ \Delta J(x_2, x_1) = 0, \]

\[ \implies F_1 = 1/2. \] \hspace{1cm} (4.19)

Class Conditional Stochastic Independence and Dependence

Two features \(x_1\) and \(x_2\) form a class conditionally stochastically independent pair if for their joint conditional density the following relation holds:

\[ p(x_1, x_2 | \omega_i) = p_1(x_1 | \omega_i)p_2(x_2 | \omega_i) \] \hspace{1cm} (4.20)

or

\[ p_2(x_2 | x_1, \omega_i) = p_2(x_2 | \omega_i), \ i = 1, 2. \] \hspace{1cm} (4.21)

It holds for certain class of criterion functions \(J\) and for a class conditionally stochastically independent pair \(x_1\) and \(x_2\) that

\[ J(x_1, x_2) = J(x_1) + J(x_2) \]

or

\[ \Delta J(x_2, x_1) = J(x_2), \]

\[ \implies F_1 = 1. \] \hspace{1cm} (4.22)

If the opposite of equations (4.20) and (4.21) holds, then the feature pair \(x_1\) and \(x_2\) is said to be subject to class conditional stochastic dependence, i.e.

\[ p(x_1, x_2 | \omega_i) \neq p_1(x_1 | \omega_i)p_2(x_2 | \omega_i) \] \hspace{1cm} (4.23)

or

\[ p_2(x_2 | x_1, \omega_i) \neq p_2(x_2 | \omega_i). \] \hspace{1cm} (4.24)
Even if $x_1$ and $x_2$ are class conditionally dependent, it is possible to have both class conditional joint distributions almost identical (see Figure 4.3). This means that the amplifying dependence effect does not follow automatically from class conditional dependence. However, a necessary condition though not a sufficient one for the existence of the amplifying dependence effect can be expressed as

$$p(x_1, x_2 | \omega_1) \neq p(x_1, x_2 | \omega_2)$$

and in terms of the criterion function, we have

$$J(x_1, x_2) \geq J(x_1) + J(x_2)$$

or

$$\Delta J(x_2, x_1) \geq J(x_2),$$

$$\Rightarrow F_1 \geq 1.$$

### 4.4.2 Amplifying Dependence

In this section the cause of this effect is investigated. From the examples in Figure 4.4, it is obvious that there is no difference between both cases in terms of unconditional dependence of features and class conditional dependence of features. Moreover, although the class conditional joint densities are substantially different in both cases, only Case 2 has the amplifying dependence effect. So, what is the fundamental difference between Case 1 and Case 2 causing the amplifying effect? The answer lies in the class conditional marginal densities.
unconditional dependence: yes  yes
class conditional dependence: yes yes
class conditional joint density functions different: yes yes
amplifying dependence effect: no yes

\[ J(x_1, x_2) \approx J(x_1) \quad J(x_1, x_2) \gg J(x_1) + J(x_2) \]
\[ \Delta J(x_2, x_1) \approx 0 \quad \Delta J(x_2, x_1) \gg J(x_2) \]
\[ F_1 \approx 1/2 \quad F_1 \gg 1 \]

Figure 4.4: Two contrasting examples of class conditional stochastic dependence cases.
Figures 4.5(a) and (b) show, respectively, the class conditional marginal densities of $x_1$ and $x_2$ from Case 1. These marginal densities can be expressed as follows:

\[
P_1(x_1|\omega_1) \neq P_1(x_1|\omega_2),
\]

\[
P_2(x_2|\omega_1) \neq P_2(x_2|\omega_2).
\]

Similarly, Figures 4.6(a) and (b) show, respectively, the class conditional marginal densities of $x_1$ and $x_2$ from Case 2. These marginal densities can be expressed as follows:

\[
P_1(x_1|\omega_1) \approx P_1(x_1|\omega_2),
\]

\[
P_2(x_2|\omega_1) \approx P_2(x_2|\omega_2).
\]

As far as the difference in class conditional joint densities of both cases is concerned, in Case 1 it is caused by the difference of class conditional marginal densities (no additional information obtained from fusion of features), whereas in Case 2 it is caused by the fusion of features (new quality from the fusion of features).
4.4.3 Modifying the Max-Min Procedure

Having defined the measure of dependence, the remaining problem is how it can be used in conjunction with the original Max-Min algorithm. Ideally, the performance of the new algorithm close to the SFS algorithm is aimed at. In [2] and [8], several approaches have been investigated with the aim to favour features which exhibit an amplifying dependence effect and on the other hand to suppress features which exhibit unconditional dependence, together with other conditions described earlier (generally, to suppress features which increase the value of the criterion function $J$ only very little).

These approaches can be summarized as follows:

Max-Max-Min Algorithm

In the original Max-Min procedure, a candidate feature $y_j$ is paired with one of the selected features $x_i$ and $\Delta J(y_j, x_i)$ is evaluated and used in the selection process. In contrast, in the following method, the value of $\Delta J(y_j, x_i)$ will be weighted by a factor which represents the dependence of $y_j$ on the other features in the set. The effect will hopefully cause the candidate feature $y_j$ not to be selected during the maximization process, if it is
Can The Performance of The Max-Min Algorithm Be Improved?

Can The Performance of The Max-Min Algorithm Be Improved?

The Max-Min Algorithm

Let $X_k$ be the current feature set. Select as the $(k+1)^{th}$ feature that measurement $y_j \in Y - X_k$ which satisfies

$$\Delta J(x_{k+1}, X_k) = \max_{y_j} \max_{x_i} \{ \Delta J(y_j, x_i) \ast \min_{k \neq i} F_k \}, \quad x_i \in X_k,$$

where $F_k$ is set to:

$$\frac{1}{J(y_j, x_k)} \quad \text{if } J(y_j, x_k) \geq J(x_k) + J(y_j);$$

$$\frac{J(y_j, x_k)}{J(x_k) + J(y_j)} \quad \text{if } J(y_j, x_k) < J(x_k) + J(y_j).$$

New Max-Min Algorithm

In this modification it is hoped that if the candidate $y_j$ is exhibiting the amplifying dependence effect with most of the already selected features, then the product of the dependence factors between the candidate $y_j$ and the other already selected features will scale up $\Delta J(y_j, x_i)$. Consequently, this will possibly cause $y_j$ to be selected in the maximization process. On the other hand, if the candidate $y_j$ is unconditionally dependent on most of the already selected features, then the product will scale $\Delta J(y_j, x_i)$ down and will possibly cause $y_j$ not to be selected in the maximization process. This algorithm is known as the new Max-Min algorithm.

The New Max-Min Algorithm

Let $X_k$ be the current feature set. Select as the $(k+1)^{th}$ feature that measurement $y_j \in Y - X_k$ which satisfies

$$\Delta J(x_{k+1}, X_k) = \max_{y_j} \min_{x_i} \{ \Delta J(y_j, x_i) \ast \prod_{k \neq i} \frac{J(y_j, x_k)}{J(x_k) + J(y_j)} \}, \quad x_i \in X_k.$$
4.4.4 Experimental Results

The experimental results obtained from both the Max-Max-Min and the new Max-Min algorithms are shown in Figure 4.7. Although some improvement on the performance of the Max-Min algorithm has been achieved, it still falls behind the SFS procedure. As far as the theoretical basis is concerned, the new Max-Min algorithm is much preferred to the Max-Max-Min algorithm. This is because if the extra term (factor) in both algorithms is set to 1, then the former will be reduced to the original Max-Min algorithm while the latter will be reduced to the Max-Max algorithm which is obviously in contradiction with the theory underlying the Max-Min algorithm.

4.4.5 Conclusion

Although various heuristic approaches like the minimum dependence factor, or a “voting” rule between the number of conditional and unconditional dependencies, or the geometric average of dependence factors have been tried, unfortunately only a slight improvement over the original Max-Min method has been achieved (for more details see [3]). Therefore, a better heuristic approach is needed so that the performance of the Max-Min method can be improved even further. Incidentally, during the search for the better
heuristic approach, our attention has been focused on the validity of the basic premise of the Max-Min method. The failure to prove the validity of the relation in equation (4.6) has lead to a suspicion that this relation might not be valid at all.

As stated earlier, though the optimal selection cannot be guaranteed, assuming the relation specified by equation (4.6) is valid, it could provide some heuristic reasoning for the Max-Min algorithm. However, unfortunately even the relation (4.6) is not necessarily valid as we shall demonstrate in the sequel. A counter-example will be given demonstrating that not only the actual increment of $I_{Y_j}$ can be bigger than the bound in inequality (4.6), i.e. $\min \Delta P_e(y_j, x_i) \forall x_i$, but it can also be even bigger than $\max \Delta P_e(y_j, x_i) \forall x_i$. Therefore, the heuristic reasoning behind the Max-Min algorithm is in the least very questionable and poor results obtained from the algorithm should not be too surprising. Conditions under which the basic inequality (4.6) governing the Max-Min algorithm is not valid are discussed in the next section and the results of a simple example are given.

4.5 Proof of Invalidity of The Key Premise of The Max-Min Algorithm

As already stated, the original paper of Backer and De Schipper [1] claims that adding a new feature $Y_j$ to the already selected feature set $X_k$ will not result in a bigger decrease of $P_e$ than when fusing that $Y_j$ with any of the features from the set $X_k$.

In this section we shall demonstrate that unfortunately this claim cannot be validated. Let us consider the case of selecting the third feature to the two already selected as an example. More exactly, let the current feature set of cardinality two be $X_2$ where $X_2 = \{x_1, x_2\}$ and $x_3$ be one of the candidate features. In the original Max-Min algorithm, it is assumed that

$$\Delta P_e(x_3, X_2) \leq \min_{\forall x_i} \Delta P_e(x_3, x_i), \quad x_i \in X_2. \quad (4.31)$$

In order to show that a counter-example for the Max-Min algorithm exists, we only need to find $x_3$ that satisfies

$$\Delta P_e(x_3, (x_1, x_2)) > \Delta P_e(x_3, x_1). \quad (4.32)$$

The inequality (4.32) can be rewritten as

$$P_e(x_1, x_2) - P_e(x_1, x_2, x_3) > P_e(x_1) - P_e(x_1, x_3). \quad (4.33)$$
Chapter 4: The Max-Min Algorithm

Since any counter-example to the inequality (4.31) will prove our claim, we can assume that the feature vector \(x\), representing the feature set \(X\) whose elements are the components of \(x\), is normally distributed. Then \(P_e(x)\) can be defined in terms of Mahalanobis distance \(J_M(x)\) as follows

\[
P_e(x) = \Phi \left( \frac{1}{2} \sqrt{J_M(x)} \right) = 1 - \Phi \left( \frac{1}{2} \sqrt{J_M(x)} \right),
\]

where \(\Phi(\cdot)\) is the standard normal distribution function which monotonically increases as \(J_M(\cdot)\) increases. This means that \(P_e(\cdot)\) decreases as \(J_M(\cdot)\) increases. Now, substituting equation (4.34) in the inequality (4.33) we obtain

\[
\Phi \left( \frac{1}{2} \sqrt{J_M(x_1, x_2, x_3)} \right) - \Phi \left( \frac{1}{2} \sqrt{J_M(x_1, x_2)} \right) > \Phi \left( \frac{1}{2} \sqrt{J_M(x_1, x_3)} \right) - \Phi \left( \frac{1}{2} \sqrt{J_M(x_1)} \right).
\]

Furthermore, it is assumed that feature \(x_2\) has been selected to form the best pair with feature \(x_1\), which means that \(J_M(x_1, x_2) \geq J_M(x_1, x_3)\). Hence, the following relation for Mahalanobis distance obviously holds

\[
J_M(x_1, x_2, x_3) \geq J_M(x_1, x_2) \geq J_M(x_1, x_3) \geq J_M(x_1) \geq 0.
\]

We shall now formulate a necessary condition for the inequality (4.33) to hold.

**Lemma:**

If the inequality (4.33) holds for \(P_e(x)\) defined by equation (4.34), then the following relation must be satisfied:

\[
\frac{1}{2} \sqrt{J_M(x_1, x_2, x_3)} - \frac{1}{2} \sqrt{J_M(x_1, x_2)} \geq \frac{1}{2} \sqrt{J_M(x_1, x_3)} - \frac{1}{2} \sqrt{J_M(x_1)}.
\]

In other words, the inequality (4.37) is a necessary condition for the inequality (4.33) to hold.

**Proof:**

For simplicity, let \(z_1, z_2, z_3, z_4\) denote \(\frac{1}{2} \sqrt{J_M(x_1)}, \frac{1}{2} \sqrt{J_M(x_1, x_2)}, \frac{1}{2} \sqrt{J_M(x_1, x_3)}, \) and \(\frac{1}{2} \sqrt{J_M(x_1, x_2, x_3)}\) respectively, \(z_i \in (0, \infty), i = 1, 2, \cdots, 4\). We can then rewrite the inequality (4.35) in the form
Proof of Invalidity of The Key Premise of The Max-Min Algorithm

\[ \Phi(z_4) - \Phi(z_3) > \Phi(z_2) - \Phi(z_1) \]  
(4.38)

and the inequality (4.37) in the form

\[ z_4 - z_3 \geq z_2 - z_1. \]  
(4.39)

Thus we need to prove that the relation (4.39) follows from the inequality (4.38). In order to prove this, let us assume for a moment that the relation (4.39) does not follow from the inequality (4.38) but on the contrary the following inequality holds

\[ z_4 - z_3 \leq z_2 - z_1. \]  
(4.40)

Now we shall utilize the “mean value theorem”. Let a and b be two real numbers where \( a \leq \xi \leq b \). Then, for the standard normal distribution, \( \Phi \) which is continuous in \( (a, b) \) we have

\[ \Phi(b) - \Phi(a) = (b - a)\phi(\xi), \]  
(4.41)

where \( \phi \) is the derivative of \( \Phi \). In this case \( \phi \) is the probability density function of the standard normal distribution and it is monotonically decreasing in \( (0, \infty) \).

It follows from the inequality (4.40) that

\[ z_4 - z_3 + \varepsilon = z_2 - z_1, \quad \varepsilon > 0. \]  
(4.42)

Applying the “mean value theorem” to the left and the right hand side of the inequality (4.38), we obtain

\[ (z_4 - z_3)\phi(\xi_1) > (z_2 - z_1)\phi(\xi_2), \]  
(4.43)

where \( z_3 \leq \xi_1 \leq z_4, \quad z_1 \leq \xi_2 \leq z_2 \).

Recalling the inequality (4.36), we can write

\[ z_4 \geq \xi_1 \geq z_3 \geq z_2 \geq \xi_2 \geq z_1 \geq 0. \]  
(4.44)

Substituting equation (4.42) in the inequality (4.43) we obtain

\[ (z_4 - z_3)\phi(\xi_1) > (z_4 - z_3)\phi(\xi_2) + \varepsilon\phi(\xi_2). \]  
(4.45)
from where we can easily derive

$$z_4 - z_3 < - \frac{\varepsilon}{1 - \frac{\phi(\xi_1)}{\phi(\xi_2)}}. \quad (4.46)$$

However, \(\phi(\xi_1) < \phi(\xi_2)\) because \(\xi_1 < \xi_2\) and \(\phi(\xi)\) is monotonically decreasing in \((0, \infty)\). Therefore

$$\frac{\phi(\xi_1)}{\phi(\xi_2)} < 1. \quad (4.47)$$

It follows from the inequality (4.46) owing to the inequalities (4.47) and (4.42)

$$z_4 - z_3 < 0. \quad (4.48)$$

Obviously, as we can see, the relation (4.48) is in contradiction with the relation (4.44). It would mean that \(\frac{1}{2}\sqrt{J_\nu(x_1, x_2, x_3)} < \frac{1}{2}\sqrt{J_\nu(x_1, x_2)}\), which is definitely not true. Hence, the relation (4.40) cannot hold which completes the proof.

The inequality (4.37) thus represents a necessary condition (but unfortunately not a sufficient one – see Choakjarernwanit et al. [3]) for a counter-example to the assumption of Max-Min algorithm to hold. In the following, two simple examples satisfying this condition and fulfilling the relations (4.32) and (4.33) will be presented:

**Case A:**

Let the mean vectors and a common covariance matrix of the two normally distributed classes are as follows:

$$\mu_1 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T, \quad \mu_2 = \begin{bmatrix} 1 & 1 & 2 \end{bmatrix}^T,$$

$$\Sigma = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & -1.75 \\ 1 & -1.75 & 9 \end{bmatrix}.$$

Computing \(z_i\) and \(\Phi(z_i)\) for \(i = 1, 2, \ldots, 4\), we get:

\[z_1 = 0.5000, \quad z_2 = 0.5303,\]
\[z_3 = 0.6124, \quad z_4 = 0.7148,\]
\[\phi(z_1) = 0.6915, \quad \phi(z_2) = 0.7021,\]
\[\phi(z_3) = 0.7299, \quad \phi(z_4) = 0.7626.\]

Then we have
Proof of Invalidity of The Key Premise of The Max-Min Algorithm

\[
\begin{align*}
    z_4 - z_3 &= 0.1024, \\
    z_2 - z_1 &= 0.0303, \\
    \phi(z_4) - \phi(z_3) &= 0.0328, \\
    \phi(z_2) - \phi(z_1) &= 0.0106,
\end{align*}
\]

and thus according to the last two equations obviously (4.32) holds, since

\[
\Delta P_e(x_3, (x_1, x_2)) > 3 \times \Delta P_e(x_3, x_1).
\]

Another example, providing even a more convincing proof of invalidity of the key premise of the Max-Min method is as follows.

Case B:

\[
\begin{align*}
    \mu_1 &= \begin{bmatrix} 0.00 & 0.00 & 0.00 \end{bmatrix}^T, \\
    \mu_2 &= \begin{bmatrix} 0.66 & 0.66 & 0.66 \end{bmatrix}^T, \\
    \Sigma &= \begin{bmatrix} 0.6 & 0.0 & 0.63 \\
                       0.0 & 1.21 & 1.09 \\
                       0.63 & 1.09 & 1.67 \end{bmatrix}.
\end{align*}
\]

The validity of the necessary condition can be easily verified. The \(P_e\)'s of the individual feature are \(P_e(x_1) = 0.3350, P_e(x_2) = 0.3821\) and \(P_e(x_3) = 0.3992\). For the pairs of features we get \(P_e(x_2, x_1) = 0.3012, P_e(x_3, x_2) = 0.3811,\) and \(P_e(x_3, x_1) = 0.3349\). Finally, for all three features \(P_e(x_1, x_2, x_3) = 0.0231\).

According to the Max-Min method, the feature \(x_1\) is selected as the first feature. Then, the remaining features are \(x_2\) and \(x_3\) with the following results:

\[
\Delta P_e(x_2, x_1) = 0.0338, \quad \Delta P_e(x_3, x_1) = 0.0001.
\]

Next, the feature \(x_2\) is chosen as the second selected feature because it gives the maximum decrease of \(\Delta P_e\). Then the results obtained from fusing \(x_3\) with \(x_1\) or \(x_2\) are

\[
\Delta P_e(x_3, x_2) = 0.001, \quad \Delta P_e(x_3, x_1) = 0.0001.
\]

It is noted that fusing feature \(x_3\) to either \(x_1\) or \(x_2\) contributes only a small amount of additional information. However, when \(x_3\) is added to both \(x_1\) and \(x_2\) it actually brings a considerable amount of discriminatory information. This is reflected in the value of
\[ \Delta P_e(x_3, (x_2, x_1)) = 0.278 \] which is not only more than two thousand times greater than 
\[ \min \{ \Delta P_e(x_3, x_1), \Delta P_e(x_3, x_2) \} \], but even more than two hundred times greater than max 
\[ \{ \Delta P_e(x_3, x_1), \Delta P_e(x_3, x_2) \} \]!

Therefore, this result explicitly verifies our claim that the basic relation (4.6) is generally not valid, and hence the Max-Min algorithm is not justified.

4.6 Conclusion

The advantage of the Max-Min over the other methods lies in the fact that instead of computationally time consuming calculations in a multidimensional space, it requires all the calculations to be done only in two-dimensional space. However, we have discovered many weaknesses in the theory underlying this method which cause the degradation of its performance.

Firstly, when a candidate feature is fused with the set of already selected features, the minimum value of the decrease in the pairwise probability of errors is used as a measure of feature effectiveness instead of the actual decrease for that candidate feature. The minimum value is used as the upper bound of the actual reduction in probability of error. Unfortunately, there is no guarantee that a higher upper bound will result in a higher actual value of the decrease in probability of error, i.e. in a higher value of additional information from that feature. Nevertheless, the use of the upper bound in the Max-Min algorithm would be at least heuristically justified, should the minimum value of the decreases in the pairwise probability of errors actually represent the upper bound on the reduction in the probability of error.

Unfortunately, the second and perhaps the most serious drawback of the algorithm is that the assumption concerning the bounding of the decrease in error probability is not true. We have shown that the actual increment of the criterion value (or decrease of the probability of error) can be both smaller or bigger (actually even much bigger) than the minimum of pairwise error probability reductions. The main reason seems to lie in the fact that the method does not take into account possible strong conditional dependencies of features. Without claiming any exactness, we can say that any feature exhibiting a strong conditional dependence with any of the already selected features will cause a great increase of discriminatory information. The resulting decrease of the probability of error can be then much bigger than any of the pairwise decreases. Therefore, the key assumption underlying the whole Max-Min algorithm is not justified and the algorithm itself is ill-founded.

Finally, the method cannot distinguish between genuine correlation and poor perfor-
mance which both manifest themselves by small or zero decrease in the pairwise probability of errors. This fact is connected with the failure to distinguish between conditional and unconditional dependence of features. Various attempts have been made to overcome this problem by introducing the concepts of dependence factors, however with a limited success. This may have been caused primarily by trying to improve the basic Max-Min algorithm which within itself contains serious shortcomings identified above.

To conclude, despite its computational advantages, the method cannot provide a desirable suboptimal solution of the problem of feature selection in a higher-dimensional space owing to its other inherent shortcomings.

References


Chapter 5

Feature Selection in Unknown Distributions

5.1 Introduction

From the previous chapters, it is apparent that when the form underlying the multidimensional probability density is assumed to be parametric, it greatly simplifies the feature selection process and the construction of the classifier. Consequently, it reduces the classifying time. The parametric model will only be as useful as the validity of the known (assumed) underlying densities. No matter how elegant the mathematics is, if the samples are not from the assumed statistics, the accuracy of the pattern recognition process will suffer accordingly. Consequently, in case of a complex pattern space of which the distribution is unknown or multimodal, the use of a simple parametric model such as a multivariate normal distribution can give a very misleading description of the data and consequently an incorrect result. An example showing that the most common assumption of a normal distribution can give a very misleading description of the data is given in Figure 5.1. Clearly, the distributions of the three data sets are different but they all have the same mean and covariance matrix. Thus, it is obvious that first- and second-order statistics are incapable of revealing all the structure in an arbitrary set of data. This leads to unsatisfactory results.

The objective of this chapter is, therefore, to develop an algorithm that can select a good feature set even though the form underlying the probability distribution of the pattern is unknown. The algorithm employed is based on approximating the unknown class conditional distributions by finite mixtures of special type. These mixtures have some useful properties. One of the important features of this mixture model is a simple switch-over from a joint distribution to a marginal one by omitting the superfluous parameters. As a result, the feature effectiveness can then be easily computed.
Figure 5.1: Data sets having identical first- and second-order statistics.
5.2 Approximating the Unknown Class Conditional Distributions

5.2.1 Why Approximation?

Consider the problem of classifying patterns described by real \( D \)-dimensional vectors

\[
\mathbf{x} = (x_1, x_2, \ldots, x_D)^T \in \mathcal{X} \subset \mathbb{R}^D
\]

into a finite set of \( c \) classes \( \Omega = \{\omega_1, \omega_2, \ldots, \omega_c\} \). The patterns are supposed to occur randomly according to some true class conditional probability densities \( p'(\omega|\mathbf{x}) \) and the respective \textit{a priori} probabilities \( P'(\omega) \), i.e., with the joint distribution

\[
p'(\mathbf{x}) = \sum_{\omega \in \Omega} p'(\mathbf{x}|\omega)P'(\omega). \tag{5.1}
\]

Given a vector \( \mathbf{x} \in \mathcal{X} \), we can express the \textit{a posteriori} probabilities of classes

\[
P'(\omega|\mathbf{x}) = \frac{p'(\mathbf{x}|\omega)P'(\omega)}{p'(\mathbf{x})}, \quad p'(\mathbf{x}) > 0, \quad \omega \in \Omega. \tag{5.2}
\]

Consequently, a unique classification of the vector \( \mathbf{x} \) can be made e.g. by using the Bayes decision function, i.e.

\[
D : \mathcal{X} \to \Omega : D(\mathbf{x}) = \omega_i : p'(\mathbf{x}|\omega_i)P'(\omega_i) \geq p'(\mathbf{x}|\omega_j)P'(\omega_j), \quad i \neq j, \quad i, j = 1, 2, \ldots, c \tag{5.3}
\]

which minimizes the probability of error. However, Bayes decision rule requires that the class distributions be known. In practice, these distributions, and in some cases, also the \textit{a priori} class probabilities are seldom specified. Therefore, the classification problem reduces to estimating the unknown distributions \( p'(\mathbf{x}|\omega)P'(\omega), \quad \omega \in \Omega \), which are assumed to be contained in the corresponding samples of independent and identically distributed observations:

\[
\mathcal{X}_\omega = \{x_{1\omega}, x_{2\omega}, \ldots, x_{N_\omega}\} \subset \mathcal{X}, \quad \omega \in \Omega,
\]

where \( N_\omega \) is the number of samples of class \( \omega \).

However, it is well known that in order to successfully estimate the unknown distributions the assumption about the form of the underlying probability distributions must be made. Unfortunately, the underlying structure is usually unknown in almost all practical problems. As a result, instead of finding structure in the data, we would be imposing structure on it. This is why the practical results of estimating multivariate distribution are mostly unsatisfactory. On the other hand, when no underlying structure is assumed, the most common problem encountered is the storage requirement for the multidimensional probability function involved (see [4] and [6] for more details). This problem is
accentuated when the number of parameters involved becomes impractically large. In order to circumvent these difficulties, several approaches have been developed involving approximations of arbitrary multivariate distributions by members of structured families that are characterized by fewer parameters [13], [19], [21], [23], [24]. All approaches are mainly involved with approximating discrete distributions, except [13] of which the general solution can be flexibly applied to both density functions and discrete distributions.

5.2.2 Constructing Finite Mixtures of the Same Subspace

The approach proposed by Grim [13] is very attractive due to the flexibility described above and the fact that it has an analytical solution. His method is based on approximating the unknown class conditional distributions by finite mixtures of special type. The components of mixtures have the form of a product distribution common to all classes which is multiplied by a modified parametric function defined on a subspace \( \mathcal{X} \). Since the subspace can be chosen independently for each component by means of a vector of binary parameters, the method can find the subspaces that describe the pattern of each particular class by means of optimization. Thus, the different classes are independently characterized on different subspaces of \( \mathcal{X} \) which are then used in the classification process. However, the idea underlying feature selection is to obtain features which maximize the similarity of objects in the same class while maximizing the dissimilarity of objects in different classes. Unfortunately, Grim's approach does not concern itself with the latter issue. Nevertheless, it is possible to modify the approach so that when approximating the unknown class conditional distributions the same subspace \( \mathcal{X} \) of each component is used in all classes. Then, feature set effectiveness can be easily calculated and the ultimate goal of feature selection can be fulfilled.

In the following, a modification to the Grim's approach is described. Note that unless otherwise stated, we will be considering the labelled samples of only one class \( \omega \) at a time in order to approximate \( p^*(x|\omega) \) with \( p(x|\omega) \); we hence often omit the subscript indicating class membership for convenience. According to [13], in order to approximate the unknown class conditional distributions (density functions or discrete distributions) the following parametric model is used:

\[
p(x|\omega) = \sum_{m=1}^{M_\omega} w_m^\omega F_0(x|b_0)F(x|b_m^\omega, \phi_m^\omega, b_0), \quad x \in \mathcal{X}, \quad \sum_{m=1}^{M_\omega} w_m^\omega = 1,
\]

where \( M_\omega \) is the number of mixture components and \( w_m^\omega \) are nonnegative weights. Each component of this finite mixture includes a background distribution \( F_0 \) common to all
Approximating the Unknown Class Conditional Distributions

classes:

\[ F_0(x|b_0) = \prod_{i=1}^{D} f(x_i|b_{0i}), \quad b_0 = (b_{01}, b_{02}, \ldots, b_{0D}) \in B^D \]  

(5.5)

and function \( F \) of the form

\[ F(x|b_m^o, \phi_m^o, b_0) = \prod_{i=1}^{D} \frac{f(x_i|b_{mi}^o)}{f(x_i|b_{0i})} \phi_m^o, \quad \phi_m^o \in \{0,1\}, \]

\[ b_m^o = (b_{m1}^o, b_{m2}^o, \ldots, b_{mD}^o) \in B^D, \quad \phi_m^o = (\phi_{m1}, \phi_{m2}, \ldots, \phi_{mD}) \in \{0,1\}^D \]

is actually defined on a subspace specified by nonzero parameters \( \phi_m^o \), i.e.

\[ X_m^o = X_{i1} \times X_{i2} \times \cdots \times X_{i_D}; \{i_1, i_2, \ldots, i_D\} \equiv \{1 \leq i \leq D : \phi_{mi}^o = 1\}. \]

(5.7)

The univariate function \( f \) is assumed to be from a parametric family of probability density functions \( f \) parameterized by \( b \in B \), i.e.

\[ F = \{f(\xi|b), \xi \in \mathcal{R}, b \in B\} \]

(5.8)

with a parameter \( b \). For example, if \( f \in F \) is of the Bernoulli form with a parameter \( b \), then

\[ f(\xi|b) = b^\xi (1-b)^{1-\xi}, \quad \xi \in \{0,1\}, \quad 0 \leq b \leq 1. \]

(5.9)

Alternatively, if \( F \) is the class of univariate Gaussian densities, then \( b = (\mu, \sigma) \) and

\[ f(\xi|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2}\left(\frac{\xi - \mu}{\sigma}\right)^2\right\}, \quad \xi, \mu \in \mathcal{R}, \quad \sigma \in (0, \infty). \]

(5.10)

It can be seen that for any choice of the binary parameters \( \phi_m^o \), the components of the finite mixture (5.4) are valid probability distributions of product type:

\[ F_0(x|b_0)F(x|b_m^o, \phi_m^o, b_0) = \prod_{i=1}^{D} \left[f(x_i|b_{0i})^{1-\phi_{mi}^o}f(x_i|b_{mi}^o)^{\phi_{mi}^o}\right], \]

(5.11)

and the finite mixture (5.4) can then be rewritten as

\[ p(x|\omega) = \sum_{m=1}^{M} w_m \prod_{i=1}^{D} \left[f(x_i|b_{0i})^{1-\phi_{mi}^o}f(x_i|b_{mi}^o)^{\phi_{mi}^o}\right]. \]

(5.12)

It is obvious from equation (5.12) that the considered mixture model is a simple switch-over from a joint distribution to a marginal one by omitting the superfluous parameters. Thus, in the case that the mixture model is Gaussian, it is equivalent to a switch from a full
covariance matrix to a diagonal one, i.e. the model is an independent multivariate Gaussian distribution. An example illustrating the approximation of a 2-dimensional arbitrary distribution by three mixtures of independent bivariate normal densities is schematically shown in Figure 5.2.

In [13] during the optimization process of the finite mixture (5.12), however, the values of the binary parameters $\phi^m_m$ of each class are allowed to continuously change according to an internal criterion function. The process is continued until all the resulting parameters $b_0$, $b^m_0$ and $w^m$ are supposed to characterize that particular class. As a result, different classes $\omega \in \Omega$ can be independently characterized on different subspaces of $\mathcal{X}$.

As far as the goal of feature selection is concerned, it is important that the same subspace $\mathcal{X}$ of each component of mixtures must be used in all classes. This condition is required because such a condition will make the evaluation of the feature set effectiveness simple. The requirement for the same subspace $\mathcal{X}$ of each component of mixtures for all classes can be easily achieved by setting the binary parameters $\phi^m_m \in \{0, 1\}$ equally for all components of mixtures and for all classes, i.e. $\phi = \phi^m_m \in \{0, 1\}$. Equation (5.12) now becomes

$$p(x|\omega) = \sum_{m=1}^{M_o} \sum_{i=1}^{D} \prod_{i=1}^{D} [f(x_i|b_{0i})]^{1-\phi_i} [f(x_i|b^m_{0i})]^{\phi_i},$$

where $m = 1, 2, \ldots, M_o$, $\omega \in \Omega$, $i = 1, 2, \ldots, D$.

Most practical applications of mixture models have been involved with samples from a mixture of two normal, log normal, exponential, or arbitrary (unknown) distributions. So far, applications have ranged from curve-fitting exercises to situations where there are strong physical bases for using mixtures and where objective exogenous information exists to accurately classify data into component distributions. A useful review of past work with mixtures can be found in [7]. To our knowledge, the approximation of the unknown distributions with a mixture of parametric models has not yet been applied to the field of feature selection. This is because previously the mixture components of each class were defined on different subspaces which made both the problems of evaluating feature set effectiveness and feature set search difficult. However, it will be seen later that as a result of approximating the unknown conditional distributions with the model (5.13), the process of feature selection becomes a very simple task.
Figure 5.2: A schematic illustration corresponding to (a) equiprobability contours of a 2-D arbitrary distribution and (b) its approximation by three mixtures of independent bivariate normal densities.
Chapter 5: Feature Selection in Unknown Distributions

5.3 Parameter Optimization

From a practical point of view the optimization of the parametric model (5.13) is of fundamental importance. In the following, a criterion for approximation and our approach to optimization are considered accordingly. Note that the term "approximating" will be used throughout to emphasize that unlike estimation problems, the form of the underlying probability distribution is not known which is usually the case in practical problems.

5.3.1 Approximation Criterion

When approximating one probability distribution with another, it is desirable that the quality of the approximations be associated with the performance of the corresponding recognition or classification schemes when these approximations are employed. For this reason, the commonly used criterion for measuring the quality of approximation $Q$ when approximating the density function $p^*(x|\omega)$ by any other density function $p(x|\omega)$ is the discrimination information or $\ell$-divergence [21], [24] expressed as

$$I(p^*, p) = E_p \{ \log \frac{p^*(x|\omega)}{p(x|\omega)} \} \geq 0.$$  \hspace{1cm} (5.14)

The distinct advantages of $I$ are that, firstly, it is nonnegative and equal to zero if and only if $p^*$ and $p$ are identical [22]. Secondly, it is a convex function on $p$ thus allowing optimization with low computational effort.

Although there may be better approximation criteria such as the Bhattacharyya distance [19], there is in this case the disadvantage that at some point of the approximation generation a full knowledge of the original distribution is required. For those cases where the knowledge of the original distributions is limited, the use of $\ell$-divergence seems quite convenient. Consequently, it follows from equation (5.14) that

$$Q(p) = E_p \{ \log p(x|\omega) \} \leq E_p \{ \log p^*(x|\omega) \}$$ \hspace{1cm} (5.15)

and therefore $Q(p)$ can be used as a criterion to be maximized by $p$. The approximating density $p(x|\omega)$ of the true class densities $p^*(x|\omega)$, $\omega \in \Omega$, can then be used to construct a parametric classifier in the usual way. However, the estimation of the parameters of $Q(p)$ in case of mixtures is known to be a difficult problem, both from the statistical and algorithmic points of view (see for example Everitt and Hand [8] and Titterington et al. [29]).
5.3.2 Maximum Likelihood Estimation

There are, of course, many approaches which can be used to optimize the parameters of $Q(p)$ or, effectively, of the approximating mixtures (5.13). The literature surrounding mixture distributions is large and goes back to the end of the last century when Pearson [25] published his well known paper on estimating the five parameters in a mixture of two normal distributions using the method of moments. Since then a number of techniques such as graphical estimation, minimum chi-squared estimation, least squares method, Bayesian estimation, and maximum likelihood estimation etc., have been developed. A useful review of these approaches can be found in [8], [17], [27] and [29]. Of these techniques, the most preferable is to use maximum likelihood (ML) estimate. This is primarily due to the following reasons. First, it is obvious from equation (5.15) that if $p^*$ is unknown, the log-likelihood function for $p$ may be viewed as an estimate of $Q(p)$. Second, several studies, e.g. [2], [5], [9], [10], [12], [15], [18] and [28], have shown that the estimation by maximum likelihood is more efficient than the other potential methods. Additionally, it also possesses desirable statistical properties. For example, under very general conditions the estimated parameters obtained by the method are consistent, i.e. they converge in probability to the true parameter values, and they are asymptotically normally distributed. In the following, the idea of the maximum likelihood estimate for our model is described.

Let us suppose that for each class $\omega \in \Omega$ there is a set $X_{\omega}$ of $N_{\omega}$ independent observations which are identically distributed according to an unknown conditional density function $p^*(x|\omega)$. In order to optimize the approximating mixture (5.13) we maximize the corresponding global log-likelihood function. Using the following notation

$$p(x|W_\omega, B_\omega, \phi, b_0) = F_0(x|b_0) \sum_{m=1}^{M_\omega} w_m^\omega F(x|b_m^\omega, \phi, b_0);$$

$$W_\omega = (w_1^\omega, w_2^\omega, \ldots, w_{M_\omega}^\omega); w_m^\omega \geq 0; \sum_{m=1}^{M_\omega} w_m^\omega = 1;$$

$$B_\omega = (b_1^\omega, b_2^\omega, \ldots, b_{M_\omega}^\omega); \omega \in \Omega;$$

$$b_0 = (b_{01}, b_{02}, \ldots, b_{0D}), \phi = (\phi_1, \phi_2, \ldots, \phi_D) \in \{0, 1\}^D,$$

the corresponding global log-likelihood function $L_G$ can be expressed as

$$L_G = \frac{1}{N_0} \sum_{\omega \in \Omega} \sum_{x \in X_\omega} \log [p(x|W_\omega, B_\omega, \phi, b_0)P(\omega)]$$

$$= \sum_{\omega \in \Omega} \frac{N_\omega}{N_0} \log P(\omega) + \sum_{\omega \in \Omega} \frac{N_\omega}{N_0} \sum_{x \in X_\omega} \log p(x|W_\omega, B_\omega, \phi, b_0)$$

(5.17)
where \( N_0 \) is the total number of samples of all classes, i.e. \( N_0 = \sum_{\omega \in \Omega} N_\omega \).

Usually the \textit{a priori} probabilities \( P(\omega) \) may be estimated by the respective relative frequencies. However, sometimes these probabilities are not accurately related to the respective sample sizes \( N_\omega \). For this reason we confine ourselves to the second part of equation (5.17) and replace the relative frequencies \( N_\omega/N_0 \) by input parameters \( P(\omega) \). Using symbols \( W = \{ W_\omega, \omega \in \Omega \} \), \( B = \{ B_\omega, \omega \in \Omega \} \) we denote the second part of equation (5.17) by

\[
L(W, B, \phi, b_0) = \sum_{\omega \in \Omega} \frac{P(\omega)}{N_\omega} \sum_{x \in X_\omega} \log p(x|W_\omega, B_\omega, \phi, b_0)
\]

\[
= \sum_{\omega \in \Omega} \frac{P(\omega)}{N_\omega} \sum_{x \in X_\omega} \log \left[ \sum_{m=1}^{M_\omega} \nu_\omega^m F_0(x|b_0) F(x|b_\omega^m, \phi, b_0) \right]. \tag{5.18}
\]

The maximum likelihood estimates of parameters for the mixture (5.13) are then found by maximizing the corresponding likelihood function \( L(W, B, \phi, b_0) \) generated by all the samples with respect to parameters \( W, B, \phi \) and \( b_0 \). Intuitively, this corresponds to the values of \( W, B, \phi \) and \( b_0 \) that in some sense best agree with the actually observed samples. Unfortunately likelihood equations obtained by setting derivatives of \( L(W, B, \phi, b_0) \) to zero seem to have no explicit solution in case of mixtures. This difficulty arises because of the complex dependence of the likelihood function on the parameters to be estimated. Consequently, the alternative is to seek an approximate solution via some iterative procedure.

5.3.3 Optimization Using The EM Algorithm

Maximum likelihood estimation has become the most widely followed approach to the mixture density estimation problem since the advent of high speed computers. Actually, maximum likelihood estimates and their associated efficiency were often the subject of wishful thinking prior to the advent of computers, but research toward obtaining maximum likelihood estimates was restricted to the simple mixtures for computational reasons.

As computers became available to ease the burden of computation, various general iterative procedures have been developed for obtaining the maximum likelihood estimates for a variety of increasingly complex mixture densities, e.g. [2], [5], [9], [14], [26] and [30]. These iterative methods are basically the generalization of either Newton's iterative method or the steepest ascent iterative method, or the combination of both. Of these generalized iterative methods, the simplest is, perhaps, that suggested by Hasselblad [14]. Although, in practice, his algorithm always converges to a maximum on the
Parameter Optimization

likelihood surface, he was unable to prove such convergence.

The recent paper by Dempster et al. [3] on the general expectation-maximization (EM) algorithm has drawn attention to the simplicity of iterative maximum likelihood estimation in a wide class of "incomplete data" problems, including mixture distributions. It is known as the EM algorithm since at each iteration of the algorithm it consists of an expectation step (E-step) followed by a maximization step (M-step). This algorithm has a number of attractive features:

• The algorithm is guaranteed to converge to a local if not a global maximum (correctly proved by [1], [31]). The speed of convergence is, however, a function of separation of the mixture components, and for components which are close together, the convergence may be slow. Dempster et al. [3] suggested a number of approaches for increasing the rate of convergence.

• Every iteration of the algorithm is guaranteed to increase the log-likelihood.

• Low cost per iteration and economy of storage.

• The iteration estimates always yield valid parameter values, e.g., positive variance and mixing proportion between zero and one.

In addition, the iterative estimates have the usual attractive asymptotic properties of all maximum likelihood estimates. Dempster et al. [3] also proved that Hasselbalch's algorithm is essentially an application of the EM algorithm and, as such, is guaranteed to converge to at least a local maximum. For these reasons, we will use the EM algorithm to optimize the parameters in the log-likelihood function (5.18) and the two fundamental steps of the EM algorithm may be specified as follows:

• E - step: Given the parameters $W, B, \phi$ and $b_0$ compute the a posteriori probabilities

$$p(m|x, \omega) = \frac{w_m^\omega F(x|b_m^\omega, \phi, b_0)}{\sum_{j=1}^{M_\omega} w_j^\omega F(x|b_j^\omega, \phi, b_0)},$$

(5.19)

$$m = 1, 2, \ldots, M_\omega, \quad x \in X_\omega, \quad \omega \in \Omega,$$

and determine the conditional expectation

$$\mathcal{L}(W, B, \phi, b_0) = \sum_{\omega \in \Omega} N_\omega \sum_{x \in X_\omega} \left\{ \sum_{m=1}^{M_\omega} p(m|x, \omega) \log [w_m^\omega F_\omega(x|b_0)F(x|b_m^\omega, \phi, b_0)] \right\}.$$

(5.20)
• M – step: Under fixed weights (5.19) compute the new values of $W, B$ and $b_0$ denoted correspondingly as $'W$, $'B$ and $'b_0$ by maximizing the function $\mathcal{L}$:

$$\mathcal{L}(W, B, \phi, b_0) = \arg\max_{W, B, b_0} \{ \mathcal{L}(W, B, \phi, b_0) \}. \quad (5.21)$$

Thus as it follows from equation (5.20), $\mathcal{L}(W, B, \phi, b_0)$ is the expectation of the complete likelihood given observation $X_o$, $\omega \in \Omega$, and the current parameters $W, B, \phi$ and $b_0$, so rather than maximizing the complete data log-likelihood, which is not known, we are maximizing the current conditional expectation of the complete data log-likelihood. In other words, the EM algorithm transforms the original problem to a repeated maximization of equation (5.20) which may be viewed as a weighted version of $L$. Obviously, the application of the EM algorithm is efficient only if we derive a simple explicit solution of equation (5.21). For this purpose, we use first the substitution (5.13) in equation (5.20)

$$\mathcal{L}(W, B, \phi, b_0) = \sum_{\omega \in \Omega} P(\omega) \sum_{m=1}^{M_o} \frac{1}{N_\omega} \sum_{x \in X_\omega} p(m|x, \omega) \log w_m^\omega$$

$$+ \sum_{i=1}^{D} (1 - \phi_i) \sum_{\omega \in \Omega} \frac{P(\omega)}{N_\omega} \sum_{m=1}^{M_o} \sum_{x \in X_\omega} p(m|x, \omega) \log f(x_i|b_0^\omega)$$

$$+ \sum_{i=1}^{D} \phi_i \sum_{\omega \in \Omega} \frac{P(\omega)}{N_\omega} \sum_{m=1}^{M_o} \sum_{x \in X_\omega} p(m|x, \omega) \log f(x_i|b_m^\omega). \quad (5.22)$$

Further, denote

$$w_m^\omega = \frac{1}{N_\omega} \sum_{x \in X_\omega} p(m|x, \omega) \quad (5.23)$$

and

$$v(x|m, \omega) = \frac{p(m|x, \omega)}{\sum_{y \in X_\omega} p(m|y, \omega)}. \quad (5.24)$$

Then equation (5.22) can be rewritten as

$$\mathcal{L}(W, B, \phi, b_0) = \sum_{\omega \in \Omega} P(\omega) \sum_{m=1}^{M_o} w_m^\omega \log w_m^\omega$$

$$+ \sum_{i=1}^{D} (1 - \phi_i) \sum_{\omega \in \Omega} P(\omega) \sum_{m=1}^{M_o} w_m^\omega \sum_{x \in X_\omega} v(x|m, \omega) \log f(x_i|b_0^\omega)$$

$$+ \sum_{i=1}^{D} \phi_i \sum_{\omega \in \Omega} P(\omega) \sum_{m=1}^{M_o} w_m^\omega \sum_{x \in X_\omega} v(x|m, \omega) \log f(x_i|b_m^\omega). \quad (5.25)$$

According to [13], it can be seen that for any fixed binary parameters $\phi_i$ and under fixed weights $v(x|m, \omega)$ the function (5.25) is maximized by $W = 'W$ (c.f. equation(5.23)), $B = 'B$ and $b_0 = 'b_0$ where
Parameter Optimization

\[ 'b_{mi}^\omega = \arg \max_{b \in B} \{ \sum_{x \in X^\omega} v(x|m, \omega) \log f(x|b) \}, \quad (5.26) \]

\[ 'b_{0i}^\omega = \arg \max_{b \in B} \{ \sum_{m=1}^{M_\omega} P(\omega) \sum_{x \in X^\omega} v(x|m, \omega) \log f(x|b) \} \quad (5.27) \]

\[ m = 1, 2, \ldots, M_\omega; \quad i = 1, 2, \ldots, D; \quad \omega \in \Omega. \]

Consequently, the following inequality is satisfied

\[ \mathcal{L}(W, B, \phi, b_0) \leq \mathcal{L}(W, 'B, \phi, 'b_0). \quad (5.28) \]

Now substituting \( W = W, B = 'B, b_0 = 'b_0 \) in equation (5.25) and introducing the quantities

\[ 'q_{mi}^\omega = P(\omega) w_m^\omega \sum_{x \in X^\omega} v(x|m, \omega) \log \frac{f(x|b_{mi}^\omega)}{f(x|'b_{0i}^\omega)}, \quad (5.29) \]

\[ m = 1, 2, \ldots, M_\omega; \quad i = 1, 2, \ldots, D; \quad \omega \in \Omega, \]

we obtain

\[ \mathcal{L}(W, 'B, \phi, 'b_0) = \sum_{\omega \in \Omega} P(\omega) \sum_{m=1}^{M_\omega} 'w_m^\omega \log 'w_m^\omega + \sum_{i=1}^{D} \phi_i \sum_{\omega \in \Omega} \sum_{m=1}^{M_\omega} 'q_{mi}^\omega \]

\[ + \sum_{i=1}^{D} \sum_{\omega \in \Omega} P(\omega) \sum_{m=1}^{M_\omega} 'w_m^\omega \sum_{x \in X^\omega} v(x|m, \omega) \log \frac{f(x|b_{mi}^\omega)}{f(x|'b_{0i}^\omega)}. \quad (5.30) \]

In addition, we define

\[ 'Q_i = \sum_{\omega \in \Omega} \sum_{m=1}^{M_\omega} 'q_{mi}^\omega \]

and if we order the terms \( 'Q_i \) in a descending way

\[ \{ 'Q_1, 'Q_2, \ldots, 'Q_k \}_{k=1}^{D}, \quad 'Q_i \geq 'Q_{i+1} \]

and set

\[ '\phi_i = \begin{cases} 1, & k = 1, 2, \ldots, d; \\ 0, & k = d + 1, \ldots, D; \quad 1 \leq i_k \leq D. \end{cases} \]

then the parameters \( \phi_i \) satisfy the inequality
Chapter 5: Feature Selection in Unknown Distributions

\( \mathcal{L}(W, B, \phi, b_0) \leq \mathcal{L}(\hat{W}, \hat{B}, \hat{\phi}, \hat{b}_0). \) (5.31)

The relations (5.28) and (5.31) already imply that the maximized function (5.18) is non-decreasing at each iteration of the E- and M-steps.

It can be seen that, without formal difficulties, the univariate function \( f \) in equation (5.25) may be chosen from different families for each index \( i \). In this way, for example, discrete and continuous variables may occur simultaneously in the vector \( x \). Note that the application of the EM algorithm to the estimation of mixture parameters is only one of many possible uses. There are many other possible applications in the field of data classification in general (see [3] for more details).

Like all other currently available estimation procedures, the EM algorithm does not guarantee convergence to a global maximum. A global maximum can only be ensured by an exhaustive search of the multidimensional likelihood surface. However in the context of approximating, the existence of local maxima of the function \( L \) merely implies different approximation possibilities of different quality. Similarly the choice of the number of components in equation (5.13) influences only the quality of approximation. The frequently discussed slow convergence of the EM algorithm in the final stages of computation is also of little importance since the corresponding changes of the criterion are usually negligible.

5.3.4 Application to a Particular Type of Mixtures

In some important cases the parameters \( \beta_{0i} \) can be expressed as linear combinations of \( \beta_{mi}^{(i)} \):

\[ \beta_{0i} = \sum_{\omega \in \Omega} P(\omega) \sum_{m=1}^{M_\omega} \omega_m \beta_{mi}^{(i)} ; i = 1, 2, \ldots, D. \] (5.32)

Using the results of Subsection 5.3.3 we summarize the EM algorithm in more detail. With respect to the particular choice of mixtures in this section we use formula (5.32) instead of (5.27). Using the results of Grim [13] we obtain
• Step 1: Given the parameters $W$, $B$, $\phi$ and $b_0$ compute the weights:

\begin{align}
 p(m|x, \omega) &= \frac{w_m^\omega F(x|b_m^\omega, \phi, b_0)}{\sum_{j=1}^{M_\omega} w_j^\omega F(x|b_j^\omega, \phi, b_0)}, \\
v(x|m, \omega) &= \frac{p(m|x, \omega)}{\sum_{y \in X_\omega} p(m|y, \omega)} \\
m = 1, 2, \ldots, M_\omega; \quad x \in X_\omega; \quad \omega \in \Omega.
\end{align}

(5.33) (5.34)

• Step 2: Under fixed weights (5.33) and (5.34) compute new values of $W$ and $B$ by the formulas

\begin{align}
 w_m^\omega &= \frac{1}{N_\omega} \sum_{x \in X_\omega} p(m|x, \omega), \\
b_{mi}^\omega &= \arg\max_{b \in B} \left\{ \sum_{x \in X_\omega} v(x|m, \omega) \log f(x_i|b) \right\} \\
m = 1, 2, \ldots, M_\omega; \quad i = 1, 2, \ldots, D; \quad \omega \in \Omega.
\end{align}

(5.35) (5.36)

• Step 3: Given the parameters $W'$, $B'$ and $\phi$ compute the new value of $b_0$ by

\begin{align}
 b_{0i}' &= \sum_{\omega \in \Omega} P(\omega) w_m^\omega b_{mi}^\omega, \quad i = 1, 2, \ldots, D; \quad \omega \in \Omega.
\end{align}

(5.37)

If $W' \neq W, B' \neq B, b_0' \neq b_0$ continue by Step 1 using the new parameters $W', B'$ and $b_0'$. Otherwise continue by Step 4.

• Step 4: Using the parameters $W', B', b_0'$ and the weights (5.34) compute the quantities

\begin{align}
 q_{mi}' &= P(\omega)' w_m^\omega \sum_{x \in X_\omega} v(x|m, \omega) \log \frac{f(x_i|b_{mi}^\omega)}{f(x_i|b_0')}, \\
m = 1, 2, \ldots, M_\omega; \quad i = 1, 2, \ldots, D; \quad \omega \in \Omega.
\end{align}

(5.38)

and

\begin{align}
 Q_i &= \sum_{\omega \in \Omega} \sum_{m=1}^{M_\omega} q_{mi}'.
\end{align}

(5.39)

Find a monotone order

\begin{align}
 Q_i' \geq Q_{i+1} \geq \cdots \geq Q_i \geq \cdots \geq Q_0',
\end{align}

(5.40)
and define
\[ \phi_k = \begin{cases} 1, & k = 1, 2, \ldots, d; \\ 0, & k = d + 1, \ldots, D; \end{cases} \]
\(1 \leq k \leq D.\)

If \(\phi \neq \phi\) continue by Step 1 with \(W = W, B = B, b_0 = b_0\) and \(\phi = \phi\).

**Gaussian Model**

Let \(F = \{f(\xi | \mu, \sigma), \xi \in \mathcal{R}, \mu \in \mathcal{R}, \sigma \in (0, \infty)\}\) be the class of univariate normal densities (5.10) with a pair of parameters \(\mu, \sigma\) standing for \(b\). By the formulas (5.11) and (5.12) we have for \(\phi_m^\omega = \phi_i\) for all \(i \in \Omega\) and \(m = 1, 2, \ldots, M\):

\[
F_0(x | \mu_0, \sigma_0) = \prod_{i=1}^{D} \left[ \frac{1}{\sqrt{2\pi\sigma_0}} \exp\left\{ -\frac{1}{2} \frac{(x_i - \mu_0)^2}{\sigma_0} \right\} \right], \tag{5.41}
\]

\[
F(x | \mu_m^\omega, \sigma_m^\omega, \mu_0, \sigma_0, \phi) = \prod_{i=1}^{D} \left[ \frac{\sigma_0}{\sigma_m^\omega} \exp\left\{ -\frac{1}{2} \frac{(x_i - \mu_m^\omega)^2}{\sigma_m^\omega} + \frac{1}{2} \frac{(x_i - \mu_0)^2}{\sigma_0} \right\} \right]^{\phi_i}. \tag{5.42}
\]

\(\mu_m^\omega \in \mathcal{R}, \sigma_m^\omega \in (0, \infty), m = 1, 2, \ldots, M; \omega \in \Omega.\)

The components of the approximating mixture are therefore normal densities with diagonal covariance matrices. In this case for a given \(w_m^\omega, \mu_m^\omega, \sigma_m^\omega, \phi_i\) compute the weights by the following equations

\[
p(m|x, \omega) = \frac{w_m^\omega \prod_{i=1}^{D} \left[ \sigma_0 \exp\left\{ -\frac{1}{2} \frac{(x_i - \mu_m^\omega)^2}{\sigma_m^\omega} + \frac{1}{2} \frac{(x_i - \mu_0)^2}{\sigma_0} \right\} \right]^{\phi_i}}{\sum_{j=1}^{M} w_j \prod_{i=1}^{D} \left[ \sigma_0 \exp\left\{ -\frac{1}{2} \frac{(x_i - \mu_j^\omega)^2}{\sigma_j^\omega} + \frac{1}{2} \frac{(x_i - \mu_0)^2}{\sigma_0} \right\} \right]^{\phi_i}}, \tag{5.43}
\]

\[
v(x | m, \omega) = \frac{p(m|x, \omega)}{\sum_{y \in \mathcal{X}_\omega} p(m|y, \omega)}, \tag{5.44}
\]

\(m = 1, 2, \ldots, M; x \in \mathcal{X}_\omega, \omega \in \Omega.\)

Under fixed weights (5.43) and (5.44) compute new value \(w_m^\omega\)

\[
'w_m^\omega = \frac{1}{N_\omega} \sum_{x \in \mathcal{X}_\omega} p(m|x, \omega). \tag{5.45}
\]

The implicit relation (5.36) is transformed by the substitution (5.10) to the form

\[
(\mu_m^\omega, \sigma_m^\omega) = \arg \max_{\mu \in \mathcal{R}, \sigma \in (0, \infty)} \left\{ \sum_{x \in \mathcal{X}_\omega} v(x | m, \omega) \log \left[ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2} \frac{(x_i - \mu)^2}{\sigma^2} \right\} \right] \right\}. \tag{5.46}
\]

where the parenthesized sum represents a weighted version of the usual log-likelihood function for normal density. It can be shown that this expression is maximized by an analogously weighted version of the maximal likelihood estimate:
\[ '\mu_{ni}^\omega = \sum_{x \in X} x_i p(x|m, \omega); \quad i = 1, 2, \ldots, D; \quad (5.47) \]
\[ ('\sigma_{ni}^\omega)^2 = \sum_{x \in X} (x_i - '\mu_{ni}^\omega)^2 p(x|m, \omega); \quad m = 1, 2, \ldots, M_\omega; \quad \omega \in \Omega. \quad (5.48) \]

Similarly, we would obtain an explicit solution of the relation (5.37) which can be rewritten in the form
\[
'\mu_{0i} = \sum_{\omega \in \Omega} P(\omega) \sum_{m=1}^{M_\omega} w_m^\omega '\mu_{mi}^\omega; \quad i = 1, 2, \ldots, D; \quad (5.49) 
\]
\[
('\sigma_0)^2 = \sum_{\omega \in \Omega} P(\omega) \sum_{m=1}^{M_\omega} w_m^\omega [('\sigma_{mi}^\omega)^2 + ('\mu_{mi}^\omega - '\mu_0)^2] \quad (5.50) 
\]

Finally, the formula (5.38) can be simplified as follows
\[
'q_{ni}^\omega = 'w_m^\omega P(\omega) \log \frac{('\sigma_0)^2}{('\sigma_{ni}^\omega)^2}. \quad (5.51) 
\]

5.4 Approach to Feature Selection

5.4.1 Feature Selection Criterion

Our approach to feature selection problem is to select the best subset of \( d \) features \( X_d = \{x_i | k = 1, 2, \ldots, d; x_i \in X\} \) from the set \( X = \{x_i | i = 1, 2, \ldots, D\} \) of \( D, D > d \) possible features representing the pattern, which minimizes the criterion

\[
J = \min_{W, B, b_0} \sum_{\omega \in \Omega} P(\omega) E_{\rho'} \left\{ \log \frac{p^*(x|\omega)}{p(x|W_{\omega}, B_{\omega}, b_0, \phi)} \right\} \quad (5.52) 
\]

with respect to any other combination of \( d \) features from \( X \), where the approximating class conditional density is a mixture of normal densities with diagonal covariance matrices. That is we attempt to find such features from \( D \) features which are best in the sense of minimizing the Kullback-Leibler distance between the true class conditional densities of \( x \) and the postulated class conditional densities mixed in the proportions in which the classes truly occur.

Let us denote for simplicity \( \Theta = (W, B, b_0) \). Following Hjort [16], we have that as \( N_0 \to \infty, L(\Theta, \phi) \) tends almost surely to

\[
\sum_{\omega \in \Omega} P(\omega) E_{\rho'} \left\{ \log p(x|\Theta_{\omega}, \phi) \right\}. \quad (5.53) 
\]
Suppose there is a unique value $\Theta_0$ of $\Theta$ that maximizes (5.53) with respect to $\Theta$. Then it also minimizes the quantity

$$
\sum_{\omega \in \Omega} P(\omega) E_p \left\{ \log \frac{p^*(x|\omega)}{p(x|\Theta_0, \phi)} \right\}
$$

(5.54)

which is a mixture in the true proportions $P(\omega_1), P(\omega_2), \ldots, P(\omega_k)$ of the Kullback-Leibler distance between the true and postulated class conditional densities of $x$.

Under mild regularity conditions it follows that if $\hat{\Theta}$ is chosen by maximization of $L(\Theta, \phi)$, it tends almost surely to $\Theta_0$. Hence the maximum likelihood estimator of $\Theta$ is a consistent estimator of $\Theta_0$ which is the value of $\Theta$ that minimizes (5.54). Therefore, we have to find the subset $X_d$ minimizing the minimum of (5.54) with respect to $\Theta$. In other words, to find the subset $X_d$ that maximizes $L(\hat{\Theta}, \phi)$. It follows from the result of Redner and Walker [27] that this is equivalent to maximization of $L(\phi', \phi)$ given by (5.30) for $\phi'_{m_i}$ as in (5.51). Therefore, we want to find a subset $X_d$ that maximizes with respect to any other combination of $d$ features criterion

$$
Q = \sum_{i=1}^{D} \phi'_i Q_i,
$$

(5.55)

where

$$
Q_i = \sum_{\omega \in \Omega} \sum_{m=1}^{M_\omega} \phi'_m P(\omega) \log \left( \frac{\sigma_{\omega,i}}{\sigma_{\omega,m}} \right)^2.
$$

(5.56)

If we order $Q_i$ in descending way, i.e.

$$
Q_{i_1} \geq Q_{i_2} \geq \cdots \geq Q_{i_d} \geq \cdots \geq Q_{i_0}
$$

and set

$$
\phi_k = \begin{cases}
1, & k = 1, 2, \ldots, d;
1, & k = d + 1, \ldots, D,
0, & k = d + 1, \ldots, D,
\end{cases}
$$

then the subset $X_d = \{x_1, x_2, \ldots, x_d\}$ minimizes the criterion $J$ with respect to any other combinations of $d$ features from $X$. Now, given approximations

$$
p(x|\omega) = \prod_{i=d+1}^{D} f(x_i|b_{0_i}) \sum_{m=1}^{M_\omega} \phi'_m \prod_{i=1}^{d} f(x_i|b_{m_i}), \omega \in \Omega,
$$

(5.57)

we may classify the observation of $x$ according to the pseudo-Bayes rule: decide that $x$ is from class $\omega_k$ if

$$
P(\omega_k) \sum_{m=1}^{M_\omega} \phi'_m \prod_{i=1}^{d} f(x_i|b_{m_i}) > P(\omega_j) \sum_{m=1}^{M_\omega} \phi'_m \prod_{i=1}^{d} f(x_i|b_{m_i}), k \neq j; j, k = 1, 2, \cdots c.
$$

(5.58)
5.4.2 Characteristics of the Proposed Feature Selection Method

In the last section we have described some theoretical properties of the chosen approach to approximate unknown distributions as well as the theoretical reasoning of the proposed feature selection method. As far as the objective of feature selection is concerned, it is obvious that the primary goal of our approach is not to select the most discriminative features but rather the features which maximize the likelihood function (5.18). As it has been shown, this is equivalent to minimizing the Kullback-Leibler distance between the true class conditional densities of \( p \) and the postulated class densities.

A question arises as to why we use the features that are optimal from the point of view of approximating the unknown distributions for discrimination between the classes. A similar situation arises in the case of Karhunen-Loeve (K-L) expansion in discrete form, which is also known as the principal component analysis. The feature extraction methods based on the K-L expansion (see e.g. Fukunaga [11], Kittler and Young [20]) have been widely used though the original goal was not to extract features with respect to their discriminative power but rather features minimizing the mean square error of representing patterns in a lower dimensional feature subspace. The underlined assumption is that features which are good for representing patterns and describing well their spatial structure will also be good for discriminating between the classes.

The analogous reasoning can be used in our case. When finding features which are best from the point of view of approximating unknown class distributions, we can hope that they will be good for discriminating between the classes as well. Obviously, this claim is not always justified, neither in the case of the K-L expansion, nor in the case of our approach to feature selection. One can easily construct counter-examples when features, which are not very useful from the point of view of representing the patterns (and which are therefore discarded by feature extraction based on the K-L expansion), are at the same time carrying almost all discriminative power. Figure 5.3 depicts such an example where it is clear that, for classification purposes, the transformation of measurements on to \( e_2 \) is preferred to \( e_1 \). This is because it preserves more discriminatory information. However, according to the K-L expansion the transformation of measurements on to \( e_1 \) minimizes the mean square error of representing patterns in a lower dimensional feature subspace. Therefore, it will be chosen as the most important feature for classification which is obviously incorrect.

Similarly, one can find examples where features maximizing the likelihood function (5.18) and minimizing the Kullback-Leibler distance between the true class conditional densities of \( p \) and the postulated class densities, are not always the best ones with respect
Figure 5.3: A counter-example for the Karhunen-Loève expansion. (a) Correlation ellipses (density contours) of both measurements $y_1$ and $y_2$. (b) Projection of $y_1$ and $y_2$ on to $e_1$ axis. (c) Projection of $y_1$ and $y_2$ on to $e_2$ axis.
Figure 5.4: A schematic illustration corresponding to equiprobability contours of 2-D distributions in which our approach fails.

to discrimination. An example demonstrating this situation is given in Figure 5.4 where the marginal density functions of the measurement $y_1$ are well separated with little overlap comparing to those of the measurement $y_2$. Obviously, the measurement $y_1$ should be chosen as the most important feature for classification, but in our approach during the experiment with similar distributions the measurement $y_2$ has been chosen instead due to the reasons stated above.

However, this does not mean that the method cannot be used. It only confirms the fact that there is no universally usable and computationally efficient method which would guarantee the optimal results under any circumstances. However, the proposed method possesses some unique properties which make it very useful in practice. First of all, we should stress once more that as it follows from Step 4 of the algorithm, after computing the parameters of the mixture and then ordering $Q_i$ according to the relation (5.40), we get at the same time the optimal ordering of all the original $D$ features. Thus a feature subset of any cardinality $d$ where $d = 1, 2, \ldots, D$ can be obtained immediately. A computationally time consuming search procedure usually associated with a selected criterion is not needed in our approach. More importantly, the criterion used in our approach is already in the form corresponding to a multiclass problem, therefore the ordering of all the original $D$ features obtained is also optimal with respect to all the classes.

Another interesting property is that the method is independent of the class distributions, not even the form of the distribution is assumed. Because it is based on the approx-
Figure 5.5: A schematic illustration corresponding to equiprobability contours of 2-D distributions in which feature selection using distance measure approach fails.

imation of unknown distributions by a mixture of Gaussian densities, it is particularly useful for the case of multimodal distributions when other feature selection methods based on distance measures (e.g. Mahalanobis distance, Bhattacharyya distance) would totally fail to provide reasonable results as shown in Figure 5.5. In this case, when the unimodal-multinormal model is assumed, the difference in the means of both classes in the $y_2$ direction is much smaller than that in the $y_1$ direction. Therefore, the measurement $y_1$ will be chosen as the most important feature instead of the measurement $y_2$ which contains more discriminatory information than $y_1$ as the computation has shown. The feature correctly selected in this case by our approach is $y_2$.

5.5 Experiments

Our aim was to develop feature selection algorithm based on modelling class probability distributions in terms of marginal Gaussian density functions. Obviously, the success of the feature selection stage is heavily dependent upon the accuracy of approximation of the unknown distributions.

A number of experiments have been conducted on both simulated and real data. The performance of our model is compared with that of the ordinary multinormal probability density model. However, a comparison, using the parametric probabilistic distance measure, between the two approaches as in the Chapter 3 is not possible since different
assumptions about the underlying probability distributions were used. As a result, the alternative way is to directly compare the performance of both approaches using the misclassification rate obtained from both types of classifiers, i.e. the mixture of normal and the multinormal based classifiers. Separate training and test sets were used in all experiments. In addition, the a priori probabilities in all experiments were taken to be equal for all classes. Note that when the unknown distribution is assumed to be unimodal multivariate normal, the best feature subset of size \( d \) is selected from the set of \( D \) available measurements using the branch and bound algorithm with the Bhattacharyya distance as the criterion function, unless stated otherwise.

5.5.1 Simulated Data

The data for each class was generated from a mixture of two four-dimensional normal distributions with the following characteristics:

<table>
<thead>
<tr>
<th>class ((\omega_i))</th>
<th>component ((k))</th>
<th>weight ((w_{ik}))</th>
<th>(\mu_{ik})</th>
<th>(\Sigma_{ik})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>[0.00]</td>
<td>1.00 0.00 0.00 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td>0.00 1.00 0.00 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td>0.00 0.00 1.00 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td>0.00 0.00 0.00 1.00</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0.5</td>
<td>[3.86]</td>
<td>8.41 0.00 0.00 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.84</td>
<td>0.00 0.22 0.00 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.08</td>
<td>0.00 0.00 1.77 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.00</td>
<td>0.00 0.00 0.00 2.73</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.5</td>
<td>[0.00]</td>
<td>1.00 0.00 0.00 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td>0.00 1.00 0.00 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.00</td>
<td>0.00 0.00 1.00 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.00</td>
<td>0.00 0.00 0.00 1.00</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.5</td>
<td>[3.86]</td>
<td>8.41 0.00 0.00 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.84</td>
<td>0.00 0.22 0.00 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.08</td>
<td>0.00 0.00 1.77 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td>0.00 0.00 0.00 2.73</td>
</tr>
</tbody>
</table>

Table 5.1: Parameters used in simulating the data.

The mixture densities are of the form

\[
p(x | \omega_i) = \sum_{k=1}^{2} \frac{w_{ik}}{\sqrt{(2\pi)^D |\Sigma_{ik}|}} \exp\left\{-\frac{1}{2} (x - \mu_{ik})^T \Sigma_{ik}^{-1} (x - \mu_{ik}) \right\}.
\] (5.59)
Table 5.2: Classifier performance on different feature subset size as a function of the size of the training set.

The experimental results are given in Table 5.2. It can be clearly seen that for all the various feature set sizes, the performance of our approach is much better than that of the multinormal model. As far as the training sample size is concerned, the larger size yields better results in both approaches. In addition, if the training set is large, a higher number of mixture components is expected to perform better because of the larger flexibility in fitting feature vector components with different widths of the individual peaks (components). It is important to note that our approach requires much more computation than the multivariate Gaussian density assumption. Furthermore, the higher the number of mixtures, the longer the computational time and the larger the training set required. However, a better result is usually obtained.

5.5.2 Texture Data

A number of different images have been tested but the two colour images shown in Appendix A were specifically chosen since they are not well separated in the measurement space. If we were to use images that are rather well separated in the measurement space, then the advantage of applying our model will be less significant.

Each colour image was divided into two halves, the top half was used for training and the bottom half was used for classification propose. The size of each image was 256 × 256 from which sample sub-images were selected with window of size 100 × 100 randomly placed. Then a 26-dimensional feature vector was extracted from each sub-image where
the first 8 features are texture features and the remaining 18 features are colour features. The texture features were derived from the discrete cosine transform (DCT) filter of size 3 × 3. The colour features were gathered from the 3-dimensional histogram model of the colour texture from which the statistical description in the form of energy, entropy, local homogeneity, inertia, mean and variance were used. The sample size for both training and test sets were 1000. Because of the high dimensionality of the original measurements of this particular experiment, when the distributions are assumed to be Gaussian the sequential forward floating selection (SFFS) method is used to select features. The use of the branch and bound algorithm is not feasible since it requires large memory and long computational time. The results of the classification with various sizes of feature set are depicted in Table 5.3.

<table>
<thead>
<tr>
<th>Approximation methods</th>
<th>Pe(X6)</th>
<th>Pe(X10)</th>
<th>Pe(X14)</th>
<th>Pe(X18)</th>
<th>Pe(X22)</th>
<th>Pe(X26)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mixture of 2 components</td>
<td>0.354</td>
<td>0.306</td>
<td>0.306</td>
<td>0.306</td>
<td>0.182</td>
<td>0.161</td>
</tr>
<tr>
<td>Mixture of 3 components</td>
<td>0.065</td>
<td>0.024</td>
<td>0.014</td>
<td>0.007</td>
<td>0.007</td>
<td>0.007</td>
</tr>
<tr>
<td>Mixture of 4 components</td>
<td>0.059</td>
<td>0.008</td>
<td>0.007</td>
<td>0.007</td>
<td>0.007</td>
<td>0.007</td>
</tr>
<tr>
<td>Multinormal</td>
<td>0.235</td>
<td>0.287</td>
<td>0.169</td>
<td>0.169</td>
<td>0.169</td>
<td>0.169</td>
</tr>
</tbody>
</table>

Table 5.3: Classifier performance on different feature subset size of image data.

From Table 5.3, it is obvious that the assumption that the distributions are unimodal Gaussian is not appropriate. This results in high classification error rate of various size of feature sets. On the contrary, when the mixture of normal densities is used, very small error rate is obtained except that of the mixture of 2 normal densities. The high classification error rate of the mixture of 2 normal densities simply indicates that the number of only 2 components in the mixture is too small to accurately approximate the distributions. Consequently, a higher number of mixture components is required. As far as feature selection is concerned, our approach works very well since many redundant features have been detected especially with the mixture of 4 components. The higher the number of the mixture components, the better the results of feature subsets of the same size which can be clearly seen with the subsets of 10 and 14 features.

5.5.3 Speech Data

The data used to train and test each approximation model was a set of 1418 pattern vectors of the utterances "YES" and "NO" spoken over the public switched telephone network. Each 15-dimensional feature vector contained 5 segments of 3 features derived by
Chapter 5: Feature Selection in Unknown Distributions

low order linear prediction analysis. From this set, 798 samples were used for training set and 620 different samples for test set. Both sets contain nearly equal number of samples for each pattern class. All data was supplied by British Telecom. The results of the experiment using the mixture model of 2, 3, 4 and 5 components are shown in Table 5.4. Note that due to the small training size, the mixture of 6 components or higher have not been tested and even if better results are obtained with higher number of components, the results cannot be considered reliable.

<table>
<thead>
<tr>
<th>Approximation methods</th>
<th>Pe($X_5$)</th>
<th>Pe($X_7$)</th>
<th>Pe($X_9$)</th>
<th>Pe($X_{11}$)</th>
<th>Pe($X_{13}$)</th>
<th>Pe($X_{15}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mixture of 2 components</td>
<td>0.262</td>
<td>0.224</td>
<td>0.18</td>
<td>0.168</td>
<td>0.146</td>
<td>0.146</td>
</tr>
<tr>
<td>Mixture of 3 components</td>
<td>0.215</td>
<td>0.177</td>
<td>0.17</td>
<td>0.099</td>
<td>0.084</td>
<td>0.089</td>
</tr>
<tr>
<td>Mixture of 4 components</td>
<td>0.186</td>
<td>0.194</td>
<td>0.113</td>
<td>0.085</td>
<td>0.076</td>
<td>0.071</td>
</tr>
<tr>
<td>Mixture of 5 components</td>
<td>0.193</td>
<td>0.2</td>
<td>0.118</td>
<td>0.082</td>
<td>0.067</td>
<td>0.069</td>
</tr>
<tr>
<td>Multinormal</td>
<td>0.076</td>
<td>0.087</td>
<td>0.074</td>
<td>0.073</td>
<td>0.076</td>
<td>0.082</td>
</tr>
</tbody>
</table>

Table 5.4: Classifier performance on different feature subset size of speech data.

From Table 5.4, it can be seen that the mixture of 2 independent normal densities was not sufficient to accurately approximate the unknown distributions. This insufficiency is depicted in the much higher classification error rate than the others, especially when all 15 features were used. When the size of the feature subset was large, slightly better results were obtained with the mixture of 4 and 5 components compared to the multinormal model. However, for smaller size of feature subsets the standard multinormal approach outperformed our approach. The reason is that as already mentioned in section 5.4.2 our approach selects the features which maximize the likelihood function, and it is hoped that these features will be good for discriminating between classes. Obviously, this assumption failed in this particular example. One of the reasons lies in the fact that the size of the training set in this particular example seems not to be sufficient for approximating very well the unknown distributions. A detailed analysis has shown that the data lie in narrow regions along the parallel hyperplanes, i.e. highly correlated data. Therefore, in order to approximate them well by a mixture of independent Gaussian distributions would need obviously still a bigger number of mixture components and correspondingly a bigger training set.
5.6 Conclusions

We have developed a feature selection method based on approximating the unknown distributions by a finite mixture of the densities of a product type using the EM algorithm. The empirical results demonstrate that our approach can be superior to the feature selection method using a multinormal model. A higher number of mixture components is expected to perform better because of the larger flexibility in fitting feature vector components with different widths of the individual peaks (components). However, the training set has to be of reasonable size compared to the number of parameters of the mixture components to be estimated.

In conclusion, we have shown that it is possible to perform feature selection despite the fact that the distributions are unknown. This allows a significant improvement in statistical classification of some real data set with a complex pattern space. However, despite the previously described advantages, we have come to the conclusion that the feature selection based on approximating unknown distributions by a finite mixture of the densities of a special type can be further improved. This idea will be discussed in Chapter 6.

References


Chapter 6

Conclusions and Future Work

6.1 Conclusions

The primary objective of this research was to develop a fast method for solving the problem of feature selection in high dimensionality, which is sometimes known as large-scale feature selection problem. In this case, the optimal solution which can be obtained from either the exhaustive search or the branch-and-bound search (if the monotonicity condition is satisfied) is computationally infeasible. Consequently, other methods which are computationally less demanding have to be considered instead. Of course, with these methods the optimality of the solution can no longer be guaranteed.

The feasibility of various methods suitable for high-dimensional feature selection have been investigated both theoretically and experimentally. As far as the computational time is concerned, it was found that the Max-Min method uses the least computational time. Unfortunately, its performance is relatively poor compared to other well known methods. The relatively poor performance of the Max-Min method is mainly due to the limited information used when a new candidate is considered since the evaluation of feature effectiveness is done mainly in 2-dimensional space. However, it is hoped that if the performance of the Max-Min method can be improved then the modified method would be an ideal method for feature selection in high dimensionality.

Obviously, the performance of the Max-Min method can only be improved after its weaknesses have been found. Our initial investigation has discovered many weaknesses of the Max-Min method such as the incapability to distinguish between poor features and correlated features, the use of the minimum of the upper bound, which is generally very loose, as the measure of the increase in information when a candidate feature is considered, etc. Basing on these findings various modifications have been made to
the Max-Min method in an attempt to improve its performance. Unfortunately, no significant improvement has been achieved. This has lead to a further investigation in to the theory underlying the Max-Min method. The result of this investigation, which is the most important finding of all, is the invalidity of the theory underlying the Max-Min method. Therefore, this indicates that the Max-Min has a little theoretical support and consequently we may not expect good results in general.

As our research has proved, the Max-Min method will certainly not be applicable in general. Consequently, the other suboptimal methods previously mentioned in Chapter 3 have to be reconsidered. Of these methods, there are 4 methods which are relatively simple and require small computational time. These methods are the sequential forward selection (SFS), sequential backward selection (SBS), sequential forward floating selection (SFFS), and sequential backward floating selection (SBFS). In the following, some recommendations for solving the problem of large-scale feature selection using these methods is given.

After obtaining a full set of measurements, it is suggested that the SFS method is first employed. Although the decision about including a particular feature at the first few of the SFS method is made on the basis of the statistical dependencies among features in spaces of low dimensionality, it has proved in practice to be reliable in case of a limited training size. The reason for this appears to be that the algorithm utilizes at the beginning only less complex mutual relations as opposed to the SBS method. Since in the case of a small sample size the training set is obviously not sufficiently representative with respect to the basic data set, any complex mutual relations are not always determined reliably and this is reflected in the performance of the SBS algorithm.

Furthermore, since the SBS method starts at full dimension $D$, where in case of Gaussian distribution the inversion of the matrix $\Sigma_i$ is computed, it may happen that due to linear dependencies of features this matrix is singular. This generally means numerical instability. Hence, its inversion cannot be computed and the process will be terminated prematurely unless one of the features is eliminated. On the other hand, the SFS method in such a case rejects a feature, the adding of which would lead to singularity, and the selection process will continue until its completion. Thus from purely numerical reasons, to begin with the SFS method is preferable in those cases when singularities in matrices can be expected. The SFS method ends in this case in a maximum dimensionality for which the matrix is regular, the SBS method can then be utilized.

It was mentioned earlier that both the SFS and SBS methods suffer from the nesting effect. Therefore, if better solution is required then the SFFS and SBFS should be
employed. However, when very little is known about the data set and some dependent components and consequently the singularity of the matrix $\Sigma_i$ is to be expected, the SBFS method should not be used prior to finding the maximum dimensionality for matrix regularity. It should be noted that the better solution from both the SFFS and SBFS methods is obtained at the expense of longer computation time. Moreover, although these two methods do not suffer from the nesting effects the solution obtained is not guaranteed to be optimal.

When deciding which of the feature search methods should be used in a particular task, one must be aware of their differences both from the theoretical and practical point of view. Because, in practice, we are primarily interested in making $D$ as large as possible and $d$ as small as possible, the ratio $d/D$ is usually not greater than a half though it depends on the particular application. It is obvious that in terms of computation, bottom up search is always to be preferred over top-down.

In the case that unknown distributions are not Gaussian, the methods mentioned above may fail if the patterns are not well separated in the measurement space. The alternative is to use the proposed feature selection method based on approximating the unknown distributions by a finite mixture of the densities of the special type. This approach has proved very useful on some high-dimensional real data with complex pattern space but at the expense of long computational time and the requirement of a large training set. This is because the method has to optimize the approximations of the unknown distributions before the feature orderings can be made. The computational time required by an optimization process depends largely on the dimensionality of the feature space, the size of the training set, and the complexity of the data. However, this is compensated by the fact that the optimal feature subset of required dimensionality is obtained immediately after the optimization ends, as opposed to the former methods.

### 6.2 Future Work

As our research has proved, the Max-Min method is not applicable in general. However, its computational feasibility is so appealing that it is worth directing future research at finding (if possible) the conditions under which the use of the Max-Min method would be theoretically justified. It is still hoped that the basic premise governing the whole Max-Min method, though proved not to be valid in general, may be valid at least under some restricting conditions. If such conditions would be found, we could test their fulfillment in any particular application and in the positive case to use the Max-Min method or its improvements proposed in this thesis. It would mean that we would effectively restrict
the use of the Max-Min method to a certain class of problems for which its use could bring good and justifiable results. In such a case, all the research work which has been carried out with the aim to improve the Max-Min method would prove to be fruitful, despite somewhat negative results achieved till now.

Recently, an optimization method known as simulated annealing technique has been successfully applied in many optimization problems. The simulated annealing takes its inspiration from statistical mechanics and the metallurgical technique of “annealing” — hardening a metal by slow cooling. The method attempts to produce a quality solution through nondeterministic hill-climbing. Since feature selection is basically an optimization process, it would be very interesting to see the performance of simulated annealing technique in solving large-scale feature selection problem.

It has been mentioned in the previous chapter that the feature selection based on approximating unknown distributions by a finite mixture of the densities of a product type selects features that maximize the likelihood function. This is equivalent to minimizing the Kullback-Leibler distance between the true class conditional densities and the postulated class densities. However, the results of the experiment have explicitly shown that these features are not necessarily the most discriminative features. Hence, it is suggested that our feature selection method can be further improved if a criterion function such as probabilistic distance measure is used instead of the likelihood function. However, this probabilistic distance measure is quite complicated unless the closed form solution of the distance measure for the mixture of components is used. Unfortunately, the derivation of the closed form solution for the mixture of components is not as simple as in the case of Gaussian distribution. Despite this fact, future research in this direction is very desirable since the derivation of a probabilistic distance measure based on the approximation of unknown distributions would mean a great achievement. The reason is that it would make it possible to extend the usability of criteria based on a probabilistic distance measure even for distributions of a general form, as opposed to the current applicability to Gaussian data only.
Appendix A – Test Images

Figure A.1: The two colour images used in the experiment in Section 5.5
(a) bianco castilla (b) rosa baveno.