Pattern Classification using Novel Order Statistics and Border Identification Methods

By

Anu Thomas

A thesis submitted to
the Faculty of Graduate and Postdoctoral Affairs
in partial fulfilment of
the requirements for the degree of
Doctor of Philosophy

Ottawa-Carleton Institute for Computer Science
School of Computer Science
Carleton University
Ottawa, Ontario

April 2013

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-94546-9

Our file Notre référence
ISBN: 978-0-494-94546-9

To My Parents

T.V. Thomas and Mercy Thomas

ABSTRACT

The basis for *statistical* pattern classification is that the individual classes are characterized by their *distributions*. These distributions have numerous indicators such as their means, variances etc., and these have, traditionally, played a prominent role in achieving pattern classification. The gold standard for a classifier is the condition of optimality attained by the Bayesian classifier. Within a Bayesian paradigm, if we are allowed to compare the testing sample with only *a single* point in the feature space from each class, the *optimal* Bayesian strategy would be to achieve this based on the (Mahalanobis) distance from the corresponding means. Apart from the indicators mentioned above, a distribution has many other characterizing indicators, for example, those related to its Order Statistics (OS). The interesting point about these indicators is that some of them are quite unrelated to the traditional moments themselves, and in spite of this, have not been used in achieving PR. The main question that we shall consider in this thesis is whether these indicators/indices possess any potential in PR. The amazing answer to this question is that OS can be used in PR, and that such classifiers operate in a completely "anti-Bayesian" manner.

In this thesis, we introduce the theory of optimal PR using the OS of the features rather than the distributions of the features themselves. Our novel methodology, is referred to as Classification by Moments of Order Statistics (CMOS). This claim has been proven for many uni-dimensional and multi-dimensional distributions within the exponential family namely the Uniform, Doubly-exponential, Gaussian, and the theoretical results have been verified by rigorous experimental testing. We have also extended these results significantly by considering asymmetric distributions within the exponential family like the Rayleigh, Gamma, and Beta distributions, for which a near-optimal accuracy has been achieved. The results have also been extended for the corresponding multi-dimensional distributions, and to yield Prototype Reduction Schemes (PRS) which contain only a single element for each class. Apart from the fact that these results are quite fascinating and pioneering in their own right, they also give a theoretical foundation for the families of Border Identification (BI) algorithms.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank *God, the Lord Almighty*, for being with me all through my life and showing me the right way, and giving me the strength, courage and wisdom to complete this research. Being my Savior and Lord, He has been in command of all the dimensions of my life. At this time, I can say that it is only because of His abounding grace that I could complete my work successfully. I bow my head before the Almighty with fear and respect. His grace was and is really sufficient for me.

I express my sincere gratitude to my supervisor *Dr. John Oommen* for the countless hours he has dedicated for my work without which this work would not have been fruitful. He is not only my Supervisor but has also been a role model for me for the last four years. He has not only supported me with the technical concepts needed to do world-class research, but has also provided life-long lessons which concern the nature and scope of fundamental human qualities like generosity, kindness, loyalty, patience, responsibility, humility and courage. He has always stood beside me in my "ups-and-downs", and in my happy and sad moments, just as he would have done for his daughter. As a perfect guide, I should say, in every respect, Dr. Oommen has really been an inspiration and encouragement for me, and I will be grateful to him for ever.

I would like to thank *Dr. Douglas Howe*, Director, *Dr. Doron Nussbaum*, Graduate Director of School of Computer Science, and the staff, especially *Anna* and *Sharmi* for all the help and support they have extended to me during these years. I am grateful to all my friends in my lab and the School for their friendship and valuable support, especially during the early days of my program.

My immediate family was the greatest support during all these years. The presence of my husband *Sam George*, and my children *Allen* and *Ann* (my all-in-all and the greatest gifts from God) have made my life complete. They have adjusted with my busy schedule comprising of weekday and weekend studies, that often extended late

into the night. Without their love, support and encouragement, I could not even think about the completion of this work. I express my sincere thanks to them.

At this time, I would like to extend my deepest gratitude to my parents, *Thomas* and *Mercy*, and would like to dedicate this thesis to them. Their love, care, support and prayers were with me not only for this work but also during every single moment of my life. My brother *Dr. Aju Thomas* has also been a constant support for me. I would also like to thank my parents-in-law, *George* and *Thankamma*, for their support and prayers.

Once again, I would like to thank every single person who has helped me directly or indirectly in completing my research.

Contents

List of Figures

ACRONYMS

ABBI Anti-Bayesian Border Identification

Acc Accuracy

AUC Area Under the Curve

BD Bhattacharyya distance

BI Border Identification

CM Confusion Matrix

CMOS Classification by Moments of Order Statistics

CNN Condensed Nearest Neighbor

ENN Edited Nearest Neighbor

FN False Negative

FP False Positive

LDC Linear Discriminant Classifier

MD Mahalanobis distance

NB Naive Bayes

NN Nearest Neighbor

OS Order Statistics

PBS Progressive Border Sampling

PNN Prototypes for Nearest Neighbor

PR Pattern Recognition

PRS Prototype Reduction Schemes

PS Progressive Sampling

QDC Quadratic Discriminant Classifier

RNN Reduced Nearest Neighbor

ROC Receiver Operating Characteristics

SNN Selective Nearest Neighbor

SOM Self Organizing Map

SVM Support Vector Machine

TN True Negative

TP True Positive

VQ Vector Quantization

NOTATIONS

Variables, Symbols, and Operations

$a \leftarrow b + 1$	The term $b + 1$ is assigned to the variable a.
$\mathbf{x}$	Random variable which takes on a specific value, say x.
$\mathbf{X}$	Random *vector* which takes on a specific value, say X.

Vectors and Matrices

X	A column vector.
$\mathbf{A}$	Matrices utilize boldface.
$\mathbb{R}^d$	d-dimensional vector space, in which each component is a real number.
X^T	Transpose of the vector X.
$\|X\|$	Euclidean norm of the vector X.

Sets

$\mathbb{R}$	The set of real numbers.
$\mathbb{N}$	The set of natural numbers.
$\|\mathcal{X}\|$	The number of elements contained in the set $\mathcal{X}$.

Probability, Distributions and Complexity

ω	State of nature.
$p(\cdot)$	Probability density function.
$\mathbf{w}$	Weight vector.

Chapter 1

Introduction

1.1 Introduction

Pattern classification is the process by which unknown feature vectors are categorized into groups or classes based on their features [10]. The age-old strategy for doing this is based on a Bayesian principle which aims to maximize the *a posteriori* probability. It is well known that when expressions for the latter are simplified, the classification criterion which attains the Bayesian optimal lower bound often reduces to testing the sample point using the corresponding distances/norms to the *means* or the "central points" of the distributions. In short, if we are allowed to compare the testing sample with only *a single* point in the feature space from each class, the *optimal* Bayesian strategy would be to achieve this based on the (Mahalanobis) distance from the corresponding means.

The basis for *statistical* pattern classification is that the individual classes are characterized by their *distributions*. These distributions have numerous indicators such as their means, variances etc., and these indices have, traditionally, played a prominent role in achieving pattern classification, and in designing the corresponding training and testing algorithms. It is also well known that a distribution has many other characterizing indicators, for example, those related to its Order Statistics (OS). The interesting point about these indicators is that some of them are quite unrelated to the traditional moments themselves, and in spite of this, have not been used in

achieving PR. The main question that we shall consider is whether these indicators and indices possess any potential in PR.

We can consider the problem that we are investigating from an alternate perspective. Consider the problem of merely using a subset of the original training set to perform a classification. Such a paradigm falls within the family of the so-called Prototype Reduction Schemes (PRS). A PRS algorithm is a generic method for reducing the number of training vectors, while simultaneously attempting to guarantee that the classifier built on the reduced design set performs as well, or nearly as well, as the classifier built on the original design set[1]. Typically, the algorithm selects a subset of the entire set based on certain criteria, and the learning (or training) is then performed on this reduced training set, which is also called the "Reference" set. This reduced set is not constrained to only involve border patterns, but contains patterns that have the potential to represent the entire training set.

Within the community of those who have studied PRSs, some researchers have attempted to perform the classification by examining the testing samples with respect to certain patterns that lie close to the boundary of the classes. These approaches are categorized as being Border Identification (BI) algorithms. It has been reported [33] that the Reference set obtained by a BI approach is not sufficient to perform an optimal classification as it contains only the so-called "Near" borders of the data points. In other words, these patterns, by themselves, will not be able to correctly classify the test patterns that are closer to the class boundaries. Recently, researchers like Li *et.al.* [33] tried to solve this problem by augmenting the "Near" borders with a set of "Far" borders, to collectively constitute the Reference set, i.e., a "Full" border set. Further, these authors have defined a stopping criterion for this process of obtaining a "Full" border set that can perform a near-optimal classification.

With all these areas as a background, the questions that are of primary concern to us are the following:

- Can indicators like the OS, which traditionally have not been used in PR, be utilized in solving PR problems?

[1]These algorithms have also been referred to as "Instance Selection" algorithms in the Machine Learning literature.

- If the OS can be used, how would one design the training and testing phases of an OS-based PR system?
- Can any OS-based system provide a good classifier that will attain a reasonable accuracy?
- Can we use an OS-based philosophy in such a way that it will guarantee an almost-optimal PR system?
- Can we incorporate an OS-based philosophy to design PRS algorithms?
- Can we propose BI algorithms that inherently apply an OS-based paradigm?
- The issues that one encounters in such a setting are many and fairly complex. To understand them, the reader must observe that the classification with respect to the Bayesian paradigm is, typically, based on the central points of the distributions. However, if we consider OS-based classification, one could be dealing with the data points of the classes which are non-typical representatives, and which occur with a small probability. In other words, is it possible to build near-optimal PR systems, get PRSs, and BI algorithms which only involve *certain* non-typical members of the distributions?

Without an intensive analysis, we can get a clue that the *extreme* outliers, in and of themselves, will not be able to yield an optimal classification system, as they will not be able to classify the points that are in the proximity of the class boundaries. The question then is whether we are able to find *other* non-typical members that have sufficient information so as lend themselves to reasonable, or even optimal classification. In other words, are there points in a distribution that are distant from the central points, and that are yet not "outlier"-enough to assist in determining such classifiers? The amazing answer to this question is in the affirmative.

The problem with using a Reference set obtained by this approach is that it is relatively large. The "ambitious" question that we attempt to tackle is the following: Can we obtain a "small" (i.e., for example as small as two patterns, one from each class for uni-dimensional 2-class problems) border-based Reference set, which can lead

to an optimal or near-optimal Bayes' classification. The answer to this interesting question is, again, in the affirmative.

The result that we shall show is the following: The patterns that we have determined to select are away from the central points of the distributions, and are yet not on the exact boundary of the classes. We have rather attempted to select those patterns which are close to the boundary of the classes, and that are not in the region which overlaps the two classes. In other words, our methodology has essentially extended the definition of the concept "border".

How then can we relate the OS-based classifiers to the border-based algorithms, and thus "close the loop"? To initiate the discussion about this, we mention that the gold standard for a classifier is the condition of optimality attained by the Bayesian classifier. In this thesis, we shall show that we can obtain optimal results by operating in a diametrically opposite way, i.e., a so-called "anti-Bayesian" manner. Indeed, we shall show the completely counter-intuitive result that by working with a *very few* (sometimes as small as two) points *distant* from the mean, one can obtain remarkable classification accuracies. Further, if these points are determined by the *Order Statistics* of the distributions, the accuracy of our method, referred to as Classification by Moments of Order Statistics (CMOS), attains the optimal Bayes' bound. This claim, which is totally counter-intuitive, has been proven for many uni-dimensional, and some multi-dimensional distributions within the Exponential family, and the theoretical results have been verified by rigorous experimental testing. It turns out, though, that this process is computationally not any more complex than working with the latter distributions.

Apart from the fact that these results are quite fascinating and pioneering in their own right, they also give a theoretical foundation for the families of BI algorithms reported in the literature.

1.2 Motivations and Objectives

By virtue of the backdrop of OS, BI and PRS algorithms that were briefly discussed above, we now state the motivations and objectives of the thesis.

1.2.1 Motivations

A significant amount of work has been done to reduce the number of training vectors without reducing the accuracy of the classification. The BI algorithms are the result of this endeavor, but as the patterns found in the Reference set, in and of themselves, are not able to yield an optimal accuracy, these algorithms cannot be considered as stand-alone schemes for a classification problem. Our goal is to operate with an enhanced version of the term "border" and to then design a BI algorithm that can attain almost-optimal Bayes' bound. In order to motivate the thesis, we list below the following issues:

- To the best of our knowledge, there is no theoretical background for the present day BI algorithms. The first motivation of this work is to see whether we can provide a theoretical framework for the family of BI algorithms by incorporating the concepts concerning the moments of the Order Statistics.

- "Borders", as defined in current-day BI algorithms, are not sufficiently informative so as to perform optimal classification. In this context, our second motivation is to update the definition of the term "border" so that the border set can provide near-optimal results for classification problems.

- As we intend to provide a theoretical framework for the families of BI algorithms, we need to determine whether this new paradigm will work with most distributions within the Exponential family.

- Traditional PRSs reduce the cardinality of the Reference set. We would like to see if we can design an "ultimate" PRS algorithm that reduces the cardinality of the Reference set to be unity for each class.

- The next motivation of this thesis is to see whether the new paradigm can be applied for real-life data sets so as to achieve near-optimal classification.

This summarizes the motivation of our thesis.

1.2.2 Objectives

With the motivations listed above serving as a beacon to develop a new philosophy that can attain near-optimal Bayes' classification, the objectives of this work are the following:

1. First of all, we would like to provide a theoretical framework for adequately responding to the question of why the border points are informative for the task of classification. To achieve this, we intend to make use of the concept of Order Statistics and its moments.

2. We would like to develop an "Anti-Bayesian" *Parametric* PR strategy that uses the OS criteria, that can attain almost-optimal classification accuracy as that of the Bayes' bound.

3. We would like to design PR algorithms based on the moments of the OS for various uni-dimensional distributions in the Exponential family.

4. We would like to design PR algorithms based on the moments of the OS for the multi-dimensional versions of the distributions studied above.

5. We would like to generalize the new paradigm so as to achieve an "ultimate PRS".

6. We would like to demonstrate that these "ultimate PRSs" can be used for real-life data sets, where the specific PRS can be of either a "selective" or a "creative" PRS sort.

7. We would like to design a PRS algorithm where the classification is based on a single value of a single feature.

8. We would like to design new families of "selective" BI algorithms based on the above-mentioned OS criteria.

1.3 Issues Not Visited in this Thesis

Although the thesis is fairly comprehensive, in that it introduces and explains the OS and some of the domains where they can be used, clearly, there is still a lot of work that remains to be done. In particular, we have not visited the following issues when it concerns OS-based PR and classification:

- The clustering of data;
- The training and testing of unsupervised data sets using OS-based schemes;
- The training and testing of OS-based classifiers when the data sets are unbalanced;
- The classification of data which inherently have more than two classes, although we remark that this problem can be trivially solved using a "one-*versus*-others" mapping onto a two-class problem domain.

We believe that the problems associated with all these areas are currently open, and represent avenues for future research.

1.4 Organization of the Thesis

Chapter 2: In the interest of completeness, this survey chapter presents a brief overview of the various phases of a PR system including feature extraction, training, classification, testing and system evaluation, and a brief analysis about a classifier's accuracy, the Receiver Operating Characteristics (ROC) curve, and the Area Under the Curve (AUC). It also provides a clear overview on the various types of classifiers such as a Nearest Neighbor classifier, the Parzen window, and the various linear, quadratic and Fisher's discriminant classifiers, the Support Vector Machine (SVM) and the family of decision trees, which are currently in use.

The chapter also presents a comprehensive survey of the concept of PRSs [17, 54] and various traditional and present day BI algorithms. Various PRS algorithms

such as Condensed Nearest Neighbor (CNN) rule [20], the Reduced Nearest Neighbor (RNN) rule [18], the Prototypes for Nearest Neighbor (PNN) classifiers [3], the Selective Nearest Neighbor (SNN) rule [40], the Edited Nearest Neighbor (ENN) rule [6], Vector Quantization (VQ) etc. are thoroughly studied in this chapter. Thereafter, the chapter discusses the difference between prototypes and border patterns, and continues to provide a detailed study of the families of BI algorithms, which, in turn, can be considered to be a subset of the PRS in which the reduced set contains only the border patterns. The chapter contains a formal description of the traditional BI algorithms such as Duch's first and second approach and Foody's approach, and the other more-recent BI algorithms including Li's "Border Identification in Two Stages" and "Progressive Border Sampling" schemes. A brief survey on the concept of Classifier fusion and of ensemble fusion techniques are also provided in this chapter.

Chapter 3: This chapter presents a novel approach to the age-old problem of pattern classification, namely, by using a non-traditional "anti-Bayesian" approach. We show that the optimal Bayes' bound can be obtained by such an "anti-Bayesian" strategy, which we refer to as Classification by Moments of Order Statistics (CMOS). To be more specific, we prove that the classification can be attained by working with a *very few* (indeed, sometimes two) points *distant* from the mean. Further, if we determine these points by the *Order Statistics* of the distributions, the method can attain the optimal Bayes' bound. After proving some fundamental results concerning OS, we have derived the conditions for a "generic" uni-dimensional classifier. The claim has then been proven for the uni-dimensional Uniform distribution. The reason for considering the Uniform distribution is that even though the distribution *itself* is rather trivial, the analysis will provide the reader with an insight into the mechanism by which the problem can be tackled, which can then be extended for other distributions within the Exponential family. The theoretical results have been verified by rigorous experimental testing.

Chapter 4: This chapter provides a thorough investigation of the application of CMOS on various uni-dimensional distributions in the Exponential family. The detailed study demonstrates that the CMOS can attain optimal Bayes' bound for various

symmetric distributions like the Doubly-Exponential, Gaussian, and some Beta distributions. The chapter also provides the theoretical analysis and experimental results that compare the efficiency of the newly proposed schemes with the Bayes classifier, and which show the strength of the proposed method.

Chapter 5: This chapter extends the application of CMOS for various asymmetric distributions namely Rayleigh, Gamma, and some other Beta distributions in the Exponential family. The study shows that CMOS can attain near-optimal classification for these asymmetric distributions. Theoretical analysis and experimental results are also provided for all these distributions.

Chapter 6: This chapter deals with the application of CMOS for multi-dimensional distributions. It provides a theoretical analysis and the experimental results for two-dimensional Uniform, Doubly-Exponential, Gaussian and Rayleigh distributions, and also paves the way to proceed when it concerns other higher-order multi-dimensional distributions. The theoretical analysis and experiments presented show that the CMOS can attain the optimal Bayes' bound for symmetric multi-dimensional distributions, and near-optimal classification for asymmetric multi-dimensional distributions.

Chapter 7: This chapter proposes the "Ultimate" PRS which is obtained by an anti-Bayesian OS-based criteria. We derive single-element PRSs which are either "selective" or "creative", where in each case we present a non-parametric or a parametric paradigm respectively. We test all the methods on artificial sets and on real life data sets which are taken from the UCI Machine Learning Repository [15].

Chapter 8: This chapter suggests novel BI algorithms which are actually formulated based on the concept of OS. We provide a new definition for the term "border", and design new algorithms based on this definition. The algorithms proposed are tested on both artificial and real-life data sets, and we show that we can obtain accuracy comparable to the best reported schemes, with the newly defined "border" values.

Chapter 9: This chapter concludes the thesis.

Chapter 2

Literature Review

In our introductory chapter, Chapter 1, we had motivated the thesis and this research endeavor, namely to see how the concepts of borders and their identification could be used in PR systems. Our hypothesis is that the phenomenon of borders can actually be utilized in almost *every* aspect and module of a PR system. Thus, even though the concepts of feature extraction, feature selection, training, testing and classifier evaluation are well established, the question of how border "samples" can influence each (or all) of these stages is one of the primary questions which we shall examine. Even though the preliminary aspects of PR are well developed, to pose our problems in the right perspective, we shall first give a brief overview of a general PR system. Much of this material can appear elementary; however, while reading any part of this overview, the reader must remember that it is written with the ultimate goal of understanding how the concept of "borders" can be used to enhance the phase currently being described. While this thesis only examines the use of borders in a few of these areas (for example, training and classification), a futuristic goal of the research is that it can also lead to novel techniques for feature selection and modeling.

A "border" is a fairly generic term. For example, between two countries, in the simplest sense, the border will define their mutual boundaries. But if the people who lived in these countries belonged to different racial groups, the "border" could be considered to be the "limiting" facial, genetic or other biological discriminating aspects which differentiated the peoples within the countries.

We believe that this analogy also pervades to the field of PR. Thus, in PR, given two classes, in the most simplistic sense, the border could consist of the samples that could help in their mutual discrimination. But we suggest that this concept can be abstracted to much higher and finer levels. Indeed, in one sense, the borders could be the *types* of measurements used in a PR system, and in a more abstract sense, a feature selection methodology can be aptly modeled as a BI algorithm.

With this in mind, we present the following sections which give a *brief* overview of the field of PR.

2.1 Salient Aspects of Pattern Recognition

The goal of a PR system is to classify the objects or data (patterns) into a number of classes based either on the *a priori* knowledge of the patterns or on the statistical[1] information extracted from them. It is an integral part of most machine intelligence systems built for decision making. A pattern can be a handwritten word, a fingerprint, a DNA sequence or a musical note [45]. PR has a wide range of applications which include computer vision, character recognition, speech recognition, robotics, computer aided diagnosis, data mining and knowledge discovery etc. In short, the field of PR is concerned with the automatic discovery of regularities in data through the use of computer algorithms, and the intention is that one is able to classify the data into different categories by using these regularities [2]. In the interest of completeness, we present in Table[2] 2.1, a summary of some of the applications of PR in various application domains.

[1]This thesis only deals with statistical PR. The concept and use of "borders" in the fields of syntactic and structural PR are yet unexplored.

[2]This table is obtained from:
`http://www.cs.bilkent.edu.tr/~saksoy/courses/cs551/slides/cs551_intro.pdf`.

Problem Domain	Application	Input Pattern	Pattern Classes
Document image analysis	Optical character recognition	Document image	Characters, Words
Document classification	Internet search	Text document	Semantic categories
Document classification	Junk mail filtering	Email	Junk/non-junk
Multimedia database retrieval	Internet search	Video clip	Video genres
Speech recognition	Telephone directory assistance	Speech waveform	Spoken words
Natural language processing	Information extraction	Sentences	Parts of speech
Biometric recognition	Personal identification	Face, iris, fingerprint	Authorized users for access control
Medical	Computer aided diagnosis	Microscopic image	Cancerous/healthy cell
Military	Automatic target recognition	Optical or infrared image	Target type
Industrial automation	Printed circuit board inspection	Intensity or range image	Defective/non-defective product
Industrial automation	Fruit sorting	Images taken on a conveyor belt	Grade of quality
Remote sensing	Forecasting crop yield	Multi-spectral image	Land use categories
Bioinformatics	Sequence analysis	DNA sequence	Known types of genes
Data mining	Searching for meaningful patterns	Points in multidimensional space	Compact and well-separated clusters

Table 2.1: Some applications of Pattern Recognition.
(Taken from `http://www.cs.bilkent.edu.tr/~saksoy/courses/cs551/slides/cs551_intro.pdf`.)

The rest of this chapter is organized as follows. We shall first present a brief overview of the various phases of a PR system including feature extraction, training, classification, testing and system evaluation. A brief analysis about a classifier's accuracy and the Receiver Operating Characteristics (ROC) curve are also provided, as these will be used in quantifying the efficiency of the algorithms that we will develop. As Prototype Reduction Schemes (PRS) are the well-known methods for reducing the number of patterns used in the training of a classifier, a detailed study of various PRS schemes is included in Section 2.2. This is because we will later achieve a comparative study of our methods against some typical PR schemes. Since we are primarily interested in the modeling and effect of "border" patterns, reduction schemes that have been developed based on the concept of borders are thoroughly examined in Section 2.3. This contributes the state-of-the-art in the science of BI. Also, a strategic scheme by which we shall utilize BI methods, can be devised by projecting the d-dimensional vectors on to a lower dimensional (for example, a 2-dimensional) subspace. Thus, BI methods which involve various two-dimensional classifiers can be "fused" to lead to the corresponding d-dimensional results. With this as a pretext, a brief survey about classifier fusion, and of ensemble fusion techniques is included in Section 2.4.

A PR system, at its front end, possesses a **sensing** module, whence the raw data, for example, an image or a sound signal, is obtained from a transducer. This module is followed by a **pre-processing** unit whose task is to process the data "files" so as to simplify the subsequent operations, without simultaneously losing any pertinent or relevant information. The **feature extraction** phase is the one in which *measurable* quantities are extracted from the preprocessed data. Next comes the **training** phase in which a classifier is built based on the properties of the feature vectors. **Classification**, which is the goal of the whole exercise, is the process by which unknown feature vectors are categorized into different classes. The final phase, which is the **system's evaluation**, involves testing the classifier by evaluating its accuracy and efficiency. The basic stages involved in a PR system are depicted in Figure 2.1.

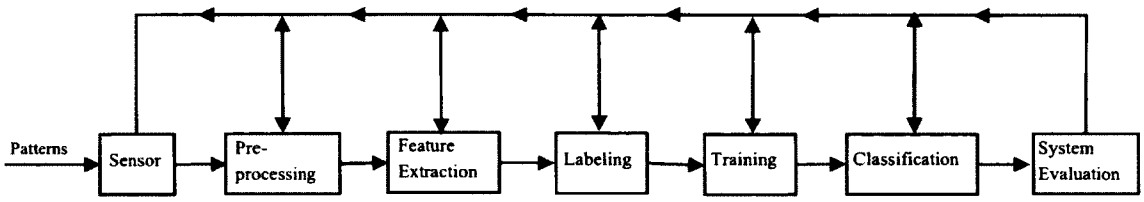


Figure 2.1: Stages of a Pattern Recognition system in which each component is allowed to receive feedback from the final classification module.

2.1.1 Sensing

The input to a PR system will be provided by a set of transducers specific to the application domain. The task of the sensor is to convert the representative signals of the real-life entity (for example, the fingerprint, human organ etc.) into quantities by which inferences can be made.

2.1.2 Pre-processing

The goal of a pre-processing phase is to identify the meaningful components of the real-life entity, and to process them so as to simplify the subsequent operations without the loss of pertinent relevant information. A review of the pre-processing operations and approaches typically done is found in [59], and it is our hope that BI methods can, in a future work, be used to enhance this phase.

2.1.3 Feature Extraction

In statistical PR, objects are represented by feature vectors. This implies that although an object is a complex entity in its own right, the relevant information about it can be formally represented as a vector of d measurements. A feature extraction phase is required to identify the measurable quantities, also known as vectors, that can categorize the objects into different classes. The actual features selected for a specific problem are again domain dependent, and constitute the so called feature vector, $X = [X_1, X_2, ..., X_d]^T$, typically given as a column vector. The hope is that if the features are expediently chosen, a single pattern or object can be uniquely identified by the corresponding feature vector, i.e., a point in a d-dimensional space.

We can summarize that the goal of the feature extractor is to characterize an object to be recognized by consequent measurements whose values are very similar for objects in the same category, and rather distinct for objects in different categories. This leads to the principal task of seeking distinguishing features that are, hopefully, invariant to irrelevant transformations of the input [9].

2.1.4 Learning

Based on the information provided by the obtained feature vectors, the task of a PR system is to "learn" the characteristics of the different classes. This learning process can be classified into three groups – Supervised learning, Unsupervised learning, and Semi-supervised learning.

Supervised Learning

If the class labels for the training data are available *a priori*, a PR system which uses Supervised Learning, attempts to build a classifier based on this available information. The classification methodology can, in turn, be categorized into two groups - parametric and non-parametric methods. Parametric methods model the classes based on assumptions related to the underlying class *distributions*, and the training samples are used to estimate the parameters in these models. On the contrary, the non-parametric methods do not make any assumptions, and the classification is based on the training samples belonging to the nearest neighborhood (for example) of the pattern to be classified. A problem of significant interest in the context of this thesis is one of achieving Supervised Learning when one only uses the *borders* obtained after a BI phase.

Unsupervised Learning

In a real-life scenario, which could involve image coding, remote sensing etc, even though it is customary that a set of feature vectors is provided for the analysis, in many cases, the practitioner will initially not have any *a priori* information about the corresponding class labels. In other words, in a new application domain, it is fair

to assume that one will not be able to acquire *a priori* indication about how these vectors can be grouped. Thus, it is realistic to believe that prior to the classification stage, the practitioner has to unravel the underlying similarities and group the similar vectors together. In this case, the learning is not considered to be "supervised" by the knowledge of *a priori* information, implying that the class labels *and* the discriminating aspects of the classes are learned in an unsupervised manner. We anticipate the possibility of performing unsupervised learning based on the *borders* obtained by the process of BI.

Semi-supervised Learning

In some situations, the set of feature vectors contain data of both types, i.e., some of the vectors enjoy the benefit of having accompanying *a priori* information about their respective class labels, and hence, can be considered as labeled data suitable for a supervised PR system. However, the remaining vectors, and often the bulk of the data, do not include any information about their class membership and is thus, essentially, "unlabeled data", suitable only for unsupervised learning. Typically, the information about the unlabeled data is also provided prior to the supervised training, whence, by applying clustering algorithms on the data, a preliminary class association can be achieved. The labeled data, which is usually more limited, can, for example, serve to yield information about the clusters (for example, pertinent to the *centers* of the clusters), and the unlabeled data, can thereafter be assigned to any of the clusters based on their distances to the feature vectors. A future goal of this work is to formulate how BI can be used for semi-supervised systems.

2.1.5 Training

Subsequent to the learning process described above, the task of a PR system involves the "training" phase, which utilizes the information obtained during learning to yield the classifier that is to classify or recognize the unknown patterns. In any real-life scenario, the efficiency and the accuracy of the classifier must also be inferred; thus, it is necessary for the practitioner to perform both the training and the system

evaluation with *only* the available data. However, if we use the same data patterns for both the training and the evaluation phase, the system is prone to the phenomenon of the classifier's "overtraining". A consequence of this is that it will be able to perfectly classify the available data, yet perform poorly on unseen data. In order to avoid this situation, the science recommends that the data should be divided into two sets, referred to as the "Training" set and the "Testing" (or really the "Pseudo Testing") set, before the actual classification on unknown samples is done. A number of methods have been reported to accomplish this task, among which cross-validation, the hold-out method, resubstitution (R-method) with/without bootstrap, are the well-known techniques.

Cross-validation

The cross-validation method to separate the whole set of patterns into training and testing subsets can be summarized as follows:

- Choose an integer K (preferably a factor of N, the total number of samples);
- Divide the training set T randomly into K disjoint subsets of size N/K;
- Train the classifier using the patterns found in $K-1$ of these subsets;
- Test the corresponding classifier using the patterns in the remaining subset and estimate the quality of the classifier[3];
- Repeat this procedure K times such that every subset is tested against the classifier obtained by training the patterns found in the union of the *remaining* subsets;
- Report the average of the K quality estimates as the overall index of the classifier's accuracy/efficiency.

Since the whole set is grouped *via* K folds, this method is often referred to as the K-fold cross-validation.

[3]The issue of quantifying the *quality* of classifier will be discussed in Section 2.1.7.

Observe that if we set the value of K to be N itself, the training set will contain $N - 1$ patterns and the testing set will contain only a single pattern. Since this involves the omission of a single pattern (i.e., the one that is left out for the validation purposes) this method is often called the "leave-one-out" method.

Hold-out Method

In the so-called "Hold-out" method, the entire set of available patterns is divided into two halves, of which one is used for training purposes while the second set is used for testing. The roles of the sets are later swapped to yield another estimate of the classifier's quality, whence the final index is obtained as the average of both the estimates. Unlike the cross-validation approach, however, the subsets of the items can be now varied from among the $\binom{N}{N/K}$ possible subsets, each of which will lead to another estimate. A version of this method is the "Data Shuffle" in which the whole set of patterns is divided into K random splits for training and testing purposes, and the final quality index is computed as the average of all the K estimates on the respective testing sets [31]. We believe that BI methods are also applicable for evaluating classifiers by means of the hold-out method.

Resubstitution with/without Bootstrap

Resubstitution is a method in which a classifier D is obtained for the whole set of patterns T, and the classifier is tested on the same set. Since both the learning and the testing is performed on the same set of patterns, the result is optimistically biased. Resubstitution without bootstrap leads to the overtraining scenario described above. In order to overcome the drawback of such a bias, researchers have recommended a technique which involves implementing "Bootstrap with resubstitution". The Bootstrap method artificially generates (with replacement) K sets, each of which is based on N patterns from the original set. Learning is performed on each set, and the final result will be the average of the results obtained from all the classifiers. Again, the use of BI to obtain Bootstrap samples is an extremely interesting problem that deserves research.

Grouping into Training, Validation and Testing Sets

If the whole set of available patterns is large, another improvement in grouping these is also recommended. If the data set is large enough to be divided into three groups, then the entire set can be divided in this manner for three different purposes – training, validation and testing. The validation set is for the pseudo-testing, and the actual testing set will remain "unseen" as far as the classifier is concerned. The training phase will proceed until the performance improvement on the training set is no longer matched by a performance improvement on the validation set. At this juncture, the training will be terminated so as to avoid overtraining.

2.1.6 Classification

The next step of a PR system is to apply the obtained classifier on the vectors in order to categorize them into various groups or classes based on their features. After the grouping, the accuracy (i.e., the quality of the classifier) should be determined. There are a variety of classifiers used for classification, among which the most wildly used ones are the Nearest Neighbor classifier, the Parzen window, the various linear, quadratic and Fisher's discriminant classifiers, the Support Vector Machine (SVM) and the family of decision trees. Each of these can potentially have their "border"-based counterparts.

Figure 2.2 (adapted from [31]) presents a schematic of various types of classifiers.

Nearest Neighbor (NN) Classifier

The Nearest Neighbor (NN) method initially identifies the nearest neighbors of the pattern to be classified. Specifically, in order to classify the vector X, the k-Nearest Neighbor (kNN) classifier identifies k vectors of the training set, which are in the closest (depending on the norm used) proximity of X. The classification decision is typically made based on the class labels of the kNNs and by a majority vote.

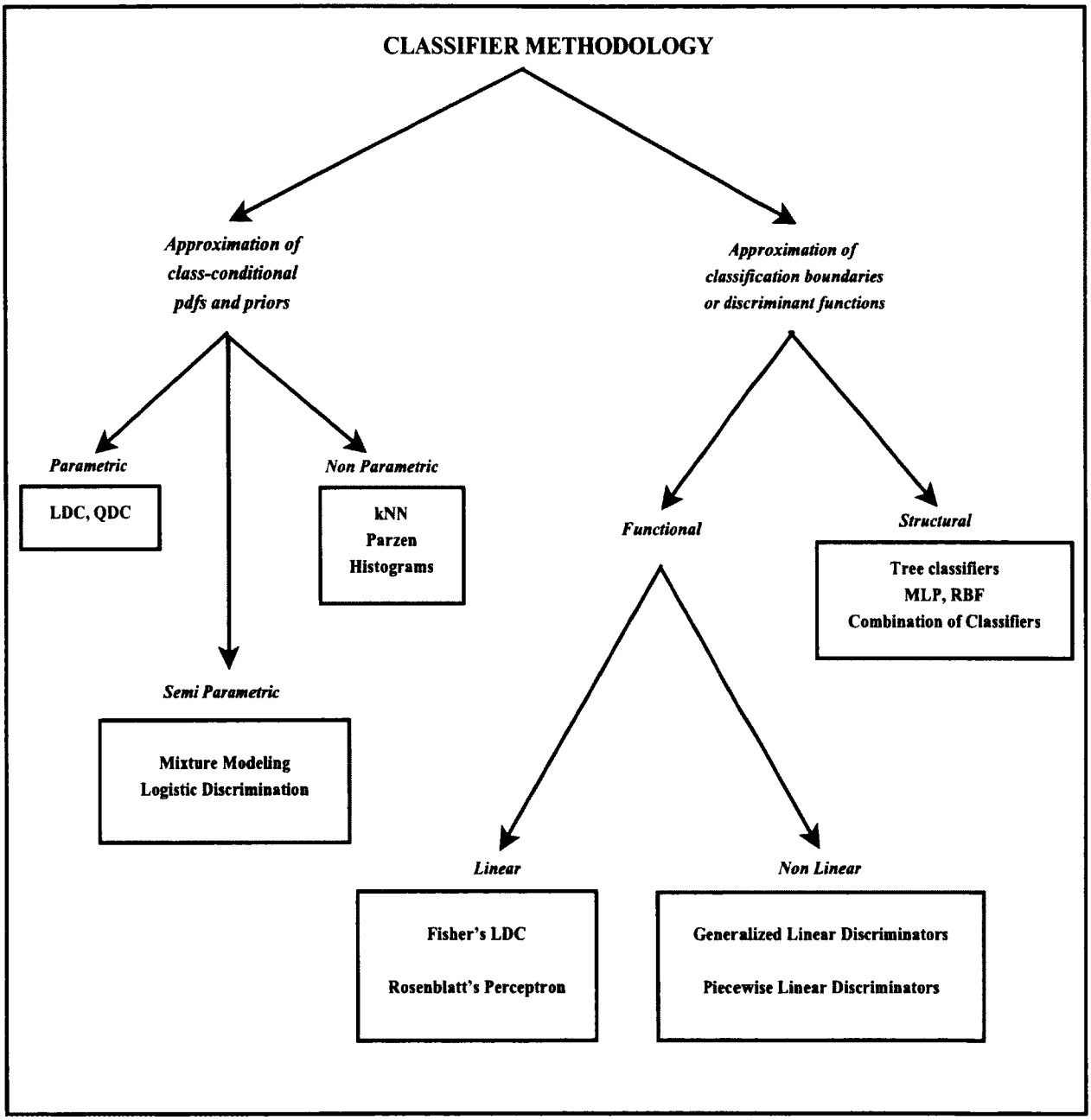


Figure 2.2: A schematic overview of the families of well-known classifiers (Adapted from [31]).

It is obvious that the NN classifier depends upon the distance metric[4] used in this evaluation. Metrics that are traditionally used for the calculation of the distance are the Euclidian metric, the Mahalanobi's metric, the Minkowski metric, and the Tanimoto metric. As a NN classifier does not explicitly require any assumptions on the class distributions, it is accurately described as a non-parametric method for classification.

A central issue that we investigate will be the use of BI methods to efficiently and accurately design NN classifiers that only utilize the "border" points of the respective classes.

Parzen Window

The Parzen window method (named after Emanuel Parzen) is a non-parametric way for determining the membership of a pattern. In this method, the practitioner creates a d-dimensional window around all the training samples, and the estimate of the density function will be based on the patterns that falls into every window.

Let $\mathbf{X}$ be a random vector. The probability density function of $\mathbf{X}$ at X, $p(X)$, can be obtained by placing a window function at X and by determining the contribution of every sample to that window. The summation of all these respective contributions are considered to yield the estimated value of $p(X)$. The Parzen window estimate can thus be defined as below.

$$p(X) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_n^d}K\left(\frac{X - X_i}{h_n}\right)$$

In general, $K(X)$ is a window function or a kernel in the d-dimensional space such that

$$\int_{\mathbb{R}^n} K(X)\,\mathrm{d}X = 1,$$

and h_n is the width of the window that corresponds to the width of the kernel. The

[4]Any metric must satisfy the four properties of non-negativity, reflexivity, symmetry and the triangle inequality.

Gaussian kernel, given below, is the most popular among the kernels for Parzen-window density estimation.

$$p(X) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{h\sqrt{2\pi}}\right)^{d} exp\left(-\frac{1}{2}\left(\frac{X - X_i}{h}\right)^{2}\right).$$

In the above, h is the standard deviation of the kernel Gaussian PDF along each dimension. This method requires the entire training set to serve as the prototype set, which results in poor performance with respect to the time, when the set is large. Another deciding factor of the performance is the selection of 'd' which brings along with it the "curse of dimensionality".

In this regard, an issue that we investigate will be the use of "border" points for the Parzen-window estimation. Even if the estimate of the class conditional density is poor for each class (when compared to the method which uses the entire training set), we believe that it will be good enough to serve for an accurate classification. Thus we propose that the use of border points for the Parzen estimation will serve to enhance the scheme's efficiency.

Support Vector Machine (SVM)

The SVM [4, 55] is a very powerful classification technique developed by Vapnik and his co-authors. The main motivating criterion used by the SVM is to separate the classes with a surface that maximizes the margin between them. The rationale for the method is to obtain an approximate implementation of the structural risk minimization induction principle that aims to minimize a *bound* on the generalization error of a model, rather than minimizing the mean square error over the entire training data set, which is the philosophy that empirical risk minimization methods often use [27].

By way of example, consider the scenario of a two-dimensional training set, that represents two linearly separable classes involving ovals and hexagons as depicted in Figure 2.3. The objective is to build a classifier to separate these classes.

Figure 2.3 displays the classifier by means of a solid line. However, it also includes a pair of dashed lines drawn parallel to the classifier, which in turn, signifies the

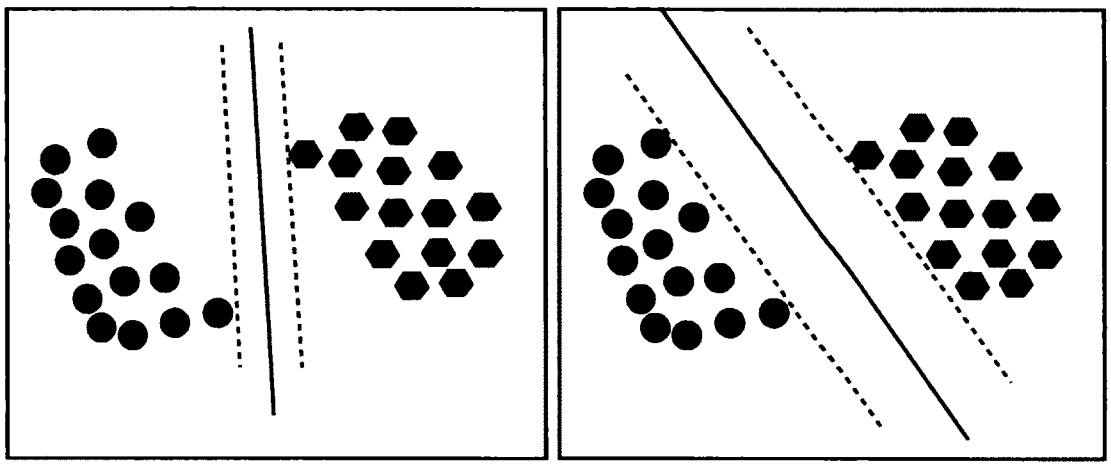


Figure 2.3: The schematic of a Support Vector Machine (SVM).

distance between the linear classifier and the feature vectors that lie closest to it. The distance between the dashed lines is called the "margin", and the vectors that constrain the width of the margin are the so-called "support vectors". The classifier of the second figure has a larger margin when compared to the classifier given in the first.

The objective of a SVM analysis is to find a classifier for which the margin among the classes is maximized. This can be achieved by mapping the samples onto a higher-dimensional space in a non-linear manner [9]. To achieve this, every pattern X_k should be transformed by a function ϕ, and this transformation can be represented as $Y_k = \phi(X_k)$. Also, for every pattern X_k, $k = 1, 2, ..., n$, we set $z_k = \pm 1$, which is to signify the class label, i.e., that the pattern X_k is in ω_1 or ω_2. A linear discriminant in the augmented Y space can be seen to be

$$g(Y) = A^t Y,$$

where both the weight vector and the transformed pattern vector are augmented by $A_0 = W_0$ and $Y_0 = 1$. The separating hyperplane ensures that the condition:

$$z_k g(Y_k) \geq 1, \qquad k = 1, 2, ..., n.$$

As mentioned earlier, the goal of training a SVM is to yield a separating hyperplane with the largest margin. The distance between the hyperplane and a transformed

pattern is $\frac{|g(Y)|}{||A||}$. If there exists a positive margin among the patterns, we can represent it as

$$\frac{z_k g(Y_k)}{||A||} \geq b, \qquad k = 1, 2, ..., n.$$

The goal of the learning is to find the weight vector A that maximizes b. The optimized hyperplane can be obtained by solving the linear optimization problem:

$$\begin{aligned} \text{maximize} \qquad L(A, \alpha) &= \frac{1}{2}||A||^2 - \sum_{k=1}^{n} \alpha_k [z_k A^t Y_k - 1] \\ \text{subject to} \quad \sum_{k=1}^{n} z_k \alpha_k &= 0 \\ \alpha_k &\geq 0 \qquad k = 1, 2, ..., n, \end{aligned}$$

where A is the weight vector, and $\alpha_k \geq 0$ are the undetermined multipliers.

One should observe that the support vectors are the most informative ones for classification, while, at the same time, the most difficult patterns to classify, as they define the optimal separating hyperplane.

A BI-related problem is that of using SVM to identify the best border points. A second one will be that of determining how the best border points can *themselves* influence the corresponding SVM.

Linear and Quadratic Discriminant Classifier(LDC/QDC)

Linear and quadratic classifiers are named as per the type of discriminant functions that they use. If the classifier function is of the general form

$$g(X) = w_0 + \sum_{i=1}^{d} w_i x_i,$$

where w_i are the components of a weight vector W, the classifier is termed to be a *Linear* Discriminant Classifier (LDC) [31]. Any set of discriminant functions obtained by a monotonic transformation from the posterior probabilities, $P(\omega_i|X)$, constitutes an optimal classifier with minimum error as below:

$$g_i(X) = log[P(\omega_i)\ p(X|\omega_i)], \quad i = 1, 2, ..., c,$$

where $P(\omega_i)$ is the prior probability for class ω_i and $p(X|\omega_i)$ is the class-conditional probability density function. The optimal classifier is linear for numerous distribution within the exponential family.

On the other hand, if all the classes are *normally* distributed with means M_i and covariance matrices Σ_i, the above equation can be shown to be

$$\begin{aligned} g_i(X) &= log[P(\omega_i)] + log[\frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{|\Sigma_i|}}\exp[-\frac{1}{2}(X-M_i)^T\Sigma_i^{-1}(X-M_i)] \\ &= log[P(\omega_i)] - \frac{n}{2}log(2\pi) - \frac{1}{2}log(|\Sigma_i|) - \frac{1}{2}(X-M_i)^T\Sigma_i^{-1}(X-M_i) \quad (2.1) \end{aligned}$$

Since Equation (2.1) has quadratic terms, the resultant classifier is a hyper-quadratic in the d dimensional space, leading to a *Quadratic* Discriminant Classifier (Quadratic Discriminant Classifier (QDC)). If the class covariance matrices are identical ($\Sigma_i = \Sigma$), the equation becomes

$$\begin{aligned} g_i(X) &= log[P(\omega_i)] - \frac{1}{2}M_i^T\Sigma^{-1}M_i + M_i^T\Sigma^{-1}X \\ &= w_{i0} + W_i^T X, \end{aligned}$$

which is clearly linear.

The general form of a QDC is:

$$g(X) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d}\sum_{j=1}^{d} w_{ij} x_i x_j, \quad (2.2)$$

and the coefficients of Equation (2.2) can be inferred by comparing it with Equation (2.1). They are usually estimated by invoking a training phase for the classifier.

BI methods can be used to get the "best" linear and quadratic classifiers too with a reduced set of points. This will also be highlighted in the forthcoming chapters.

2.1.7 System Evaluation

The final phase of a PR system is the evaluation process, in which the performance of the classifier is analyzed based on some criteria. Typically, the performance is a complex and composite phenomenon from which the classification accuracy can

be considered to be one of the most important measures. The accuracy of a classifier is usually measured as a function of the number of successful and unsuccessful classifications. We shall elaborate on this below.

Suppose that the testing data set T has N patterns, and the set is tested against the classifier D. If the classifier mistakenly classifies E patterns, then the **Error**, which is proportional to the number of misclassified vectors, is given as:

$$Error(D) = \frac{E}{N}.$$

As this index is purely based on the number of misclassifications, it is often referred to as the *counting estimator* of the error rate.

A more detailed study of the number of errors requires the concept of the so-called Confusion Matrix (CM). For a two-class problem, each test instance can result in any one of four possible outcomes. A correctly classified vector is the one whose true class label matches with the class label obtained by the classifier. These incorporate the True Positive (TP) and True Negative (TN) cases. An incorrectly classified case can be a positive instance classified as negative, i.e., a False Negative (FN), or a negative instance classified as positive, i.e., a False Positive (FP). The CM records the number of TPs, TNs, FNs, and FPs. In general, the CM will be as shown in Figure 2.4. In the figure, the actual number of samples in the classes are represented by P and N, and the predicted outcomes are P' and N' respectively.

From the CM, the following rates can be used to quantify the quality (or efficiency) of the classifier:

$$True\ Positive\ Rate,\ TPR = \frac{TP}{P}$$
$$False\ Positive\ Rate,\ FPR = \frac{FP}{N},$$

where the Accuracy (Acc) is given by:

$$Accuracy = \frac{TP + TN}{P + N}.$$

Typically, the costs associated with erroneous decisions will also have to be incorporated so as to express the classifier's quality. However, since this is not an issue for our analysis, we shall not delve into this any further.

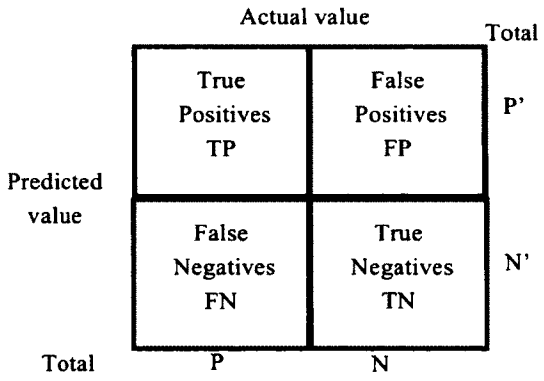


Figure 2.4: The schematic of a typical Confusion Matrix, where the notation is in the text.

Crucial to our study is the understanding of how BI algorithms can be used to enhance the accuracy of the classifier in question.

Receiver Operating Characteristics (ROC) Curves

Since the Accuracy (Acc) is a fairly generic metric for quantifying the quality of a classifier, researches have, more recently, resorted to the so-called Receiver Operating Characteristics (ROC) to capture its finer details.

An ROC is a graphical plot which shows the comparison of the TP and FP operating characteristics of the classifier. In an ROC curve, the TP rate is plotted against the FP rate, and thus the curve can categorize classifiers based on their ranking performance. Each point in the ROC space represents a confusion matrix, which is evaluated based on the results of the classification. To be more specific, the coordinate (0, 0) is the trivial classification that represents the case with zero accuracy, which means that all the positive instances are incorrectly classified as negatives and that all the negative instances are incorrectly classified as positives. On the other extreme, the point (1, 1) represents the trivial classification in which all the positive instances are correctly classified and all negative instances are incorrectly classified as positives. The point (0, 1) represents the optimal performance as it depicts a 100% TP rate and a 0% FP rate. Finally, the point (1, 0) represents the performance of

a learning model whose predictions are always inverted, which signifies that positive instances are consistently classified as negatives and vice versa.

Consider the following confusion matrix given in Figure 2.5 that represents four results, in which there are 100 positive instances and 100 negative instances. The TP and FP rates and the accuracies are also given below the confusion matrices.

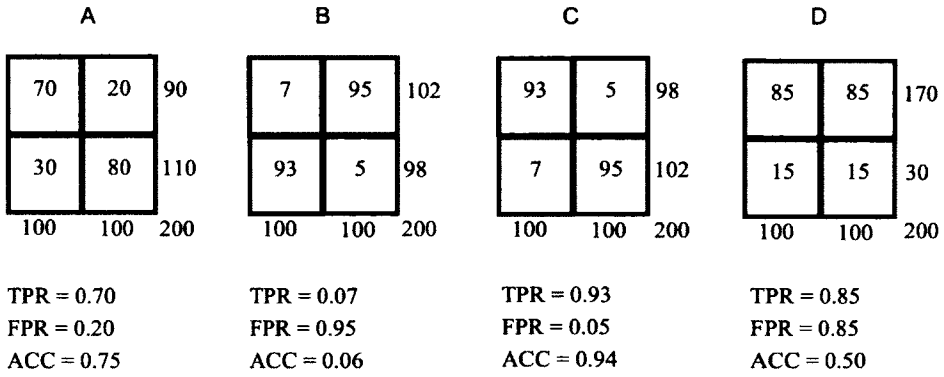


Figure 2.5: Various Confusion Matrices with their corresponding TP and FP rates, and Accuracies.

The results of these experiments can be plotted in the ROC space, as shown in Figure 2.6, where the blue line indicates random sampling with 50% accuracy. The space above the line indicates a better performance as compared to the random sampling, and the space below the line represents an inferior performance. Thus, the result given by Scenario A has a better performance, and the result B has the worst performance. Observe that Result C is just the mirrored case of Result B.

In order to obtain a ROC curve, we have to plot the TP rates against the FP rates for different cut-off points. To be more specific, consider the following example (in Table 2.3) in which we are given the data of T4 levels that determine hyperthyroidism [44]. From this data, we can evaluate the TP and FP rates for different cut-off points, leading to the ROC curve given in Figure 2.7.

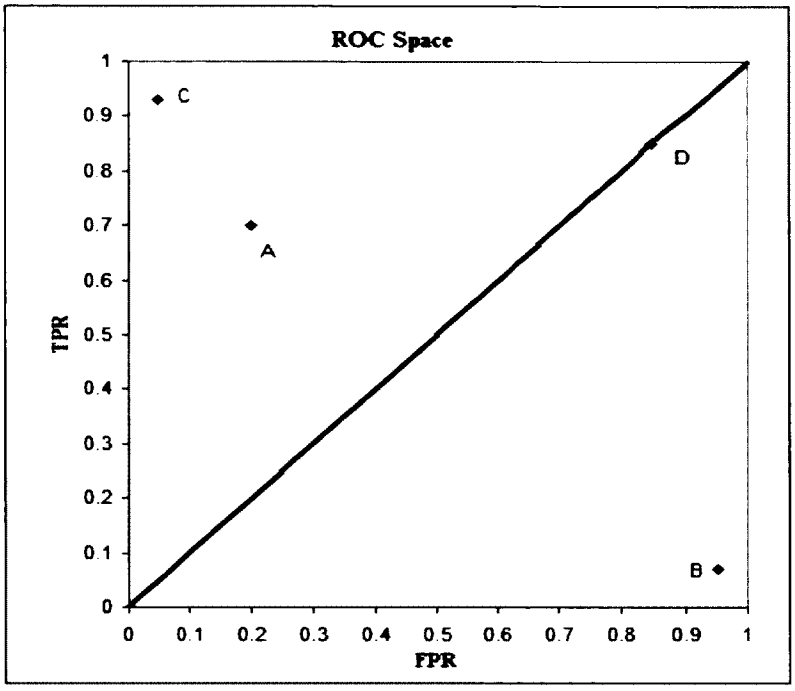


Figure 2.6: The ROC space plotted for the data in Figure 2.5.

T4 value	Hypothyroid	Euthyroid
5 or less	18	1
5.1 - 7	7	17
7.1 - 9	4	36
9 or more	3	39

Table 2.2: Sample data for T4 levels for hypothyroidism.
(Taken from `http://gim.unmc.edu/dxtests/ROC1.htm`.)

Cutoff point	TPR	FPR
5	0.56	0.99
7	0.78	0.81
9	0.91	0.42

Table 2.3: The TP and FP rates for the data given in Table 2.2.

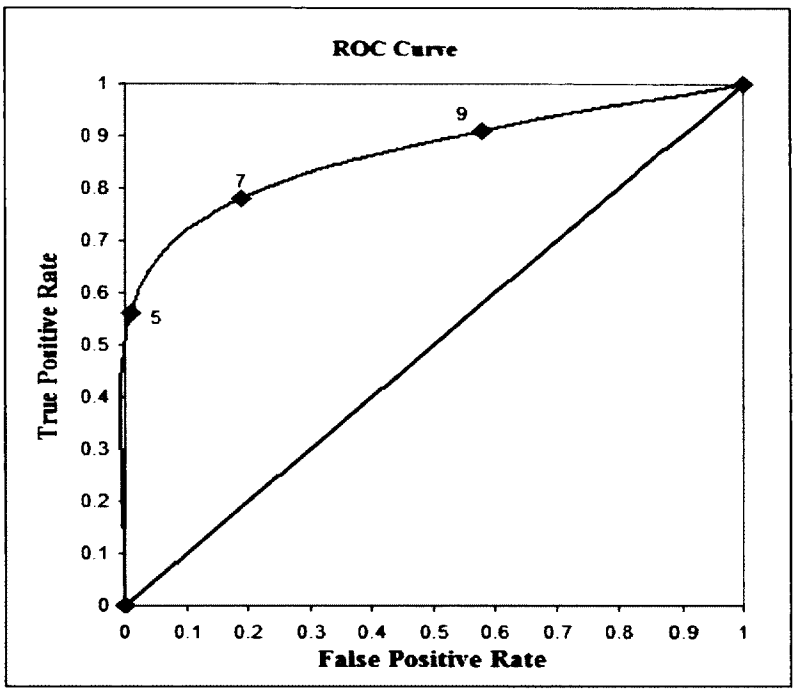


Figure 2.7: An example of an ROC Curve.

2.2 Prototype Reduction Schemes (PRS)

We have argued earlier that the premise for our study is to overcome a huge computational bottleneck, i.e., one which occurs due to the size of the training set. In this regard, although we investigate the use of BI schemes, alternate methods which use Prototype Reduction Schemes (PRS) have also been extensively studied. This section examines some of these methods in some detail.

In traditional non-parametric classification, such as those invoking a NN-type algorithm, the training patterns play a significant role in the classification process. This is because a decision boundary is obtained by considering *all* the samples in the training set. However, modern rapid advancements in this field have led to the development of efficient classification methods in which the schemes achieve the classification based on a *subset* of the training patterns. The question really is one of knowing how this subset can be obtained.

Nearest Neighbor (NN) methods often suffer from the computational complexity caused by the large number of distance computations, especially as the cardinality of the data set increases in high-dimensional problems. Various methods that have been proposed to resolve this drawback can be summarized into the three categories [27]:

1. Reducing the size of the set of training vectors without sacrificing the performance;

2. Accelerating the computation by eliminating the necessity of calculating superfluous distances;

3. Increasing the accuracy of the classifiers designed with the set of limited samples.

A Prototype Reduction Schemes (PRS) is a generic method for reducing the number of training vectors, while simultaneously attempting to guarantee that the classifier built on the reduced design set performs as well, or nearly as well, as the classifier built on the original design set [28]. Instead of considering all the training patterns for the classification, a subset of the whole set is selected based on certain criteria. The learning (or training) is then performed on this reduced training set, which is also called the Reference set.

Numerous PRS techniques have developed over the years. These include the Condensed Nearest Neighbor (CNN) rule [20], the Reduced Nearest Neighbor (RNN) rule [18], the Prototypes for Nearest Neighbor (PNN) classifiers [3], the Selective Nearest Neighbor (SNN) rule [40], two modifications of the CNN [52], the Edited Nearest Neighbor (ENN) rule [6], and the non-parametric data reduction method [16], some of which are briefly explained below. Additionally, in [56], the Vector Quantization (VQ) technique [29] was also reported as an extremely effective approach to data reduction. It has also been shown that the SVM itself can be used as a means of selecting initial prototype vectors, which are subsequently operated on by LVQ3-type methods [27]. While some of the above techniques merely *select* some of the existing patterns as prototypes, other techniques *create* new prototypes such that the latter can represent *all* the existing patterns in the best manner. Out of the above listed PRS techniques, CNN, RNN, SNN and ENN merely *select* prototypes from

the existing patterns, while the PNN and VQ create new prototypes that collectively represent the entire training set.

2.2.1 Condensed Nearest Neighbor (CNN)

The CNN is suggested as a rule that retains the basic approach of the Nearest Neighbor rule to determine a consistent subset of the original sample set. However, this technique, in general, will not lead to a minimal consistent sample set, which is a set that contains a minimum number of samples able to correctly classify all the remaining samples in the given set.

Initially, the first pattern of the original training set T is copied to T_{CNN}. Then, the second pattern of T is classified by considering T_{CNN} as the Reference set. If that pattern is correctly classified, it is moved to the set of patterns to be removed. Otherwise, it is moved to the Reference set. This procedure is repeated for all the patterns of T. Once all the patterns have been considered for such a verification phase, the same procedure is repeated for the set R, which contains the patterns to be removed. This phase will be repeated until either the set R becomes empty , or no more patterns are left in R which have any effect on the classification.

Once this pre-processing has been achieved, T_{CNN} will be the Reference set for the NN rule. The patterns that are moved to R will be discarded.The CNN algorithm can be summarized as follows:

2.2.2 Reduced Nearest Neighbor (RNN)

The Nearest Neighbor Method, originally proposed by Cover and Hart [18], is a simple straightforward efficient program which is being used for several applications. By proposing the CNN, Hart then suggested a strategy for decreasing the computations and memory required to achieve efficient classification. After the development of the NN and CNN, Gates proposed the RNN as an extension of the CNN, that attempts to further reduce the original training set as suggested by the CNN.

The RNN algortihm first invokes the CNN algorithm to obtain T_{CNN}, the reduced

Algorithm 1 CNN(T, N)

Input:

i) T, the original training set

ii) N, total number of patterns

Output:

i) T_{CNN}, the reduced training set by CNN rule

ii) R, the set of patterns to be removed

Method:

```
1: T_CNN ← T[1]
2: for i ← 2 to N do
3:    classify each X_i ∈ T using T_CNN as the Reference set
4:    if X_i is correctly classified then
5:       R ← R ∪ {X_i}
6:    else
7:       T_CNN ← T_CNN ∪ {X_i}
8:    end if
9: end for
10: repeat
11:    MOVE ← FALSE
12:    for all X_i ∈ R do
13:       classify each X_i ∈ R using T_CNN as the Reference set
14:       if X_i is incorrectly classified then
15:          T_CNN ← T_CNN ∪ {X_i}
16:          MOVE ← TRUE
17:       end if
18:    end for
19: until MOVE ≠ FALSE
```

End Algorithm

training set derived by the CNN rule. It then tries to discard those patterns[5] which do not have any influence in the classification process. To accomplish this, the RNN algorithm removes one pattern per iteration, by classifying T using the set T_{RNN}

[5]This is exactly what we will attempt to do for BI schemes.

as the Reference set. If at least one pattern is not correctly classified, it is obvious that the removed pattern has some influence in the classification. Consequently, the removed pattern is again included into T_{RNN}, and the procedure is continued with the next pattern of T_{RNN}.

The RNN algorithm can be summarized as below [18].

Algorithm 2 RNN(T, N)

Input:

i) T, the original training set

ii) N, total number of patterns

Output:

i) T_{RNN}, the reduced training set by RNN rule

Method:

```
1: CNN(T, N)    // T_CNN, the reduced training set by CNN rule is obtained
2: T_RNN ← T_CNN
3: for i ← 1 to N do
4:    X ← T_RNN[i]
5:    remove T_RNN[i] from T_RNN
6:    classify T by T_RNN
7:    if at least one pattern is incorrectly classified then
8:       return X into T_RNN
9:    end if
10: end for
End Algorithm
```

2.2.3 Prototypes for Nearest Neighbor (PNN)

Another PRS scheme, the Prototypes for Nearest Neighbor (PNN) algorithm [3], can be described as follows: Given a training set T, the algorithm starts with every point in T as a prototype. We now define two auxiliary sets A and B. Initially, set A is empty and set B is equal to T, where every prototype (data sample) has an associated weight of unity. The algorithm selects an arbitrary point in B and initially assigns

it to A. After this, the two closest prototypes P in A and Q in B of the same class are merged, successively, into a new prototype, P^*. This is done only if the merging will not degrade the classification of the patterns in T, where P^* is the weighted average of P and Q. For example, if P and Q are associated with weights w_P and w_Q, respectively, P^* is defined as $(w_P P + w_Q Q)/(w_P + w_Q)$, and is assigned a weight, $w_P + w_Q$. After determining the new P^*, P from A and Q from B are deleted, and P^* is included into A. Thereafter, the procedure is repeated until a static condition attains.

If either P and Q are not of the same class, or if merging is unsuccessful, Q is moved from B to A, and the procedure is repeated. When B becomes empty, the entire procedure is repeated by letting B be the final A obtained from the previous cycle, and by resetting A to be the empty set, until no new merged prototypes are obtained. The final prototypes in A are then used in a NN classifier. The bottom-up nature of this method is crucial to its convergence.

Though the algorithm is fairly straightforward, in the interest of completeness, it is given below. However, to clarify issues, we also give an example for the PNN in Figure 2.2.3.

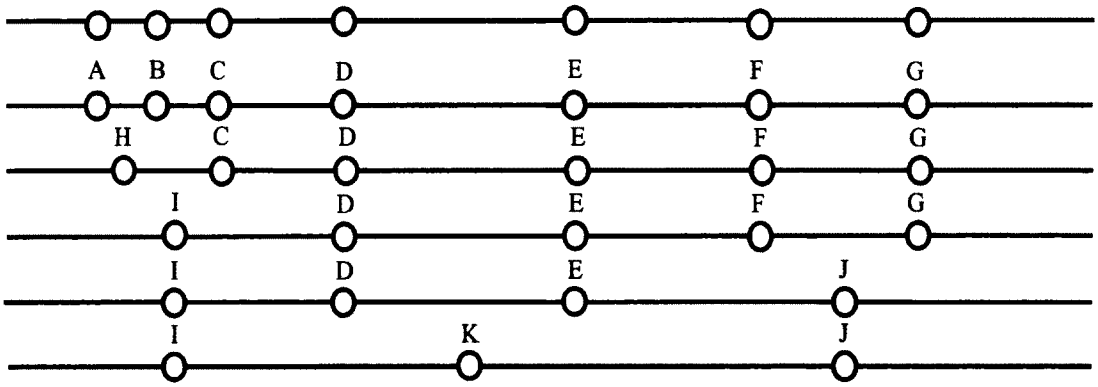


Figure 2.8: The example for the Prototypes for Nearest Neighbor (PNN) algorithm.

Here, the initial patterns are represented by the letters A, B, C, D, E, F, and G. While executing the PNN, the patterns A and B are merged to form the pattern H, and in the next iteration, patterns H and C are merged to form I. Similarly, patterns F and G are merged to form the pattern J, and the patterns D and E are merged to

form the pattern K. After the execution of the PNN, the Reference set obtained is the set {I, K, J}.

2.2.4 Vector Quantization (VQ) and the Self Organizing Map (SOM)

Vector quantization is a lossy data compression method in which large data sets are reduced to a set consisting of a few representative virtual points. In general, a subset of data points, which are in "close" proximity and which fall in a particular region are represented by a single virtual point. Thus, based on the features of the data points, the whole data set will be divided into a number of regions, and the reduced set will consist of fairly unique representatives of the respective regions. This phase is referred to as intra-regional polarizing phase [27].

In both VQ and SOM, the polarizing algorithm is repeatedly invoked by using the data vectors $\{X_i\}$ from the set of feature vectors of the class in question. The neurons attempt to incorporate the topological information present in $\{X_i\}$. This is done as follows. First of all, the closest neuron to X_i, Y_{j*}, is determined. This neuron and a group of neurons in its neighborhood, B_{j*}, are now moved in the direction of X_i. The set B_{j*} is called the "activation bubble". The actual migration of the neurons is achieved by rendering the new Y_j to be a convex combination of the current Y_j and the data point X_i for all $j \in B_{j*}$. More explicitly, the updating algorithm is as follows:

$$Y_j(t+1) = \begin{cases} (1-\alpha(t))Y_j(t) + \alpha(t)\ X_i & \text{if } j \in B_{j*}(t); \\ Y_j(t) & \text{otherwise,} \end{cases}$$

where t is the discretized (synchronized) time index. This basic algorithm has two fundamental parameters, $\alpha(t)$ and the size of the bubble $B_{j*}(t)$. $\alpha(t)$ is called the adaptation constant, and satisfies $0 < \alpha(t) < 1$. Kohonen and others [22, 29] recommend steadily decrementing $\alpha(t)$ linearly from unity for the initial learning phase, and then switching it to small values which decrease linearly from 0.2 for the fine-tuning phase.

The parameter activation bubble, $B_{j*}(t)$, determines the nature of the scheme. If the size of the bubble is always set to zero, the set B_{j*} is always empty, and so only the closest neuron is migrated, yielding a VQ scheme. However, in the SOM, a non-zero value of the activation bubble results in the migration of nearest neuron *and* the neurons within the bubble of activation. This widened migration process enables the algorithm to be *both* topology preserving and self-organizing. The size of the bubble is initially assigned to be fairly large in order to lead to a global ordering. Consequently, all the neurons tend to "coalesce" into a knot when $\alpha(t) \approx 1$. They later move away from this location to find their ultimate positions. Subsequent to this coarse spatial resolution, the size of the bubble is steadily decreased. Thereafter, only those neurons which are most relevant to the processed input point will be effected by the prescribed migration, implying that the ordering which has been achieved by the coarse resolution is not disturbed, but only a fine tuning is permitted.

The following figure (adapted from [23]) represents a 2D VQ. The regions are marked with blue lines, and the virtual representatives are given with red stars. The red stars are called the codebook vectors, and the regions defined by the blue borders are referred to as the encoding regions, which partition the space.

Although a number of revisions and enhancements have been suggested by researchers, the basic idea remains the same. Observe that both the VQ and the SOM participate in "competitive learning" in which the patterns compete for the right to respond to the input data points. One should note that the VQ scheme preserves only the distribution. But by the widened migration process, the SOM preserves both the distribution and achieves a topology ordering. An example[6] in which one can observe the ability of the SOM to preserve the essential properties of the input vectors is shown in Figures 2.10 to 2.12. The first figure, Figure 2.10, displays the original random location of the codebooks and the distribution of the data points, which falls along a curve in the plane. Different phases of the convergence of the SOM are shown in Figure 2.11, and the final output is given in Figure 2.12, where the codebook points are observed to be sequential. By examining Figures 2.10 and 2.12, one can see the distribution and topology preserving ability of SOM.

[6]We gratefully acknowledge Dr. Astudillo, a previous student of Dr. Oommen for these figures.

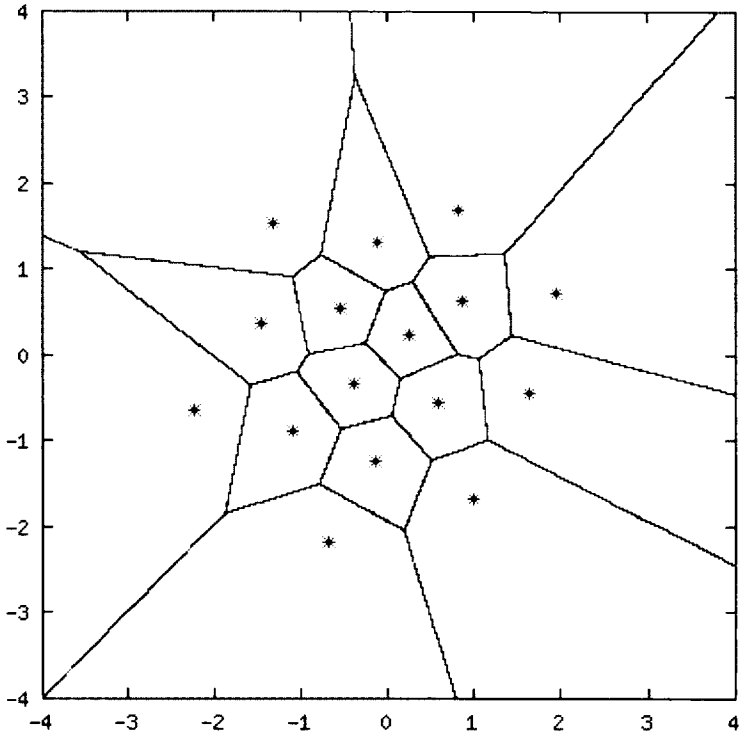


Figure 2.9: An example of Vector Quantization.
(Taken from `http://www.data-compression.com/vq.shtml`.)

2.2.5 Hybridized Kim-Oommen algorithm using LVQ3

A more recent PRS is the hybridized Kim-Oommen algorithm given in [27]. This method is an enhancement of the conventional data reduction methods which additionally use a variant of VQ, namely the the LVQ3 technique. In LVQ3 (as opposed to LVQ1), two code-book vectors M_i and M_j, which are the two nearest neighbors to X, are simultaneously updated, where X and M_j belong to the same class, and X and M_i belong to different classes. Moreover, X must fall into a zone of values called the "window", which is defined around the mid-plane of M_i and M_j. Let d_i and d_j be the Euclidean distances of X from M_i and M_j, respectively. Then X is defined to fall in a window of relative width w if

$$min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > \left(\frac{1-w}{1+w}\right). \tag{2.3}$$

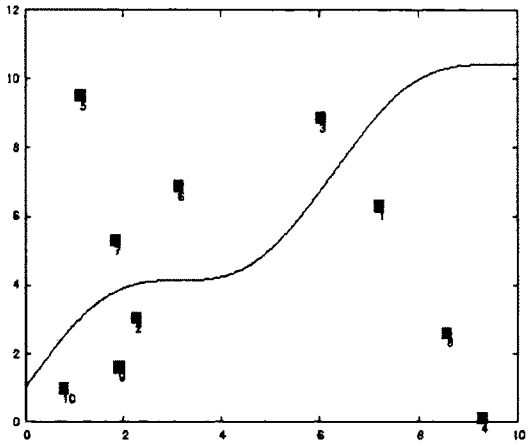


Figure 2.10: Input data points provided to the SOM.

The updating rules for M_i and M_j ensure that the codebook vectors continue to approximate the respective class distributions, and simultaneously enhance the quality of the classification boundary. These rules are

$$M_i(t+1) = M_i(t) - \alpha(t)[X(t) - M_i(t)]; \tag{2.4}$$

$$M_j(t+1) = M_j(t) + \alpha(t)[X(t) - M_j(t)]. \tag{2.5}$$

Additionally, even when X, M_i and M_j belong to the same class, the code-book vectors are adjusted to enhance the improvement as follows for $k = i, j$:

$$M_k(t+1) = M_k(t) - \epsilon(t)\alpha(t)[X(t) - M_k(t)]. \tag{2.6}$$

In Equations (2.5) and (2.6), t is the discretized (synchronized) time index, and $\alpha(t)$ and $\epsilon(t)$ are called the learning rate and relative learning rate, respectively.

Oommen and Kim enhanced any arbitrary conventional data reduction method by applying a post processing LVQ3 phase.Thus, the first step either selects or creates initial prototypes by applying any of the above-described traditional PRSs. After that, the optimal positions are learned by invoking an LVQ3-type scheme. The procedure is formalized below for each class[7]:

[7]The following steps have been taken verbatim from [27].

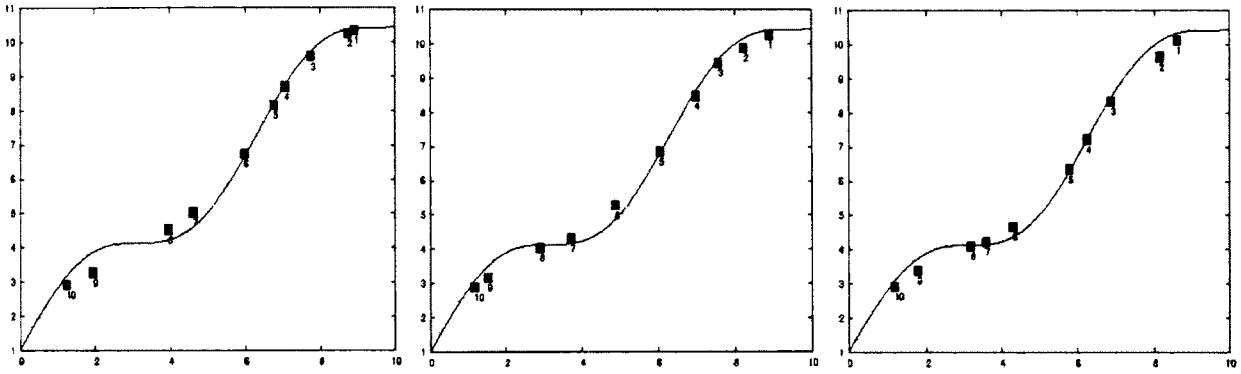


Figure 2.11: The location of the neurons after different stages of migration for the input given in Figure 2.10, i.e., after different numbers of training cycles respectively.

1. For every class j, select an initial condensed prototype set $Y_{j,Test}$ by using any one of the reduction methods described earlier, and the entire training sets $T_{i,t}$

2. Using $Y_{j,Test}$ as the set of condensed prototype vectors for class j, do the following using the Placement sets, $T_{i,p}$, and the Optimizing sets, $T_{i,O}$ for all the classes:

   a Perform LVQ3 using the points in the Placement set, $T_{i,P}$. The parameters of the LVQ3 are spanned by considering increasing values of w from 0.0 to 0.5, in steps of Δw. The sets $Y_{j,Test}$ (for all j) and Y_{Test} are updated in the process. Select the best value w_0 after evaluating the accuracy of the classification rule on $T_{i,0}$, where the NN-classification is achieved by the adjusted Y_{Test}

   b Perform LVQ3 using the points in the Placement set, $T_{i,P}$. The parameters of the LVQ3 are again spanned by considering increasing values of ϵ from 0.0 to 0.5, in steps of $\Delta\epsilon$. The sets $Y_{j,Test}$ (for all j) and Y_{Test} are updated in the process. Select the best value ϵ_0 after evaluating the accuracy of the classification rule on $T_{i,0}$, where the NN classification is achieved by the adjusted Y_{Test}.

   c Repeat the above steps with the current w_0 and ϵ_0, till the best values w^* and ϵ^* are obtained

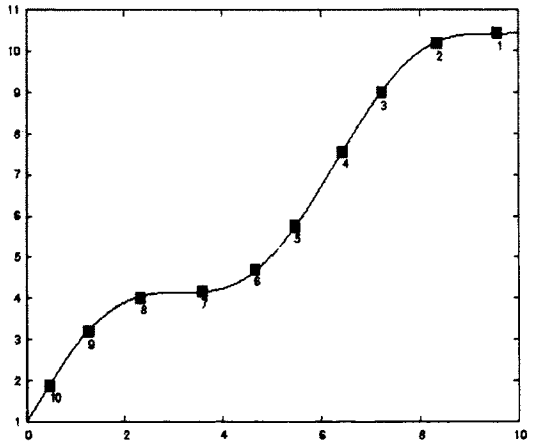


Figure 2.12: Output of the SOM after it has converged for the input given in Figure 2.10.

3. Determine the best prototype set Y_{Final} by invoking LVQ3, η times, with the data in $T_{i,P}$, and where the parameters are w^* and ϵ^*. Again, the 'pseudo-testing' is achieved using the Optimizing set, $T_{i,O}$.

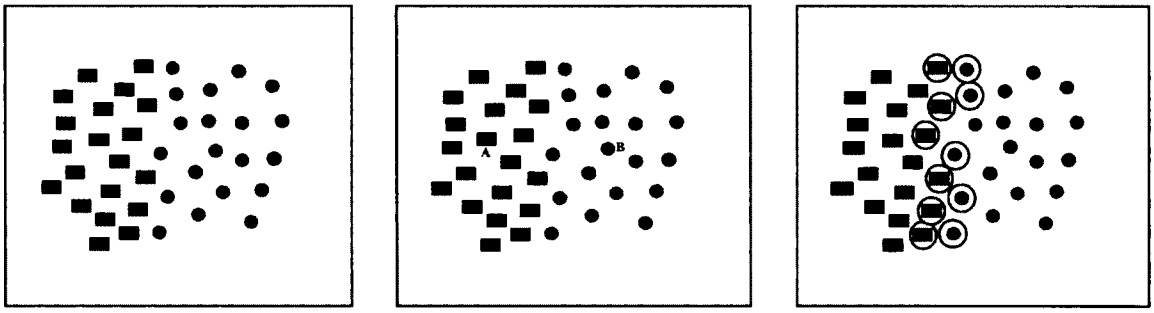
The actual classification accuracy is obtained by testing the classifier using the final values Y_{Final} and the original testing (validation) data points, $T_{i,V}$.

2.3 Prototype Reduction based on Border Concept

2.3.1 Border patterns versus Prototypes

Before we delve deeper into the concept of borders, it is crucial for us to understand the difference between "prototypes", as described in the previous section, and "border" samples. Consider Figure 2.13a in which the circles belong to ω_1 and rectangles belong to ω_2. A PRS would attempt to determine the relevant samples in both the classes which are capable of achieving near-optimal classification. Thus, the samples in the two classes which are marked in Figure 2.13b could be potential prototypes. Observe that some samples which fall strictly *within* the collection of points in each class, such as A and B, could be prototypes, because testing samples that fall close to them will be correctly classified.

As opposed to this, in a BI algorithm, the aim is to distinguish those samples which lie close to the boundaries of the two classes as shown in Figure 2.13c. The reader should observe that although this seems to be an obvious task in a two-dimensional world, it is far from trivial in a multi-dimensional space.



(a) Training Set (b) Prototypes (c) Border Patterns

Figure 2.13: Border patterns vs Prototypes.

A number of algorithms have been proposed for determining the reduced Reference set based on the concept of "border" patterns [8]. These algorithms can be classified into three families:

1. Clusterization-based algorithms
2. Algorithms starting with the entire training set, and
3. Algorithms starting with an empty set.

We shall discuss each of these families below.

2.3.2 Clusterization-based algorithms

The family of clusterization-based algorithms works using the same principles of a previously-mentioned PRS scheme. In Section 2.2.2, we had described the RNN which operated by starting with an initial set and excluding elements based on the accuracy of the classifier derived from the set. This is exactly the principle motivating BI algorithms within this category. For such clusterization-based algorithms, an initial Reference set is selected using an established clustering technique. A classifier is then

built for the current Reference set, using which the classification accuracy is evaluated. After the testing, wrongly classified patterns are then added to the Reference set. Observe that unlike the RNN, our aim here is not that of determining the prototypes, but rather of inferring the Reference set comprising of the borders.

2.3.3 Traditional Border Identification Algorithms

Duch's Algorithms

Duch has developed algorithms to obtain the Reference set based on a border analysis of every training pattern. He has designed two techniques which serve to select the most effective reference vectors *near* the class borders. The first method, referred to as `Duch1`, starts with an empty Reference set. For every training pattern X, the scheme identifies k nearest patterns, and those patterns which are from the class other than the class of X are added to the Reference set. In this way, the algorithm, in effect, attempts to add patterns which are closer to the class boundary, to the Reference set.

The whole procedure is repeated a number of times, from a maximum value of k, denoted by K_2, to a minimum value of k, denoted by K_1, with $K_1 < K_2$.
The algorithm `Duch1`, which operates in a top-down manner, can be summarized as follows in Algorithm 4:

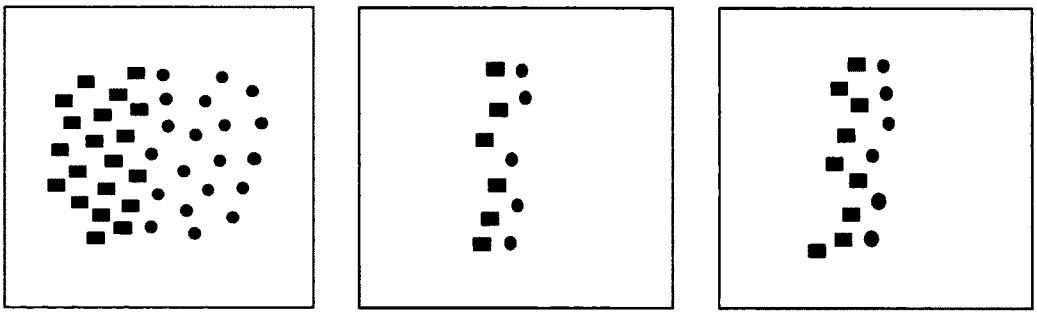
The function $kNN(X, T, i)$ returns the i nearest neighbors of X from the training set T. The function $clscheck(P, X)$ returns `True` if all the vectors of P are from the same class as that of X.

The alternate method proposed by Duch to select the Reference set (without initial clusterization) starts with the entire training set T. For every pattern X in T, k nearest patterns are identified. If all the k nearest patterns are from the same class as that of X, then the pattern X is removed from the Reference set, since all the removed patterns are, possibly, farther from the class borders. Thus, the Reference set that is retained contains only the patterns which are closer to the class borders.
Duch's second algorithm, which is essentially a bottom-up strategy, is summarized in Algorithm 5 below:

An illustration for both of Duch's approaches for the determination of the Reference set is shown in Figure 2.14, where we have a training set which has patterns of rectangles and circles. Each class has 20 patterns, with a total of 40 patterns. The original training set is given in Figure 2.14a.

Algorithm 4 (`Duch1`) starts with an empty set. Training patterns will be added to this set based on the test given by line 5 of Algorithm 4. In this example, $k = 2$ nearest neighbors of every pattern X are determined, and if the nearest neighbors are not from the class of X, they will be added to the Reference set. The final Reference set, consisting of the consequent borders for the given training set is shown in Figure 2.14b.

Algorithm 4 (`Duch2`), on the other hand, starts with the entire training set, and the training patterns are discarded from it during the iterations of the algorithm. For this example, when $k = 5$, nearest neighbors of each pattern X are determined, and if all of the nearest neighbors are from the same class as that of X, the latter pattern, X, will be removed from the set. This process will then be repeated for all the training patterns. The border set obtained after applying this algorithm on the given training set is presented in Figure 2.14c.



(a) Original Training Set

(b) The final Reference set, consisting of the borders, as obtained by Algorithm 4

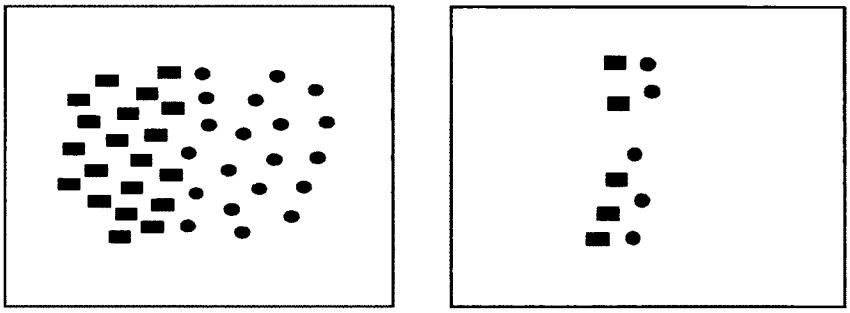
(c) The final Reference set, consisting of the borders, as obtained by Algorithm 5

Figure 2.14: Illustration of Duch's Approaches.

Foody's Algorithm

Another important development in this area was proposed by Foody [13]. According to his approach, the training set is divided into two sets - the first comprising of the set of border patterns, and the second being the set of non-border patterns. A border training set should contain patterns from different classes, but which are close together in the feature space and which are thus in the proximity of the true classification boundary. In order to decide whether a training pattern is a border pattern or not, a scale of "borderness" is defined. Foody expressed "borderness" as the difference between the two smallest Mahalanobi's distances measured for each training pattern. To achieve this, all the training patterns are ordered in terms of the magnitudes of their respective differences between the two smallest distances. The patterns with a low difference, i.e., those with a value smaller than a threshold value δ, between the two smallest distances are included in the border training set. Likewise, the patterns with a relatively large distance between the two smallest distances are considered as being non-border or the core patterns.

The original training set and the border set obtained by applying Foody's approach is as shown in Figures 2.15a and 2.15b for the same training set used in Figure 2.14a.



(a) Original Training Set for Foody

(b) The final Reference set, consisting of the borders, as obtained by Algorithm 6

Figure 2.15: Illustration of Foody's Approach.

Foody's algorithm can be summarized as in Algorithm 6.

In order to apply Duch's first and second approaches, the value of k should be considered as a user-defined parameter. Similarly, the threshold value, δ, is the corresponding parameter for Foody's approach. For the rest of this section, we shall refer to Duch's and Foody's schemes as being "traditional" BI algorithms.

2.3.4 Border Identification in Two Stages (BI_2)

The "traditional" BI algorithms proposed by Duch and Foody are based on the concept of similarity distance metrics. However, the drawback of those algorithms is that they are unable to learn the so-called "full" border. The traditional BI algorithms are able to obtain the samples close to the classification boundary, but are not able to determine the so-called "Far" borders. As these algorithms are not able to learn the "Full" border, more recent research had proposed the concept of finding the border patterns in two stages [33]. The border patterns obtained by the traditional approaches are considered to be the "Near" borders, and using the latter, the "Far" borders are identified from the remaining data points. It turns out that the final border points computed in this manner are more accurate than the initially identified "Near" borders. The Near and the Far borders collectively constitute the so-called Full border set for the training data.

The algorithm that we allude to, due to Li *et.al.*, will be referred to as BI_2. In order to understand the principles motivating BI_2, we need to understand two terms - i.e., those concerning "informative patterns" and "redundant patterns", as in [33].

Informative Patterns: A pattern X of class C_i is an "informative" pattern of class C_j, $i \neq j$, if X is one of the k nearest neighbors of any of the patterns of the class C_j. In other words, the informative patterns of X are its nearest neighbors from the alternate class.

Redundant patterns: A pattern can be considered to be "redundant" iff it is not selected as an informative pattern, and if it is close to an informative pattern of the same class.

For the first stage of the BI_2 algorithm, Li *et al.* used an approach similar to Algorithm 4 to get the initial Reference set. Thus, in the first stage, BI_2 starts with

an empty set, and generates the Near borders between the classes C_i and C_j by generating the informative data points of *either* C_i from C_j, *or* C_j from C_i. The decision is based on the first NN of each training pattern, i.e., quantified by the use of an appropriate (dissimilarity) distance metric. The points not included here constitute the "Remaining" set. After obtaining the Near borders, the algorithm then identifies the Far borders from the Remaining set. To achieve this, the scheme removes the redundant data patterns from the Remaining set. This process is repeated until the Remaining set becomes empty.

The $\texttt{BI}_2$ algorithm for Border Identification in Two Stages is given as Algorithm 8.

2.3.5 Progressive Border Sampling

BI has been suggested as a new alternative method for the reduction of the training set. However, the scheme suffers from the handicap of possessing high uncertainty because the resulting border is not always adequate for the purpose of classification. In order to overcome this impediment, the authors of [33] proposed the scheme BI_2, which is based on obtaining the Far border patterns that are closer to the class boundaries. The idea was to iteratively find more border patterns from the Remaining set. The concept of Progressive Sampling (PS) has been further applied to BI_2 so as to detect when the scheme has converged to, thereby, yield a more optimal border set. The new algorithm, the Progressive Border Sampling (PBS), proposed by Li *et al.*, progressively learns sufficient borders that can classify the whole training set. In other words, it is used to yield a criterion by which one can get an iterative process for finding new border patterns.

2.4 Classifier Fusion

Traditional PR systems perform classification based on a single strategy, which could result in a poor performance when there are noisy and distorted samples. Consequently, researchers have proposed new techniques for combining the decision of different classifiers, each with a different feature description or with a different classification strategy. A variety of methods have been designed to combine these different classifiers where the main objective of combining them is to improve the overall efficiency and accuracy.

The methods for combining classifiers can be categorized into two groups: ***classifier selection*** and ***classifier fusion*** [32]. The presumption in ***classifier selection*** is that each classifier is "an expert" in some local area of the feature space. When a feature vector $X \in R^d$ is submitted for classification, the classifier responsible for the neighborhood of X is given the highest credit when assigning the class label to X. The system is permitted to nominate *exactly* one classifier to make the decision, or more than a single "local expert". ***Classifier fusion***, on the other hand, assumes that all the classifiers are trained over the entire feature space, and that they are thereby considered to be competitive, rather than complementary.

Let $X \in \mathbb{R}^d$ be a feature vector and $\{1, 2, ..., c\}$ be the label set of c classes. Every mapping

$$D : \mathbb{R}^d \rightarrow [0,1]^c - \{\mathbf{0}\},$$

where $\mathbf{0} = [0, 0, ..., 0]^T$ is the origin of $\mathbb{R}^c$, is considered as a classifier. The output of D is referred to as a "class label" and is denoted by $\boldsymbol{\mu}_D(X) = [\boldsymbol{\mu}_D^1(X), \boldsymbol{\mu}_D^2(X), ..., \boldsymbol{\mu}_D^c(X)]^T$, $\boldsymbol{\mu}_D^i(X) \in [0,1]$. The components $\{\mu_D^i(X)\}$ are regarded as the posterior probabilities for the classes given X [32].

Classifiers are categorized into being one of three types:

- ***Crisp Classifier***: $\boldsymbol{\mu}_D^i(X) \in \{0,1\}, \quad \sum_{i=1}^c \boldsymbol{\mu}_D^i = 1, \ \forall X \in \mathbb{R}^n$
- ***Fuzzy Classifier***: $\boldsymbol{\mu}_D^i(X) \in [0,1], \quad \sum_{i=1}^c \boldsymbol{\mu}_D^i = 1, \ \forall X \in \mathbb{R}^n$
- ***Possibilistic Classifier***: $\boldsymbol{\mu}_D^i(X) \in [0,1], \quad \sum_{i=1}^c \boldsymbol{\mu}_D^i > 0 \ \forall X \in \mathbb{R}^n$

Various classifier fusion techniques can be categorized as shown in Figure 2.16.

First Level Output ↓	Training at Fusion Level	
	No	Yes
Crisp	Majority Vote	Behavior Knowledge Space, Naive Bayes
Soft	Min, Max, Average, Product	LDC, QDC, Fisher, Logistic classifier, Neural Networks, Decision Templates

Figure 2.16: Classifier Fusion Techniques.

2.4.1 Fusion of Label Outputs

Voting strategies can be applied to a multiple classifier system assuming that each classifier yields a single class label as an output. In order to combine all the possible information to lead to the best final decision, a number of approaches have been proposed, which are based on the voting strategy [41]. Some of the well known methods are explained below.

Majority Vote

Let us assume that the label outputs forms the decision vector $D = [D_{i1}, D_{i2}, ..., D_{ic}]^T \in \{0, 1\}$, $i = 1, 2, ..., L$, where $D_{ij} = 1$ if D_i labels X in ω_j, and 0 otherwise. The majority vote method results in an ensemble decision for the class ω_k if

$$\sum_{i=1}^{L} d_{i,k} = \max_{j=1}^{c} \sum_{i=1}^{L} D_{i,j}$$

In this method, ties are resolved arbitrarily. Another enhancement to this method was proposed by adding one more class, ω_{c+1} to represent the case when the classifier

fails to determine the class or when the decision results in a tie [57]. In this case, the decision is

$$\begin{cases} \omega_k & if \ \sum_{i=1}^{L} D_{i,k} \geq \alpha \cdot L \\ \omega_{c+1} & otherwise \end{cases}$$

where $0 < \alpha \leq 1$. When $\alpha = 1$, the above rule can be called as the "unanimity" vote rule, which means that a decision is made for some class label if all decision makers agree on that label. Otherwise, the label ω_{c+1} will be assigned to that X.

Weighted Majority Vote

In some scenarios, all the classifiers are not of equal accuracy. In such cases, in order to get an accurate result, appropriate decision power should be given for the more accurate classifiers. Such a fusion technique is named as the "Weighted Majority Vote" rule. The label outputs can be represented as degrees of support for the classes as shown below.

$$d_{i,j} = \begin{cases} 1 & if \ \ D_i \ labels \ X \ in \ \omega_j \\ 0 & otherwise. \end{cases}$$

The discriminant function for class ω_j obtained through a weighted vote is

$$g_j(X) = \sum_{i=1}^{L} b_i D_{i,j},$$

where b_i is a coefficient for class D_i. Thus, the value of the discriminant function $g_j(X)$ will be the sum of the coefficients for these members, which in turn, labels X into ω_j.

2.5 Conclusions

In this chapter, we have thoroughly investigated various PRS and traditional BI algorithms which are currently in use. Prior to this, we presented a *very brief* overview of the various phases of a PR system including feature extraction, training, classification, testing and system evaluation, and a brief analysis about a classifier's accuracy, the Receiver Operating Characteristics (ROC) curve, and the Area Under the

Curve (AUC). This was done because these concepts are fundamental in comparing and quantifying the efficiency of the algorithms that we will develop. The objective of PRSs is to reduce the cardinality of the training set to be as small as possible by selecting some training patterns based on various criteria, as long as the reduction does not affect the performance.

The algorithms designed by Duch, Foody and Li *et al.* (for determining the Border patterns) are also intended for selecting a Reference set which contains *border* patterns obtained from the set of training patterns. While explaining these, we emphasized that, in effect, these algorithms also result in reduced training sets. However, we are interested in the patterns which can classify the entire training set into the available categories, i.e., only those patterns which are closer to the true optimal classifier. We believe that in this case, the Reference set will contain the patterns drawn from different classes but which are simultaneously close in the feature space. Observe that as they are from different classes, they are more expected to lie near the true classifier. Therefore, we believe that these border patterns will more accurately classify the testing patterns, as we shall show in the next chapters.

In the interest of completeness, a brief survey about classifier fusion, and of ensemble fusion techniques is also provided in this chapter.

Algorithm 3 PNN(T, N)

Input:

i) T, the original training set

ii) N, total number of patterns

Output:

i) B, the reduced training set by PNN rule

Method:

1: $A \leftarrow \emptyset$, $B \leftarrow T$
2: **for** $i \leftarrow 1$ to N **do**
3: $w_Q = 1$
4: **end for**
5: $A \leftarrow A \cup \{X_k\}$ where $X_k \in B$
6: $MERGE = 0$
7: **repeat**
8: Find closest prototypes $P \in A$ and $Q \in B$
9: **if** $cls(P) \neq cls(Q)$ **then**
10: $A \leftarrow A \cup \{X_k\}$, $B \leftarrow B \setminus \{X_k\}$
11: **else**
12: $P^* = (w_P P + w_Q Q)/(w_P + w_Q)$
13: **end if**
14: **if** (classification error is increased) **then**
15: $A \leftarrow A \cup \{Q\}$
16: $B \leftarrow B \setminus \{Q\}$
17: **else**
18: $A \leftarrow A \setminus \{P\}$
19: $B \leftarrow B \setminus \{Q\}$
20: $A \leftarrow A \cup \{P^*\}$
21: $MERGE \leftarrow MERGE + 1$
22: **end if**
23: **until** $(B \neq \emptyset)$
24: **if** $(MERGE \neq 0)$ **then**
25: $B \leftarrow A$
26: GOTO 5
27: **end if**

End Algorithm

Algorithm 4 `Duch1`(T, K_1, K_2)

Input:

i) T, the original training set

ii) K_1, the minimum number of nearest neighbors

iii) K_2, the maximum number of nearest neighbors

Output:

i) A border, B, for the training set T

Method:

```
1: R ← ∅
2: for i ← K1 to K2 do
3:    for all X ∈ T do
4:       P ← kNN(X, T, i)
5:       if clscheck(P, X) then
6:          R ← R ∪ P
7:       end if
8:    end for
9: end for
10: B ← T \ R
```

End Algorithm

Algorithm 5 Duch2(T, k)

Input:

i) T, the original training set

ii) k, the number of nearest neighbors

Output:

i) A border, B, for the training set T

Method:

1: $B \leftarrow T$
2: **for all** $X \in T$ **do**
3: $P \leftarrow kNN(X, T, k)$
4: **if** *clscheck*(P, X) **then**
5: $B \leftarrow B \setminus \{X\}$
6: **end if**
7: **end for**

End Algorithm

Algorithm 6 Foody(T, N)

Input:

i) T, the original training set

ii) δ, a small threshold distance

Output:

i) A border, B, for the training set T

Method:

1: $R \leftarrow \emptyset$
2: **for all** $X, Y \in T$ **do**
3: **if** $cls(X) \neq cls(Y)$ **then**
4: **if** $dis(X, Y) < \delta$ **then**
5: $B = B \cup \{X\} \cup \{Y\}$
6: **end if**
7: **end if**
8: **end for**

End Algorithm

Algorithm 7 $\mathtt{BI}_2(C_i, C_j)$

Input:

i) C_i, C_j, two classes

Output:

i) B_{ij}, the identified border for the classes C_i and C_j

Method:

1: $B \leftarrow \emptyset$
2: $P_i = \cup_{p \in C_j} 1NN(p, C_i)$
3: $P_j = \cup_{p \in C_i} 1NN(p, C_j)$
4: $B_{ij} = B_{ij} \cup P_i \cup P_j$
5: $C_i = C_i \setminus P_j$
6: $C_j = C_j \setminus P_i$
7: $FarBorder(C_i, B_{ij})$
8: $FarBorder(C_j, B_{ij})$

End Algorithm

Algorithm 8 `FarBorder`(D, B_k)

Input:

i) D, class after removing the 1NN of other class

ii) B_k, initially obtained border points

Output:

i) B_f, far border for the classes C_i and C_j

Method:

1: $D' = D$
2: $B_f = \emptyset$
3: **while** TRUE **do**
4: $D' = RemoveRedundant(D', B_k)$
5: **if** $D' = \emptyset$ **then**
6: BREAK
7: **end if**
8: $B'_f = \cup_{p \in B_k} 1NN(p, D')$
9: $B_k = B_k \cup B'_f$
10: $B_f = B_f \cup B'_f$
11: $D' = D' \setminus B'_f$
12: **end while**
13: $D = D \setminus B_f$

End Algorithm

Chapter 3

The Foundational Theory of Optimal "Anti-Bayesian" *Parametric* PR Using OS Criteria

3.1 Introduction

The age-old strategy for achieving PR is based on a Bayesian principle which aims to maximize the *a posteriori* probability. It is well known that when expressions for the latter are simplified, the classification criterion which attains the Bayesian optimal lower bound often reduces to testing the sample point using the corresponding distances/norms to the *means* or the "central points" of the distributions. The gold standard for a classifier is the condition of optimality attained by such a Bayesian classifier. This chapter[1] demonstrates that this standard can be attained by other means – i.e., by resorting to an "anti-Bayesian paradigm".

[1] A preliminary version of some of the results of this chapter can be found in the *Proceedings of CIARP2012, the* 17^{th} *Iberoamenrican Congress on Pattern Recognition* held in Argentina, in September 2012 [47]. *This talk was a Plenary/Keynote Talk at the Conference.* More detailed descriptions of these results have been published in the journal *Pattern Recognition* [51].

3.1.1 Motivation of the Chapter

Within a Bayesian paradigm, if we are allowed to compare the testing sample with only *a single* point in the feature space from each class, the *optimal* Bayesian strategy would be to achieve this based on the (Mahalanobis) distance from the corresponding means. The reader should observe that, in this context, the mean, in one sense, is the most *central* point in the respective distribution. In this chapter, we shall show that we can obtain optimal results by operating in a diametrically opposite way, i.e., a so-called "anti-Bayesian" manner. Indeed, we shall show the completely counter-intuitive result that by working with a *very few* (sometimes as small as two) points *distant* from the mean, one can obtain remarkable classification accuracies. Further, if these points are determined by the *Order Statistics* of the distributions, the accuracy of our method, referred to as Classification by Moments of Order Statistics (CMOS), attains the optimal Bayes' bound. In this chapter, this claim, which is totally counter-intuitive, has been proven for a generic classifier, and the theoretical results have been verified by rigorous experimental testing. Apart from the fact that these results are quite fascinating and pioneering in their own right, they also give a theoretical foundation for the families of BI algorithms reported in the literature.

Before we proceed, we also mention that it is well-known that any specific OS from a set of points can be obtained *without sorting* the data set, but rather in *linear* time. Thus, computing the required OS will not require any more time than is needed for computing the sample mean itself.

3.1.2 PRSs, BI and OS-based PR

If we fast-forward the clock by five decades since the initial formulation of PR as a research field, the informed reader will also be aware of the development of efficient classification methods in which the schemes achieve their task based on a *subset* of the training patterns. These are PRS [17, 54] and BI schemes, which, for example, work with a Reference set containing only "border" points and which were surveyed in Section 2.2. As mentioned in the latter section, in order to oversee the task of achieving the classification, the samples extracted by a BI scheme, and which lie *close*

to the boundaries of the discriminant function, have significant information when it concerns the classification ability of the classifier.

Although this is quite amazing, as mentioned earlier, *the formal analytical reason for this is yet unproven.* We intend to resolve this.

3.1.3 Bridging the Conceptual Gap

In this chapter, we present some pioneering results which bridge the conceptual gap. First of all, we shall formally and analytically show that operating in a totally anti-Bayesian perspective can, in spite of the diametrically opposite philosophy, still lead to an *optimal* Bayesian classification. More precisely, as alluded to earlier, we shall show that by computing the appropriate distances/norms to certain (in many cases, as few as *two*) points that are distant from the means, we can obtain classification *identical* to the Bayesian scheme – as long as the corresponding comparisons are achieved using the appropriate data points that characterize the distributions. Indeed, the points that we refer to here will be shown to be the expected values of the moments of Order Statistics (OS) of the respective distributions. These representative OS points will model the above-mentioned representative prototypes derived by means of a BI algorithm, thus closing the conceptual gap.

The question of how we can compute the BI points which match these criteria from the training data is addressed in a later chapter.

We prove the claim for the generic classifier and for the uni-dimensional Uniform distributions. Even though the Uniform distribution *itself* is rather trivial and the calculations are straightforward, we still opt to work with it because the analysis will provide the reader with an insight into the mechanism by which the problem can be tackled, which can then be extended for other distributions, for example, within the exponential family.

3.1.4 Problem Formulation

The objective of a PRS is to reduce the cardinality of the training set to be as small as possible by selecting some training patterns based on various criteria, as long as the

reduction does not significantly affect the performance. Specializing this criterion, the current-day BI algorithms, designed by Duch, Foody, and Li *et al.*, and which are briefly explained in Section 2.3.3, attempt to select a Reference set which contains border patterns derived, in turn, from the set of training patterns. Observe that, in effect, these algorithms also yield reduced training sets. Once the Reference set is obtained, all of these traditional methods perform the classification by invoking some sort of classifier, like the SVM, a neural network etc. Recent research in the field of PR have claimed that the points which are closer to the class borders are more informative, and that they can play a significant role in the classification. Contrary to a Bayesian intuition, these border patterns have the ability to accurately classify the testing patterns, as we shall presently demonstrate. The prime objective of this chapter is to explain the theoretical reason for why those points are more informative and important.

Our main hypothesis is that the classification could just as well be attempted in the OS space as in the original space itself. Our work will also show that these OS points themselves are not necessarily central to the distribution.

3.1.5 Contributions of this Chapter

The novel contributions of this chapter are the following:

- We propose an "anti-Bayesian" paradigm for the classification of patterns within the parametric mode of computation, where the distance computations are not with regard to the "mean" but with regard to some samples "distant" from the mean. These points, which are sometimes as few as *two*, are the moments of OS of the distributions;
- We provide a theoretical framework for adequately responding to the question of why the border points are more informative for the task of classification;
- To justify these claims, we submit a formal analysis and the results of various experiments which have been performed for a generic classifier and for the uni-dimensional Uniform distributions, and the results are clearly conclusive.

3.1.6 Order Statistics

Let $\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_n$ be a univariate random sample of size n that follows a continuous distribution function Φ, where the probability density function (pdf) is $\varphi(\cdot)$. Let $x_{1,n}$, $x_{2,n}$, ..., $x_{n,n}$ be the corresponding Order Statistics (OS). The r^{th} OS, $x_{r,n}$, of the set is the r^{th} smallest value among the given random variables [5]. The pdf of $\mathbf{y} = \mathbf{x}_{r,n}$ is given by:

$$f_{\mathbf{y}}(y) = \frac{n!}{(r-1)!(n-r)!} \{\Phi(y)\}^{r-1} \{1-\Phi(y)\}^{n-r} \varphi(y),$$

where $r = 1, 2, ..., n$. The reasoning for the above expression is straightforward and is omitted here. It is found in [51].

Although the distribution $f_{\mathbf{y}}(y)$ contains all the information resident in $\mathbf{y}$, the literature characterizes the OS in terms of quantities which are of paramount importance, namely its moments [53], as briefly cited below.

Using the distribution $f_{\mathbf{y}}(y)$, one can see that the k^{th} moment of $\mathbf{x}_{r,n}$ can be formulated as:

$$E[\mathbf{x}_{r,n}^k] = \frac{n!}{(r-1)!(n-r)!} \int_{-\infty}^{+\infty} y^k \Phi(y)^{k-1} (1-\Phi(y))^{n-r} \varphi(y) dy,$$

provided that both sides of the equality exist [1, 36].

The fundamental theorem concerning the OS that we invoke is found in many papers [34, 36, 53]. The result is merely cited below inasmuch as the details of the proof are irrelevant and outside the scope of this study. The theorem, proven in [34], can be summarized as follows.

Theorem 1. *Let $n \geq r \geq k+1 \geq 2$ be integers. Then, since Φ is a nondecreasing and right-continuous function from $\mathbb{R} \to \mathbb{R}$, $\Phi(\mathbf{x}_{r,n})$ is Uniform in [0,1]. If we now take the k^{th} moment of $\Phi(\mathbf{x}_{r,n})$, it has the form:*

$$E[\Phi^k(\mathbf{x}_{r,n})] = \frac{B(r+k, n-r+1)}{B(r, n-r+1)} = \frac{n!\ (r+k-1)!}{(n+k)!\ (r-1)!}, \tag{3.1}$$

where $B(a,b)$ denotes the Beta function, and $B(a,b) = \frac{(a-1)!(b-1)!}{(a+b-1)!}$. □

The above fundamental result can also be used for characterization purposes as follows [34]. Let $n \geq r \geq k+1 \geq 2$ be integers, with Φ being nondecreasing and right-continuous. Let G be *any* nondecreasing and right-continuous function from $\mathbb{R} \rightarrow \mathbb{R}$ on the same support as Φ. The relation

$$E[G^k(\mathbf{x}_{r,n})] = \frac{n!\ (r+k-1)!}{(n+k)!\ (r-1)!} \tag{3.2}$$

holds if and only if $\forall x, \Phi(x) = G(x)$. In other words, $\Phi(\cdot)$ is the unique function that satisfies Eq. (3.2), implying that every distribution is characterized by the moments of its OS.

The implications of the above are the following:

1. If $n = 2$, implying that only *two* samples are drawn from $\mathbf{x}$, we can deduce from Eq. (3.1) that:

$$E[\Phi^1(\mathbf{x}_{1,2})] = \frac{1}{3}, \implies E[\mathbf{x}_{1,2}] = \Phi^{-1}\left(\frac{1}{3}\right), \text{and} \tag{3.3}$$

$$E[\Phi^1(\mathbf{x}_{2,2})] = \frac{2}{3}, \implies E[\mathbf{x}_{2,2}] = \Phi^{-1}\left(\frac{2}{3}\right). \tag{3.4}$$

   Thus, from a computational perspective, the first moment of the first and second 2-order OS would be the values where the cumulative distribution Φ equal $\frac{1}{3}$ and $\frac{2}{3}$ respectively.

2. For any $n > 2$, implying that we are considering the k^{th}-OS from n samples drawn from $\mathbf{x}$, we can deduce from Eq. (3.1) that:

$$E[\Phi^1(\mathbf{x}_{k,n})] = \frac{k}{n+1}, \implies E[\mathbf{x}_{k,n}] = \Phi^{-1}\left(\frac{k}{n+1}\right), \text{and} \tag{3.5}$$

$$E[\Phi^1(\mathbf{x}_{n-k,n})] = \frac{n-k+1}{n+1}, \implies E[\mathbf{x}_{n-k,n}] = \Phi^{-1}\left(\frac{n-k+1}{n+1}\right). \tag{3.6}$$

   Again, computationally, the first moment of the k^{th} and $n-k^{th}$ n-order OS would be the values where the cumulative distribution Φ equal $\frac{k}{n+1}$ and $\frac{n-k+1}{n+1}$ respectively.

Although the analogous expressions can be derived for the higher order *moments* of these OS, for the rest of this chapter we shall merely focus on the *first* moment of these OS, and derive the consequences of using them in classification.

3.2 Optimal Bayesian Classification using *Two* OS

3.2.1 The Generic Classifier

Having characterized the moments of the OS of arbitrary distributions, we shall now consider how they can be used to design a classifier.

Let us assume that we are dealing with the 2-class problem with classes ω_1 and ω_2, where their class-conditional densities are $f_1(x)$ and $f_2(x)$ respectively (i.e, their corresponding distributions are $F_1(x)$ and $F_2(x)$ respectively)[2]. Let ν_1 and ν_2 be the corresponding *medians* of the distributions. Then, classification based on ν_1 and ν_2 would be the strategy that classifies samples based on a *single* OS. We shall show the fairly straightforward result that for all symmetric distributions, the classification accuracy of this classifier attains the Bayes' accuracy.

This result is not too astonishing because the median is centrally located close to (if not exactly) on the mean. The result for higher order OS is actually far more intriguing because the higher order OS are not located centrally (close to the means), but rather distant from the means. Consequently, we shall show that the classification based on *these* OS again attains the Bayes' bound.

We shall initiate this discussion by examining the Uniform distribution. The reason for this is that even though the distribution itself is rather trivial, the analysis will provide the reader with an insight into the mechanism by which the problem can be tackled, which can then be extended for other distributions.

[2]Throughout this section, we will assume that the *a priori* probabilities are equal. If they are unequal, the above densities must be weighted with the respective *a priori* probabilities.

3.3 The Uniform Distribution

The continuous Uniform distribution is characterized by a constant function $U[a, b]$, where a and b are the minimum and the maximum values that the random variable $\mathbf{x}$ can take. If the class conditional densities of ω_1 and ω_2 are uniformly distributed,

$$f_1(x) = \begin{cases} \frac{1}{b_1 - a_1} & \text{if } a_1 \leq x \leq b_1; \\ 0 & \text{if } x < a_1 \text{ or } x > b_1, \text{ and} \end{cases}$$

$$f_2(x) = \begin{cases} \frac{1}{b_2 - a_2} & \text{if } a_2 \leq x \leq b_2; \\ 0 & \text{if } x < a_2 \text{ or } x > b_2. \end{cases}$$

The reader should observe the following:

- If $a_2 > b_1$, the two distributions are non-overlapping, rendering the classification problem trivial.
- If $a_2 < b_1$, but $b_1 - a_1 \neq b_2 - a_2$, the optimal Bayesian classification is again dependent only on the heights of the distributions. In other words, if $b_2 - a_2 < b_1 - a_1$, the testing sample will be assigned to ω_2 whenever $x > a_2$. This criterion again is not related to the mean of the distributions at all, and is thus un-interesting to our current investigations.
- The meaningful scenario is when $b_1 - a_1$ is exactly equal to $b_2 - a_2$, and if $a_2 < b_1$. In this case, the heights of the two distributions are equal and the distributions are overlapping. This is really the interesting case, and corresponds to the scenario when the two distributions are identical. We shall analyze this in greater detail and demonstrate that the optimal Bayesian classification is also attained by using the OS.

3.3.1 Theoretical Analysis: Uniform Distribution - 2-OS

We shall now derive the formal properties of the classifier that utilizes the OS for the Uniform distribution.

Theorem 2. For the 2-class problem in which the two class conditional distributions are Uniform and identical, CMOS, the classification using two OS, attains the optimal Bayes' bound.

Proof. The proof of the result is done in two steps. We shall first show that when the two class conditional distributions are Uniform and identical, the optimal Bayesian classification is achieved by a comparison to the corresponding *means*. The equivalence of this to a comparison to the corresponding OS leads to the final result.

Without loss of generality let the class conditional distributions for ω_1 and ω_2 be $U[0,1]$ and $U[h,1+h]$, with means $\mu_1 = \frac{1}{2}$ and $\mu_2 = h + \frac{1}{2}$, respectively. In this case, the optimal Bayes' classifier assigns x to ω_1 whenever $x < h$, x to ω_2 whenever $x > 1$, and x to ω_1 and to ω_2 with equal probability when $h < x < 1$. Since:

$$\begin{aligned} D(x,\ \mu_1) < D(x,\ \mu_2) &\iff x - \frac{1}{2} < h + \frac{1}{2} - x \\ &\iff 2x < 1 + h \\ &\iff x < \frac{1+h}{2}, \end{aligned} \tag{3.7}$$

we see that the optimal Bayesian classifier assigns the sample based on the proximity to the corresponding mean, proving the first assertion.

We now consider the moments of the OS of the distributions. If $\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_n$ are n independent univariate random variables that follow the Uniform distribution $U[0,1]$, by virtue of Eq.(3.1), the expected values of the first moment of the k-order OS can be seen to be $E[\mathbf{x}_{k,n}] = \frac{k}{n+1}$. Thus, for $U[0,1]$, $E[\mathbf{x}_{1,2}] = \frac{1}{3}$ and $E[\mathbf{x}_{2,2}] = \frac{2}{3}$. Similarly, for the distribution $U[h,1+h]$, the expected values are $E[\mathbf{x}_{1,2}] = h + \frac{1}{3}$ and $E[\mathbf{x}_{2,2}] = h + \frac{2}{3}$.

The OS-based classification is thus as follows: Whenever a testing sample comes from these distributions, the CMOS will compare the testing sample with $E[\mathbf{x}_{2,2}]$ of the first distribution, i.e., $\frac{2}{3}$, and with $E[\mathbf{x}_{1,2}]$ of the second distribution, i.e., $h + \frac{1}{3}$, and the sample will be labeled with respect to the class which minimizes the corresponding quantity, as shown in Figure 3.1. Observe that for the above rule to work, we must enforce the ordering of the OS of the two distributions, and this requires that $\frac{2}{3} < h + \frac{1}{3} \implies h > \frac{1}{3}$.

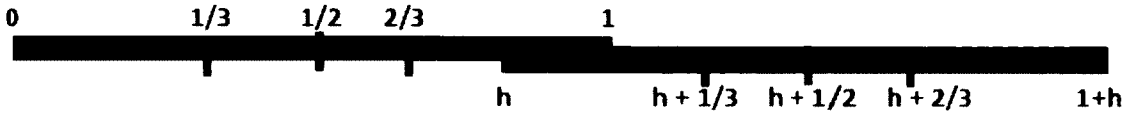


Figure 3.1: The Uniform Distributions of the two classes.

In order to prove that for $h > \frac{1}{3}$ the OS-based classification is identical to the mean-based classification, we have to prove that $D(x, \mu_1) < D(x, \mu_2) \implies D(x, o_1) < D(x, o_2)$, where o_1 is $E[\mathbf{x}_{2,2}]$ of the first distribution and o_2 is $E[\mathbf{x}_{1,2}]$ of the second distribution. By virtue of Eq.(3.7),

$$D(x, \mu_1) < D(x, \mu_2) \iff x < \frac{h+1}{2}. \tag{3.8}$$

Similarly,

$$\begin{aligned} D(x, o_1) < D(x, o_2) &\iff D\left(x, \frac{2}{3}\right) < D\left(x, h + \frac{1}{3}\right) \\ &\iff x - \frac{2}{3} < h + \frac{1}{3} - x \\ &\iff x < \frac{h+1}{2}. \end{aligned} \tag{3.9}$$

The result follows by observing that (3.8) and (3.9) are identical comparisons.

For the analogous result for the case when $h < \frac{1}{3}$, the CMOS will compare the testing sample with $E[\mathbf{x}_{1,2}]$ of the first distribution, i.e., $\frac{1}{3}$, and with $E[\mathbf{x}_{2,2}]$ of the second distribution, i.e., $h + \frac{2}{3}$. Again, the sample will be labeled with respect to the class which minimizes the corresponding quantity. The proofs of the equivalence of this to the Bayesian decision follows along the same lines as the case when $h > \frac{1}{3}$, and is omitted to avoid repetition. Hence the theorem. □

By way of example, consider the distributions $U[0, 1]$ and $U[0.8, 1.8]$. Our claim is demonstrated in Figure 3.2. In the figure, d_1 and d_2 are the distances of the testing sample with respect to the means of the first and the second class (i.e., 0.5 and 1.3)

respectively, and dd_1 and dd_2 are the distances of the testing sample with respect to the moments of the OS for both the classes. The testing sample will be assigned to class ω_1 if $dd_1 < dd_2$ otherwise to class ω_2. The interesting point is that the latter classification is, indeed, the Bayesian conclusion too.

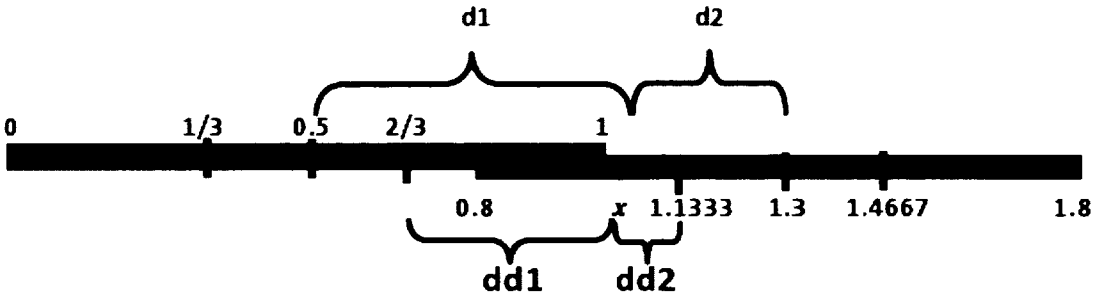


Figure 3.2: Analysis for the Uniform Distribution.

3.3.2 Experimental Results: Uniform Distribution - 2-OS

The CMOS method explained in Section 3.3.1 has been rigorously tested for various Uniform distributions with 2-OS. In the interest of brevity, a few typical results are given below.

For each of the experiments, we generated 1,000 points for the classes ω_1 and ω_2 characterized by $U[0, 1]$ and $U[h, 1 + h]$ respectively. We then invoked a classification procedure by utilizing the Bayesian and the CMOS strategies. In every case, CMOS was compared with the Bayesian classifier for different values of h, as tabulated in Table 3.1. The results in Table 3.1 were obtained by executing each algorithm 50 times using a 10-fold cross-validation scheme.

Observe that in every case, the accuracy of CMOS attained the Bayes' bound.

By way of example, we see that CMOS should obtain the Bayesian bound for the distributions $U[0, 1]$ and $U[0.8, 1.8]$ whenever $n < \frac{1+0.8}{1-0.8} = 9$. In this case, the expected values of the moments are $\frac{1}{10}$ and $\frac{9}{10}$ respectively. These results justify the

h	0.95	0.90	0.85	0.80	0.75	0.70
Bayesian	97.58	95.1	92.42	90.23	87.82	85.4
CMOS	97.58	95.1	92.42	90.23	87.82	85.4

Table 3.1: A comparison of the accuracy of the Bayesian and the 2-OS CMOS classifier for the Uniform Distribution for different values of h.

claim of Theorem 2.

3.3.3 Theoretical Analysis: Uniform Distribution - k-OS

We have seen from Theorem 2 that the moments of the 2-OS are sufficient for the classification to attain a Bayes' bound. We shall now consider the scenario when we utilize other k-OS. The formal result pertaining to this is given in Theorem 3.

Theorem 3. *For the 2-class problem in which the two class conditional distributions are Uniform and identical as U[0,1] and U[h, 1+h], optimal Bayesian classification can be achieved by using symmetric pairs of the n-OS, i.e., the $n-k$ OS for ω_1 and the k OS for ω_2 if and only if $k > \frac{(n+1)(1-h)}{2}$. If $k < \frac{(n+1)(1-h)}{2}$, optimal Bayesian classification can be achieved by using the* Dual *symmetric pairs of the n-OS, i.e., the k OS for ω_1 and the $n-k$ OS for ω_2.*

Proof. We know that for the Uniform distribution $U[0,1]$, the expected values of the first moment of the k-order OS have the form $E[\mathbf{x}_{k,n}] = \frac{k}{n+1}$. Our claim is based on the classification in which we can choose any of the symmetric pairs of the n-OS, i.e., the $n-k$ OS for ω_1 and the k OS for ω_2, whose expected values are $\frac{n-k+1}{n+1}$ and $h+\frac{k}{n+1}$ respectively.

Consider the case when $h > 1-\frac{2k}{n+1}$, the relevance of which will be argued presently. Whenever a testing sample comes, it will be compared with the corresponding k-OS symmetric pairs of the expected values of the n-OS, and the sample will be labeled with respect to the class that minimizes the distance. Observe that for the above rule to work, we must again enforce the ordering of the OS of the two distributions, and

this requires that:

$$\frac{n-k+1}{n+1} < h + \frac{k}{n+1} \implies k > \frac{(n+1)(1-h)}{2}. \tag{3.10}$$

Eq.(3.10) can be seen to be:

$$k > \frac{(n+1)(1-h)}{2} \implies h > 1 - \frac{2k}{n+1}, \tag{3.11}$$

which justifies the case under consideration. As we have already proved that the Bayesian bound can be achieved by a comparison to the corresponding means (in Eq.(3.7)), which in turn simplifies to $x \sim \omega_1 \iff x < \frac{h+1}{2}$, we need to show that to obtain optimal accuracy using these symmetric $n-k$ and k OS, $D(x, o_1) < D(x, o_2) \iff x < \frac{h+1}{2}$. Indeed, the OS-based classification also attains the Bayesian bound because:

$$\begin{aligned} D(x, o_1) < D(x, o_2) &\iff D\left(x, \frac{n-k+1}{n+1}\right) < D\left(x, h + \frac{k}{n+1}\right) \\ &\iff x - \frac{n-k+1}{n+1} < h + \frac{k}{n+1} - x \\ &\iff x < \frac{h+1}{2}. \end{aligned} \tag{3.12}$$

For the symmetric argument when $h < 1 - \frac{2k}{n+1}$, the CMOS will compare the testing sample with $E[\mathbf{x}_{k,n}]$ of the first distribution and $E[\mathbf{x}_{n-k,n}]$ of the second distribution and the classification is obtained based on the class that minimizes *this* distance. The details of the proof are analogous and omitted. Hence the theorem. □

Remark: We can visualize this result from another perspective when we observe that we are concerned about the *ensemble* of symmetric pairs that can be considered to be *effective* for the classification. In order to obtain the maximum accuracy, the expected value of the first moment of the OS for the first class should be less than $\frac{1+h}{2}$, which implies that $\frac{n-k+1}{n+1} < \frac{1+h}{2}$, because if this condition is not satisfied, the Dual CMOS has to be invoked, in which the symmetric pairs should be reversed. Thus:

$$\begin{aligned} \frac{n-k+1}{n+1} < \frac{1+h}{2} &\iff 2n - 2k + 2 < (n+1)(1+h) \\ &\iff k > \frac{(n+1)(1-h)}{2}, \end{aligned} \tag{3.13}$$

which is again the same condition found in the statement of Theorem 3 and Eq. (3.10). This, indeed, implies that the optimal Bayesian bound can be obtained with respect to different symmetric pairs of the n-OS, $\frac{n-k+1}{n+1}$ and $h + \frac{k}{n+1}$, if and only if $k > \frac{(n+1)(1-h)}{2}$.

3.3.4 Experimental Results: Uniform Distribution - k-OS

The CMOS method has also been tested for the Uniform distribution for other k OS. In the interest of brevity, we merely cite one example where the distributions for ω_1 and ω_2 were characterized by $U[0, 1]$ and $U[0.8, 1.8]$ respectively. For each of the experiments, we generated 1,000 points for each class, and the testing samples were classified based on the selected *symmetric* pairs for values k and $n - k$ respectively. The results are displayed in Table 3.2.

To clarify the table, consider the row given by Trial No. 6 in which the 7-OS were invoked for the classification. Observe that the k-OS are now given by $\frac{n-k+1}{n+1}$ and $\frac{k}{n+1}$ respectively. In this case, the possible symmetric OS pairs could be $\langle 1, 6\rangle$, $\langle 2, 5\rangle$, and $\langle 3, 4\rangle$ respectively. In every single case, the accuracy attained the Bayes' bound, as indicated by the results in the table.

The importance of the condition imposed by Theorem 3 can be seen from the results given in the row denoted by Trial No. 9. In this case, the testing attained the Bayes' accuracy for the symmetric OS pairs $\langle 2, 9\rangle$, $\langle 3, 8\rangle$, $\langle 4, 7\rangle$ and $\langle 5, 6\rangle$ respectively. However, for the specific 10-OS, when the OS used were $\frac{10}{11}$ and $h+\frac{1}{11}$, the Dual CMOS has to be invoked as these values did not satisfy the condition $h > 1 - \frac{2k}{n+1}$. In such a case, the symmetric pairs should be reversed, i.e., $\frac{k}{n+1}$ for the first distribution, and $h + \frac{n-k+1}{n+1}$ for the second distribution, to obtain the optimal Bayesian bound. The astonishing facet of this result is that one obtains the Bayes' accuracy even though the classification requires only *two* points distant from the mean, justifying the rationale for BI schemes, and yet operating in an anti-Bayesian manner.

Trial No.	Order(n)	Moments	OS_1	OS_2	CMOS	CMOS/ Dual CMOS
1	Two	$\{\frac{i}{3} \mid 1 \leq i \leq 2\}$	$\frac{2}{3}$	$h + \frac{1}{3}$	90.23	CMOS
2	Three	$\{\frac{i}{4} \mid 1 \leq i \leq 3\}$	$\frac{3}{4}$	$h + \frac{1}{4}$	90.23	CMOS
3	Four	$\{\frac{i}{5} \mid 1 \leq i \leq 4\}$	$\frac{4}{5}$	h + $\frac{1}{5}$	90.23	CMOS
4	Five	$\{\frac{i}{6} \mid 1 \leq i \leq 5\}$	$\frac{4}{6}$	$h + \frac{2}{6}$	90.23	CMOS
5	Six	$\{\frac{i}{7} \mid 1 \leq i \leq 6\}$	$\frac{4}{7}$	$h + \frac{3}{7}$	90.23	CMOS
6	Seven	$\{\frac{i}{8} \mid 1 \leq i \leq 7\}$	$\frac{5}{8}$	$h + \frac{3}{8}$	90.23	CMOS
7	Eight	$\{\frac{i}{9} \mid 1 \leq i \leq 8\}$	$\frac{6}{9}$	$h + \frac{3}{9}$	90.23	CMOS
8	Nine	$\{\frac{i}{10} \mid 1 \leq i \leq 9\}$	$\frac{7}{10}$	$h + \frac{3}{10}$	90.23	CMOS
9	Ten	$\{\frac{i}{11} \mid 1 \leq i \leq 10\}$	$\frac{10}{11}$	$h + \frac{1}{11}$	90.23	Dual CMOS
10	Ten	$\{\frac{i}{11} \mid 1 \leq i \leq 10\}$	$\frac{9}{11}$	$h + \frac{2}{11}$	90.23	CMOS
11	Ten	$\{\frac{i}{11} \mid 1 \leq i \leq 10\}$	$\frac{7}{11}$	$h + \frac{4}{11}$	90.23	CMOS
12	Ten	$\{\frac{i}{11} \mid 1 \leq i \leq 10\}$	$\frac{6}{11}$	$h + \frac{5}{11}$	90.23	CMOS

Table 3.2: Results of the classification of Uniformly distributed classes obtained by using the symmetric pairs of the OS for different values of n. The value of h was set to be 0.8. Note that in every case, the accuracy attained the Bayes' value whenever the conditions stated in Theorem 3 were satisfied.

3.4 Conclusions

In this chapter, we have presented a novel approach to the age-old problem of pattern classification, namely, by using a non-traditional "anti-Bayesian" approach. We have shown that the optimal Bayes' bound can be obtained by such an "anti-Bayesian" strategy, which we have referred to as Classification by Moments of Order Statistics (CMOS). To be more specific, we have proved that the classification can be achieved by working with a *very few* (sometimes as small as two) points *distant* from the mean. Further, if these points are determined by the *Order Statistics* of the distributions, the optimal Bayes' bound can be attained. After proving some fundamental results

concerning OS, we have derived the conditions for a "generic" uni-dimensional classifier. The claim has then been proven for the uni-dimensional Uniform distribution. The reason for considering the Uniform distribution is that even though the distribution *itself* is rather trivial, the analysis will provide the reader with an insight into the mechanism by which the problem can be tackled, which can then be extended for other distributions within the exponential family. The theoretical results have been verified by rigorous experimental testing.

The results presented here naturally lead to the problems considered in the next chapter, namely the design, implementation and testing of CMOS schemes for other distributions within the exponential family.

Chapter 4

Optimal "Anti-Bayesian" OS-based Parametric PR for Symmetric Distributions in the Exponential Family

4.1 Introduction

In the previous chapter, we had argued and proved that the Bayes' bound can be attained by working with a *very few* (sometimes as small as two) points which are *distant* from the mean and are obtained based on the OS criteria. The claim was proven for a generic classifier and the uni-dimensional Uniform distribution which was used as a *prima facie* case because the fundamental principles are also valid for more complicated distributions. In this chapter[1], we consider extending the results for more realistic distributions from the exponential family, namely the Doubly-Exponential,

[1] A preliminary version of some of the results of this chapter can be found in the in the *Proceedings of ICIAR2012, the* 9^{th} *International Conference on Image Analysis and Recognition* held in Portugal in June 2012 [46]. More detailed descriptions of these results have been published in the journal *Pattern Recognition* [38].

the Gaussian and a form of Beta distribution. This hypothesis, that CMOS classification can be achieved, has been formally proved and experimentally verified. Again, as in the case of the Uniform distribution, we have worked with 2-OS and extended the results for k-OS, and considered the scenarios where the Dual CMOS has to be invoked. Our results for classification using the OS are both pioneering and novel to the best of our knowledge. Again, the beauty of this approach is that out of the entire training set, only 2 points (for uni-dimensional distributions) are sufficient to achieve a classification that attains the Bayes' bound.

4.1.1 Contributions of this Chapter

The novel contributions of this chapter are the following:

- We extend the "anti-Bayesian" generic paradigm of Section 3.2 for the classification of uni-dimensional symmetric distributions within the exponential family, and show that the strategy requires a very few number of non-central training patterns;
- To justify these claims, we submit a formal analysis for the Laplace, Gaussian and a form of the Beta distributions, by invoking the 2-OS CMOS and the k-OS CMOS. Further, for the k-OS CMOS, the condition at which the Dual CMOS has to be invoked is also thoroughly investigated.
- The claims are experimentally verified by rigorous testing for uni-dimensional Laplace, Gaussian and a form of Beta distributions, and the results are clearly conclusive.

4.2 The Laplace (or Doubly-Exponential) Distribution

The *Laplace distribution* is a continuous uni-dimensional pdf named after Pierre-Simon Laplace. It is sometimes called the *Doubly-Exponential distribution*, because

it can be perceived as being a combination of two exponential distributions, with an additional location parameter, spliced together back-to-back.

If the class conditional densities of ω_1 and ω_2 are Doubly-Exponentially distributed,

$$f_1(x) = \frac{\lambda_1}{2} e^{-\lambda_1 |x - c_1|}, \quad -\infty < x < \infty, \text{and}$$

$$f_2(x) = \frac{\lambda_2}{2} e^{-\lambda_2 |x - c_2|}, \quad -\infty < x < \infty,$$

where c_1 and c_2 are the respective means of the distributions. By elementary integration and straightforward algebraic simplifications, the variances of the distributions can be seen to be $\frac{2}{\lambda_1^2}$ and $\frac{2}{\lambda_2^2}$ respectively.

By way of example, the pdfs of Doubly-Exponential distributions for different values of the parameter λ are given in Figure 4.1 where the optimal Bayes' classifier will evidently be at the point x^*. Thus, if $\lambda_1 \neq \lambda_2$, the samples can be classified based on the heights of the distributions and their point of intersection. The formal results for the general case are a little more complex. However, to prove the analogous results of Theorem 2 of the Uniform distribution, we shall first consider the case when $\lambda_1 = \lambda_2$. In this scenario, the reader should observe the following:

- Because the distributions have the equal height, i.e. $\lambda_1 = \lambda_2$, the testing sample $\mathbf{x}$ will obviously be assigned to ω_1 if it is less than c_1 and be assigned to ω_2 if it is greater than c_2.

- Further, the crucial case is when $c_1 < x < c_2$. In this regard, we shall analyze the CMOS classifier and prove that it attains the Bayes' bound even when one uses as few as *only* 2 OSs.

4.2.1 Theoretical Analysis: Doubly-Exponential Distribution - 2-OS

We shall first derive the moments of the 2-OS for the Doubly-Exponential distribution. By virtue of Eq. (3.3) and (3.4), the expected values of the first moments of the two OS can be obtained by determining the points where the cumulative distribution

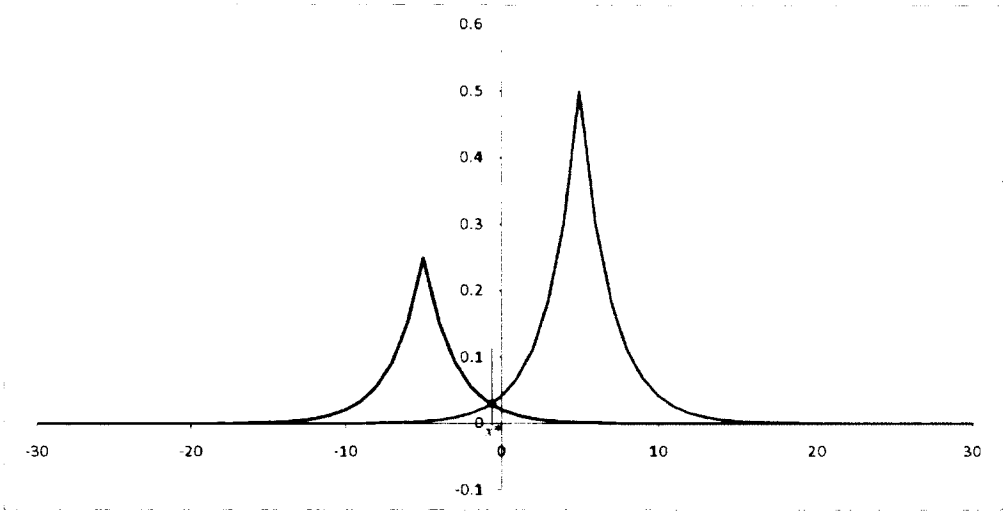


Figure 4.1: Doubly-Exponential Distributions for different values for λ.

function attains the values $\frac{1}{3}$ and $\frac{2}{3}$. Let o_1 be the point for the percentile $\frac{2}{3}$ of the first distribution, and o_2 be the point for the percentile $\frac{1}{3}$ of the second distribution. Then:

$$\int_{c_1}^{o_1} \frac{\lambda_1}{2} e^{-\lambda_1|x-c_1|}dx = \frac{2}{3} - \frac{1}{2} = \frac{1}{6}, \text{ and} \tag{4.1}$$

$$\int_{-\infty}^{o_2} \frac{\lambda_2}{2} e^{\lambda_2|x-c_2|}dx = \frac{1}{3}. \tag{4.2}$$

The points of interest, i.e., o_1 and o_2, can be obtained by straightforward integrations and simplifications as follows:

$$\int_{c_1}^{o_1} \frac{\lambda_1}{2} e^{-\lambda_1|x-c_1|}dx = \frac{1}{6} \implies \left[-\frac{1}{2}e^{-\lambda_1(x-c_1)}\right]_{c_1}^{o_1} = \frac{1}{6}$$

$$\implies o_1 = c_1 - \frac{1}{\lambda_1} ln\left(\frac{2}{3}\right). \tag{4.3}$$

Using a similar argument, we can see that:

$$o_2 = c_2 + \frac{1}{\lambda_2} ln\left(\frac{2}{3}\right). \tag{4.4}$$

With these points at hand, we shall now demonstrate that, for Doubly-Exponential distributions, the classification based on the expected values of the moments of the 2-OS, CMOS, attains the Bayesian bound.

Theorem 4. *For the 2-class problem in which the two class conditional distributions are Doubly-Exponential and identical, CMOS, the classification using two OS, attains the optimal Bayes' bound.*

Proof. The proof can be done in two steps. As in the Uniform case, first of all, we shall show that when the class conditional distributions are Doubly-Exponential and identical, the optimal Bayes' bound can be attained by a comparison to the corresponding means, and as the concluding step, this can be shown to be equal to the accuracy of the CMOS, which lead to the proof of the theorem.

Without loss of generality, let the distributions of ω_1 and ω_2 be $D(c_1, \lambda)$ and $D(c_2, \lambda)$, where c_1 and c_2 are the means, and λ is the identical scale parameter. Then, to get the Bayes' classifier, we argue that:

$$\begin{aligned} p(x|\omega_1)P(\omega_1) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} p(x|\omega_2)P(\omega_2) &\implies \frac{\lambda}{2} e^{-\lambda|x-c_1|} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{\lambda}{2} e^{-\lambda|x-c_2|} \\ &\implies \lambda(x-c_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \lambda(c_2-x) \\ &\implies x \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \frac{c_1+c_2}{2}. \end{aligned} \tag{4.5}$$

We now consider the classification with respect to the expected values of the moments of the 2-OS, o_1 and o_2, where as per Eq. (4.3) and (4.4), $o_1 = c_1 - \frac{1}{\lambda} ln\left(\frac{2}{3}\right)$ and $o_2 = c_2 + \frac{1}{\lambda} ln\left(\frac{2}{3}\right)$. In order to prove our claim, we need to show that

$$D(x, c_1) < D(x, c_2) \iff D(x, o_1) < D(x, o_2). \tag{4.6}$$

We first consider the LHS of Eq. (4.6). Indeed,

$$\begin{aligned} D(x, c_1) < D(x, c_2) &\implies x - c_1 < c_2 - x \\ &\implies 2x < c_1 + c_2 \\ &\implies x < \frac{c_1+c_2}{2}. \end{aligned} \tag{4.7}$$

What remains to be proven is that the RHS of Eq. (4.6) also simplifies to the same expression. This is true because:

$$\begin{aligned} D(x,o_1) < D(x,o_2) &\implies D\left(x,\ c_1 - \frac{1}{\lambda}\, ln\left(\frac{2}{3}\right)\right) < D\left(x,\ c_2 + \frac{1}{\lambda}\, ln\left(\frac{2}{3}\right)\right) \\ &\implies 2x < c_1 + c_2 \\ &\implies x < \frac{c_1 + c_2}{2}. \end{aligned} \tag{4.8}$$

The result follows by observing that Eq. (4.7) and (4.8) are identical comparisons. Hence the theorem. □

4.2.2 Data Generation: Doubly-Exponential Distribution

In order to generate data that follow non-uniform distributions, we made use of a Uniform $[0, 1]$ random variate generator. Data values that follow a Doubly-Exponential distribution can be generated by using the expression $x = c \pm \lambda ln|2u|$ where c is the mean of the distribution, λ is the scale parameter, and u is Uniform in $U[0, 1]$ [7]. For both the classes, 1,000 points were generated with means c_1 and c_2, and with identical values for λ_1 and λ_2.

4.2.3 Experimental Results: Doubly-Exponential Distribution - 2OS

The CMOS classifier was rigorously tested for a number of experiments with various Doubly-Exponential distributions having means c_1 and c_2. In every case, the 2-OS CMOS gave exactly the same classification as that of the Bayesian classifier. The method was executed 50 times with the 10-fold cross validation scheme. The test results are depicted in Table 4.1. From the experimental results and the theoretical analysis, we conclude that the expected values of the first moment of the 2-OS of the Doubly-Exponential distribution can always be utilized to yield the exact accuracy as that of the Bayes' bound, even though this is a drastically anti-Bayesian operation.

We now proceed to consider the analogous result for the k-OS.

c_1	0	0	0	0	0	0	0	0	0
c_2	10	9	8	7	6	5	4	3	2
Bayesian	99.75	99.65	99.25	99.05	98.9	97.85	96.8	94.05	89.9
CMOS	99.75	99.65	99.25	99.05	98.9	97.85	96.8	94.05	89.9

Table 4.1: A comparison of the accuracy of the Bayesian and the 2-OS CMOS classifier for the Doubly-Exponential Distribution.

4.2.4 Theoretical Analysis: Doubly-Exponential Distribution - k-OS

We have seen from Theorem 4 that for the Doubly-Exponential distribution, the moments of the 2-OS are sufficient for the classification to attain a Bayes' bound. We shall now consider the scenario when we utilize other k-OS. The formal result pertaining to this is given in Theorem 5.

Theorem 5. *For the 2-class problem in which the two class conditional distributions are Doubly-Exponential and identical, the optimal Bayesian classification can be achieved by using symmetric pairs of the n-OS, i.e., the $n-k$ OS for ω_1 and the k OS for ω_2 if and only if $ln\left(\frac{2k}{n+1}\right) > \frac{c_1-c_2}{2}$. Again, if the latter condition is violated, optimal Bayesian classification can be achieved by using the* Dual *symmetric pairs of the n-OS, i.e., the k OS for ω_1 and the $n-k$ OS for ω_2.*

Proof. We shall first show that the expected values of the first moment of the k-order OS for the Doubly-Exponential distribution have the form $E[\mathbf{x}_{k,n}] = ln\left(\frac{2k}{n+1}\right)$. This result is proven by invoking a formal mathematical induction on k and is omitted here for the present.

We have already solved the case when $n = 2$ and can be seen in Section 4.2.1. Now, we shall consider the case when $n = 4$, for which the possible symmetric OS pairs could be $\langle 1, 4\rangle$ and $\langle 2, 3\rangle$ respectively. Considering the OS pair $\langle 1, 4\rangle$, let o_1 be the point for the percentile $\frac{4}{5}$ of the first distribution, and o_2 be the point for the

percentile $\frac{1}{5}$ of the second distribution. Then:

$$\int_{c_1}^{o_1} \frac{\lambda_1}{2} e^{-\lambda_1|x-c_1|}dx = \frac{4}{5} - \frac{1}{2} = \frac{3}{5}, \text{ and} \tag{4.9}$$

$$\int_{-\infty}^{o_2} \frac{\lambda_2}{2} e^{\lambda_2|x-c_2|}dx = \frac{1}{5}. \tag{4.10}$$

By straightforward integration and simplifications, we obtain:

$$o_1 = c_1 - \frac{1}{\lambda_1} ln\left(\frac{2}{5}\right), o_2 = c_2 + \frac{1}{\lambda_2} ln\left(\frac{2}{5}\right). \tag{4.11}$$

Arguing in the same way, if we consider the symmetric pair $\langle 2, 3\rangle$, we obtain:

$$o_1 = c_1 - \frac{1}{\lambda_1} ln\left(\frac{4}{5}\right), o_2 = c_2 + \frac{1}{\lambda_2} ln\left(\frac{4}{5}\right). \tag{4.12}$$

The CMOS points for different values for n is given in Table 4.2.

n	OS percentiles	o_1	o_2
2	$\langle\frac{2}{3}, \frac{1}{3}\rangle$	$c_1 - \frac{1}{\lambda} ln\left(\frac{2}{3}\right)$	$c_2 + \frac{1}{\lambda} ln\left(\frac{2}{3}\right)$
4	$\langle\frac{4}{5}, \frac{1}{5}\rangle$	$c_1 - \frac{1}{\lambda} ln\left(\frac{2}{5}\right)$	$c_2 + \frac{1}{\lambda} ln\left(\frac{2}{5}\right)$
4	$\langle\frac{3}{5}, \frac{2}{5}\rangle$	$c_1 - \frac{1}{\lambda} ln\left(\frac{4}{5}\right)$	$c_2 + \frac{1}{\lambda} ln\left(\frac{4}{5}\right)$
6	$\langle\frac{6}{7}, \frac{1}{7}\rangle$	$c_1 - \frac{1}{\lambda} ln\left(\frac{2}{7}\right)$	$c_2 + \frac{1}{\lambda} ln\left(\frac{2}{7}\right)$
6	$\langle\frac{5}{7}, \frac{2}{7}\rangle$	$c_1 - \frac{1}{\lambda} ln\left(\frac{4}{7}\right)$	$c_2 + \frac{1}{\lambda} ln\left(\frac{4}{7}\right)$
6	$\langle\frac{4}{7}, \frac{3}{7}\rangle$	$c_1 - \frac{1}{\lambda} ln\left(\frac{6}{7}\right)$	$c_2 + \frac{1}{\lambda} ln\left(\frac{6}{7}\right)$
8	$\langle\frac{8}{9}, \frac{1}{9}\rangle$	$c_1 - \frac{1}{\lambda} ln\left(\frac{2}{9}\right)$	$c_2 + \frac{1}{\lambda} ln\left(\frac{2}{9}\right)$
8	$\langle\frac{7}{9}, \frac{2}{9}\rangle$	$c_1 - \frac{1}{\lambda} ln\left(\frac{4}{9}\right)$	$c_2 + \frac{1}{\lambda} ln\left(\frac{4}{9}\right)$
8	$\langle\frac{6}{9}, \frac{3}{9}\rangle$	$c_1 - \frac{1}{\lambda} ln\left(\frac{6}{9}\right)$	$c_2 + \frac{1}{\lambda} ln\left(\frac{6}{9}\right)$
8	$\langle\frac{5}{9}, \frac{4}{9}\rangle$	$c_1 - \frac{1}{\lambda} ln\left(\frac{8}{9}\right)$	$c_2 + \frac{1}{\lambda} ln\left(\frac{8}{9}\right)$

Table 4.2: CMOS values for different values of n.

Our present claim is based on the classification in which we can choose any of the symmetric pairs of the n-OS, i.e., the $n-k$ OS for ω_1 and the k OS for ω_2, where these quantities are $c_1 - ln\left(\frac{2k}{n+1}\right)$ and $c_2 + ln\left(\frac{2k}{n+1}\right)$ respectively.

It is obvious that an OS value can correctly classify a testing point only when the position is somewhere before the intersection of the curves. This point of intersection of the curves can be obtained by equating them as below:

$$\begin{aligned}
\frac{\lambda_1}{2}e^{-\lambda_1(x-c_1)} = \frac{\lambda_2}{2}e^{-\lambda_2(x-c_2)} \quad &\Longrightarrow \quad \frac{\lambda_1}{\lambda_2} = \frac{e^{\lambda_2(x-c_2)}}{e^{-\lambda_1(x-c_1)}} \\
&\Longrightarrow \quad \frac{\lambda_1}{\lambda_2} = e^{\lambda_1(x-c_1)+\lambda_2(x-c_2)} \\
&\Longrightarrow \quad ln\left(\frac{\lambda_1}{\lambda_2}\right) = x(\lambda_1+\lambda_2) - \lambda_1 c_1 - \lambda_2 c_2 \\
&\Longrightarrow \quad x = \frac{\lambda_1 c_1 + \lambda_2 c_2 + ln\left(\frac{\lambda_1}{\lambda_2}\right)}{\lambda_1+\lambda_2}. \qquad (4.13)
\end{aligned}$$

Observe that this equality will reduce to $\frac{c_1+c_2}{2}$ when $\lambda_1 = \lambda_2$.

In order to prove the bounds specified in the statement of the theorem, we enforce the ordering of the OS of the distributions as:

$$c_1 - ln\left(\frac{2k}{n+1}\right) < \frac{c_1+c_2}{2} < c_2 + ln\left(\frac{2k}{n+1}\right). \qquad (4.14)$$

The LHS of Eq. (4.14) can easily be simplified to:

$$\begin{aligned}
c_1 - ln\left(\frac{2k}{n+1}\right) < \frac{c_1+c_2}{2} < c_2 \quad &\Longrightarrow \quad c_1 - \frac{c_1+c_2}{2} < c_2 < ln\left(\frac{2k}{n+1}\right) \\
&\Longrightarrow \quad ln\left(\frac{2k}{n+1}\right) > \frac{c_1-c_2}{2}. \qquad (4.15)
\end{aligned}$$

The RHS of Eq. (4.14) can also be simplified to the same expression, for which the algebraic details are omitted.

The fact that the scheme attains the Bayes' accuracy when these bounds are

enforced is now demonstrated by observing that:

$$\begin{aligned} D(x,o_1) < D(x,o_2) &\implies D\left(x, c_1 - ln\left(\frac{2k}{n+1}\right)\right) < D\left(x, c_2 + ln\left(\frac{2k}{n+1}\right)\right) \\ &\implies x - \left(c_1 - ln\left(\frac{2k}{n+1}\right)\right) < \left(c_2 - ln\left(\frac{2k}{n+1}\right)\right) - x \\ &\implies x < \frac{c_1 + c_2}{2}. \end{aligned} \tag{4.16}$$

The fact that the Dual criterion is valid when the condition is not satisfied can also be proven with identical arguments, and the details are again omitted to avoid repetition. Hence the theorem. □

4.2.5 Experimental Results: Doubly-Exponential Distribution - k-OS

The CMOS method has been rigorously tested with different possibilities of k-OS and for various values of n, and the test results are given in Table 4.3.

No.	Order(n)	Moments	OS_1	OS_2	CMOS	CMOS/ Dual CMOS
1	Two	$(\frac{2}{3}, \frac{1}{3})$	$c_1 - \frac{1}{\lambda_1}\log(\frac{2}{3})$	$c_2 + \frac{1}{\lambda_2}\log(\frac{2}{3})$	95.2	CMOS
2	Three	$(\frac{3}{4}, \frac{1}{4})$	$c_1 - \frac{1}{\lambda_1}\log(\frac{1}{2})$	$c_2 + \frac{1}{\lambda_2}\log(\frac{1}{2})$	95.2	CMOS
3	Four	$(\frac{5-i}{5}, \frac{i}{5}), 1 \le i \le \frac{n}{2}$	$c_1 - \frac{1}{\lambda_1}\log(\frac{4}{5})$	$c_2 + \frac{1}{\lambda_2}\log(\frac{4}{5})$	95.2	CMOS
4	Five	$(\frac{6-i}{6}, \frac{i}{6}), 1 \le i \le \frac{n}{2}$	$c_1 - \frac{1}{\lambda_1}\log(\frac{1}{3})$	$c_2 + \frac{1}{\lambda_2}\log(\frac{1}{3})$	95.2	CMOS
5	Six	$(\frac{7-i}{7}, \frac{i}{7}) 1 \le i \le \frac{n}{2}$	$c_1 - \frac{1}{\lambda_1}\log(\frac{4}{7})$	$c_2 + \frac{1}{\lambda_2}\log(\frac{4}{7})$	95.2	CMOS
6	Seven	$(\frac{8-i}{8}, \frac{i}{8}) 1 \le i \le \frac{n}{2}$	$c_1 - \frac{1}{\lambda_1}\log(\frac{1}{4})$	$c_2 + \frac{1}{\lambda_2}\log(\frac{1}{4})$	95.2	CMOS
7	Eight	$(\frac{9-i}{9}, \frac{i}{9}), 1 \le i \le \frac{n}{2}$	$c_1 - \frac{1}{\lambda_1}\log(\frac{2}{9})$	$c_2 + \frac{1}{\lambda_2}\log(\frac{2}{9})$	95.2	Dual CMOS
8	Eight	$(\frac{9-i}{9}, \frac{i}{9}), 1 \le i \le \frac{n}{2}$	$c_1 - \frac{1}{\lambda_1}\log(\frac{4}{9})$	$c_2 + \frac{1}{\lambda_2}\log(\frac{4}{9})$	95.2	CMOS
9	Nine	$(\frac{10-i}{10}, \frac{i}{10}), 1 \le i \le \frac{n}{2}$	$c_1 - \frac{1}{\lambda_1}\log(\frac{3}{5})$	$c_2 + \frac{1}{\lambda_2}\log(\frac{3}{5})$	95.2	CMOS

Table 4.3: Results of the classification obtained by using the symmetric pairs of the OS for different values of n. The value of c_1 and c_2 were set to be 0 and 3. Note that in every case, the accuracy attained the Bayes' value whenever the conditions stated in Theorem 5 were satisfied.

To clarify the table, consider the row given by Trial No. 5 in which the 6-OS were invoked for the classification. In this case, the possible symmetric OS pairs could

be $\langle 1,6\rangle$, $\langle 2,5\rangle$, and $\langle 3,4\rangle$ respectively. Observe that the expected values for the first moment of the k-OS has the form $E[\mathbf{x}_{k,n}] = ln\left(\frac{2k}{n+1}\right)$. In every single case, the accuracy attained the Bayes' bound, as indicated by the results in the table.

Now, consider the results presented in the row denoted by Trial No. 7. In this case, the testing attained the Bayes' accuracy for the symmetric OS pairs $\langle 2,7\rangle$, $\langle 3,6\rangle$ and $\langle 4,5\rangle$ respectively. However, for the specific 8-OS, when the OS used were $c_1 - \frac{1}{\lambda_1} ln\left(\frac{2}{9}\right)$ and $c_2 + \frac{1}{\lambda_2} ln\left(\frac{2}{9}\right)$, as these values violate the condition $ln\left(\frac{2k}{n+1}\right) > \frac{c_1-c_2}{2}$, imposed by Theorem 5, the Dual CMOS has to be invoked. In such a case where $ln\left(\frac{2k}{n+1}\right) < \frac{c_1-c_2}{2}$, the symmetric pairs should be reversed to obtain the optimal Bayes' bound.

This concludes our discussion on the use of OS for the PR of features obeying a Doubly-Exponential distribution.

4.3 The Gaussian Distribution

The Normal (or Gaussian) distribution is a continuous probability distribution that is often used as a first approximation to describe real-valued random variables that tend to cluster around a single mean value. It is particularly pertinent due to the so-called Central Limit Theorem. The univariate pdf of the distribution is:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}.$$

The Gaussian curves for two sets of values for μ and σ is given in Figure 4.2. As is well known, the optimal Bayesian classifier for equiprobable classes is determined by the point of intersection of the curves, i.e., x^*.

We shall now consider the consequence of utilizing CMOS with the 2-OS for classification and again argue the strength of the anti-Bayesian method.

4.3.1 Theoretical Analysis: Gaussian Distribution

Working with the OS of *Normal* distributions is extremely cumbersome because its density function is not integrable in a closed form. One has to resort to tabulated

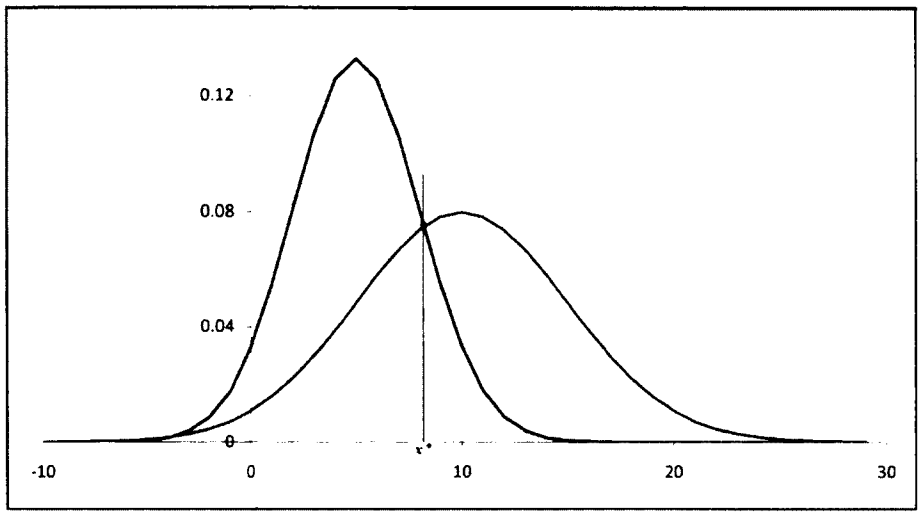


Figure 4.2: The Gaussian Distribution for different means and variances.

cumulative error functions or to numerical methods to obtain precise percentile values. However, a lot of work has been done in this area for *certain* OS, and can be found in [1, 19, 34, 37], from which we can make some interesting conclusions.

The moments of the OS for the Normal distribution can be determined from the generalized expression:

$$E[\mathbf{x}_{k,n}^r] = \frac{n!}{(k-1)!(n-k)!} \int_{-\infty}^{+\infty} x^r \Phi^{k-1}(x)(1-\Phi(x))^{n-k}\varphi(x)dx,$$

where $\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$ and $\Phi(x) = \int_{-\infty}^{x} \varphi(t)dt$. From this expression, the expected values of the first moment of the 2-OS can be determined as:

$$E[\mathbf{x}_{1,2}] = \mu - \frac{\sigma}{\sqrt{2\pi}}, \quad \text{and} \tag{4.17}$$

$$E[\mathbf{x}_{2,2}] = \mu + \frac{\sigma}{\sqrt{2\pi}}, \tag{4.18}$$

as shown in [1]. Using this result, we now show that for identically distributed classes differing only in the means, the CMOS with 2-OS yields the same Bayesian accuracy, which is the primary thrust of this thesis.

Theorem 6. *For the 2-class problem in which the two class conditional distributions*

are Gaussian and identical, CMOS, the classification using 2-OS, attains the optimal Bayes' bound.

Proof. As in the previous cases, we shall first show that when the class conditional distributions are Gaussian and identical, the optimal Bayes' bound can be attained by a comparison to the corresponding means, which can then be shown to be equal to the accuracy of the CMOS, whence the theorem is proven.

Without loss of generality, let ω_1 and ω_2 be two classes that follow the Gaussian distribution with means μ_1 and μ_2, and with equal standard deviations, σ. Let o_1 and o_2 be the first moments of the 2-OS, where $o_1 = \mu_1 - \frac{\sigma}{\sqrt{2\pi}}$ and $o_2 = \mu_2 + \frac{\sigma}{\sqrt{2\pi}}$. It is well known that for this scenario the optimal Bayes' classifier can be obtained as:

$$\begin{aligned} p(x|\omega_1)P(\omega_1) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} p(x|\omega_2)P(\omega_2) &\implies \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu_1)^2}{2\sigma^2}} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu_2)^2}{2\sigma^2}} \\ &\implies x - \mu_1 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \mu_2 - x \\ &\implies x \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \frac{\mu_1 + \mu_2}{2}. \end{aligned} \tag{4.19}$$

We now prove that: $D(x, \mu_1) < D(x, \mu_2) \implies D(x, o_1) < D(x, o_2)$.

We first consider the LHS of the claim. Then,

$$\begin{aligned} D(x, \mu_1) < D(x, \mu_2) &\implies x - \mu_1 < \mu_2 - x \\ &\implies 2x < \mu_1 + \mu_2 \\ &\implies x < \frac{\mu_1 + \mu_2}{2}. \end{aligned} \tag{4.20}$$

For the result to be proved, we have to also prove that the RHS simplifies to the same expression. Indeed, this is true because,

$$\begin{aligned} D(x,\ o_1) < D(x,\ o_2) \\ &\implies D\left(x,\ \mu_1 - \frac{\sigma}{\sqrt{2\pi}}\right) < D\left(x,\ \mu_2 + \frac{\sigma}{\sqrt{2\pi}}\right) \\ &\implies x - \left(\mu_1 - \frac{\sigma}{\sqrt{2\pi}}\right) < \left(\mu_2 + \frac{\sigma}{\sqrt{2\pi}}\right) - x \\ &\implies 2x < \mu_1 + \mu_2 \\ &\implies x < \frac{\mu_1 + \mu_2}{2}. \end{aligned} \tag{4.21}$$

The theorem follows. □

4.3.2 Data Generation: Gaussian Distribution

As in the previous cases, we made use of a Uniform $[0, 1]$ random variable generator to generate data values that follow a Gaussian distribution. The expression $z = \sqrt{-2ln(u_1)}\ cos(2\pi u_2)$, is known to yield data values that follow $N(0, 1)$ [7], from which the data values that follow $N(\mu, \sigma)$ can be generated as $x = \mu + z\sigma$, where μ is the mean and σ is the standard deviation of the required distribution. For both the classes, 1,000 points were generated with means μ_1 and μ_2, and with identical values for σ_1 and σ_2.

4.3.3 Experimental Results: Gaussian Distribution

After the data points were generated, the CMOS classifier was rigorously tested for a number of experiments with various Gaussian distributions having means μ_1 and μ_2. In every case, the 2-OS CMOS gave *exactly* the same accuracy as that of the Bayesian classifier. The method was executed 50 times with the 10-fold cross validation scheme. The test results are displayed in Table 4.4, whence the power of the scheme is clear.

μ_1	0	0	0	0	0	0
μ_2	14	12	10	8	6	4
Bayesian	99.2	96.5	95.1	95	90	85
CMOS	99.2	96.5	95.1	95	90	85

Table 4.4: A comparison of the accuracy of the Bayesian and the 2-OS CMOS classifier for the Gaussian Distribution.

We believe that the optimal Bayes' bound can also be attained by performing the classification with respect to the k-OS. However, as the density function is not integrable, the expected values of the moments of the k-OS should rather be obtained by invoking a numerical integration. It can be easily seen that the error function given by:

$$erf(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{\frac{-t^2}{2}} dt, \tag{4.22}$$

is related to Φ, the cumulative distribution function of the Normal density as per:

$$\Phi(x) = \frac{1}{2}\left[1 + erf\left(\frac{x}{\sqrt{2}}\right)\right]. \tag{4.23}$$

The inverse of Φ, named as the *probit* function or the normal quantile function, is expressed as:

$$probit(p) = \sqrt{2}\ erf^{-1}(2p-1). \tag{4.24}$$

In order to get the exact values of the $\left(\frac{k}{n}\right)^{th}$ percentile, the value computed as per Eq. (4.22 - 4.24) should be interpolated.

Alternatively, the $\left(\frac{k}{n}\right)^{th}$ percentile of the normal distribution can be obtained [7] by making use of the inverse of the normal distribution function as:

$$g(u) = \sqrt{-2\ lnu} \cdot \frac{A\sqrt{-2\ lnu}}{B\sqrt{-2\ lnu}}, \tag{4.25}$$

where $A(x) = \sum_{i=0}^{4} a_i x^i$, $B(x) = \sum_{i=0}^{4} b_i x^i$, and where the coefficients are as shown in Table 4.5.

i	a_i	b_i
0	-0.3222232431088	0.0993484626060
1	-1.0	0.588581570495
2	-0.342242088547	0.531103462366
3	-0.0204231210245	0.103537752850
4	-0.0000453642210148	0.0038560700634

Table 4.5: Coefficients for the Inverse Normal function.

One can easily see that the $\left(\frac{k}{n}\right)^{th}$ and the $\left(\frac{n-k+1}{n}\right)^{th}$ percentiles of the Normal function obtained in this manner are precisely the CMOS points, which are to be used in the corresponding classification strategy. Using these, the method has been rigorously tested with different possibilities of k-OS and for various values of n, and the test results are given in Table 4.6.

To clarify the table, consider the row given by Trial No. 4 in which the 8-OS were invoked for the classification. In this case, we know that the possible symmetric

No.	Order(n)	Moments	CMOS	CMOS/ Dual CMOS
1	Two	$\left(\frac{2}{3}, \frac{1}{3}\right)$	91.865	CMOS
2	Four	$\left(\frac{4}{5}, \frac{1}{5}\right)$	91.865	CMOS
3	Six	$\left(\frac{6}{7}, \frac{1}{7}\right)$	91.865	CMOS
4	Eight	$\left(\frac{8}{9}, \frac{1}{9}\right)$	91.865	CMOS
5	Ten	$\left(\frac{10}{11}, \frac{1}{11}\right)$	91.865	Dual CMOS
6	Ten	$\left(\frac{9}{11}, \frac{2}{11}\right)$	91.865	CMOS
7	Twelve	$\left(\frac{12}{13}, \frac{1}{13}\right)$	91.865	Dual CMOS
8	Twelve	$\left(\frac{10}{13}, \frac{3}{13}\right)$	91.865	CMOS

Table 4.6: Results of the classification obtained by using the symmetric pairs of the k-OS for different values of n.

OS pairs can be $\langle 1, 8\rangle$, $\langle 2, 7\rangle$, $\langle 3, 6\rangle$ and $\langle 4, 5\rangle$ respectively. In every single case, the accuracy attained the Bayes' bound, as indicated by the results in the table.

Now, consider the results presented in the row denoted by Trial No. 5. In this case, the testing attained the Bayes' accuracy for the symmetric OS pairs $\langle 2, 9\rangle$, $\langle 3, 8\rangle$, $\langle 4, 7\rangle$ and $\langle 5, 6\rangle$ respectively. However, for the specific 10-OS, when the moments used were $\langle 1, 10\rangle$, the Dual CMOS has to be invoked. As in the Uniform and Doubly-Exponential distributions, if the chosen moments are in the near proximity of the Bayesian classifier, they do not possess sufficient information and capability to classify a testing point inasmuch as both these moments would be almost equidistant from the testing point. However, unlike the Uniform and Doubly-Exponential distributions, since the Gaussian pdf is not integrable, it is not possible to derive a closed form expression for this condition. The tabulated cases, however, demonstrate this phenomenon.

4.4 The Beta Distribution

The Beta distribution is a family of continuous probability distributions defined in $(0, 1)$ parameterized by two shape parameters α and β. The distribution can take different shapes based on the specific values of the parameters. If the parameters are identical, the distribution is symmetric with respect to $\frac{1}{2}$. Further, if $\alpha = \beta = 1$, $B(1, 1)$ becomes $U[0, 1]$. The pdf of the Beta distribution is:

$$f(x; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\ \Gamma(\beta)}\ x^{\alpha-1}(1 - x)^{\beta-1}. \tag{4.26}$$

The mean and the variance of the distribution are $\frac{\alpha}{\alpha+\beta}$ and $\frac{\alpha\beta}{(\alpha+\beta)^2\ (\alpha+\beta+1)}$ respectively.

The Beta distribution can take different shapes based on the values of the shape parameters, and hence any systematic analysis will have to be performed on a case-by-case basis. Some of the cases are given in Table 4.7 and are plotted in Figure 4.3.

No	α, β	Distribution
1	$\alpha = 1,\ \beta = 1$	Uniform Distribution
2	$\alpha < 1,\ \beta < 1$	U-shaped
3	$\alpha = 1,\ \beta = 2$	Straight line
4	$\alpha = \beta$	Symmetric about $\frac{1}{2}$
5	$\alpha = \frac{1}{2},\ \beta = \frac{1}{2}$	Arcsine Distribution
6	$\alpha > 1,\ \beta > 1$	Unimodal Distribution
7	$\alpha = 1,\ \beta > 1$	Strictly Convex
8	$\alpha = 1,\ 1 < \beta < 2$	Strictly Concave

Table 4.7: Different forms of the Beta Distribution for the various values of its parameters α and β.

For this study, we mainly consider three cases:

- $\alpha = 1,\ \beta = 1$: Uniform Distribution.

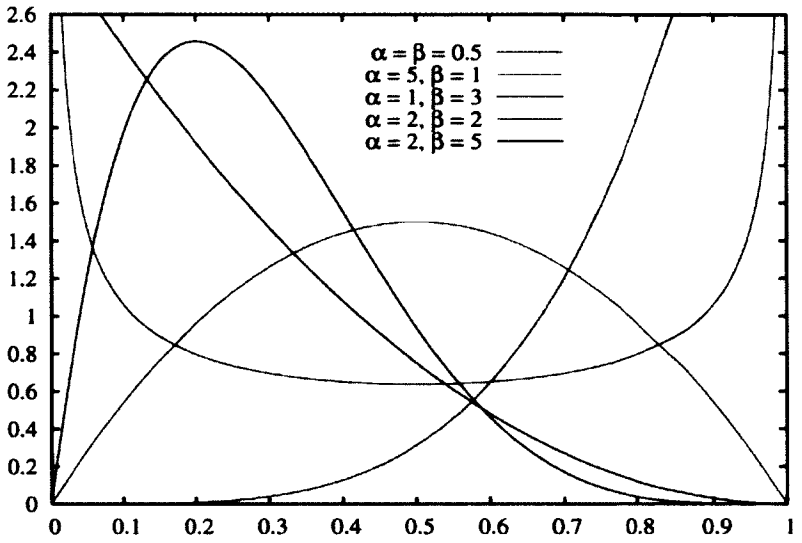


Figure 4.3: Beta Distribution (taken from www.wikipedia.org.).

- $\alpha = \beta$: Symmetric about $\frac{1}{2}$. In this case, in this present work, we merely deal with the scenario when $\alpha = \beta = 2$. The results for the more general case (i.e., $\alpha = \beta > 2$) will be briefly alluded to, but is left as an avenue for future work.

- $\alpha > 1$, $\beta > 1$: Unimodal Distribution.

Earlier, in [51], where we had initially introduced the concept of CMOS-based PR, we had analyzed the 2-OS and k-OS CMOS schemes for the Uniform distribution, and had provided the corresponding theoretical analysis and the experimental results. These were briefly catalogued in Section 3.3, from which we can see that for the 2-class problem in which the two class conditional distributions are Uniform and identical, CMOS can, indeed, attain the optimal Bayes' bound. To avoid repetition, the analysis for the Beta distribution, B(1,1) (which reduces to the analysis for Uniform U[0,1]) is omitted here, closing the first of the above three cases.

We now proceed to consider the Beta distribution in which $\alpha = \beta$.

4.4.1 Theoretical Analysis: Beta Distribution ($\alpha = \beta$) - 2-OS

Consider two classes ω_1 and ω_2 where the class ω_2 is displaced by a quantity θ, and the values of the shape parameters are identical. As in the previous cases, we consider the scenario when $\alpha_1 = \alpha_2 = \alpha$, $\beta_1 = \beta_2 = \beta$, and for the sake of simplicity[2], $\alpha = \beta = 2$. With these settings, the respective distributions become: $f(x, 2, 2) = 6x(1-x)$ and $f(x-\theta, 2, 2) = 6(x-\theta)(1-x+\theta)$.

We first derive the moments of the 2-OS, which are the points of interest for the CMOS, for the Beta distribution. By virtue of Eq. (3.3) and (3.4), the expected values of the first moments of the two OS can be obtained by determining the points where the cumulative distribution functions attain the values of $\frac{1}{3}$ and $\frac{2}{3}$ respectively. As the distribution takes different forms based on the values of the shape parameters, we have to solve each case separately, which we shall embark on. Let o_1 be the point for the percentile $\frac{2}{3}$ of the first distribution, and o_2 be the point for the percentile $\frac{1}{3}$ of the second distribution. Then:

$$\int_0^{o_1} 6x(1-x)dx = \frac{2}{3} \implies -6o_1^3 + 9o_1^2 - 2 = 0. \tag{4.27}$$

By a similar argument, the CMOS point for the $\frac{1}{3}$ percentile of the second distribution (if we don't take the displacement, θ, into consideration) leads to the equation:

$$-6o_2^3 + 9o_2^2 - 1 = 0. \tag{4.28}$$

We shall now prove that in this symmetric case, CMOS, indeed, attains the optimal Bayes' bound.

Theorem 7. *For the 2-class problem in which the two class conditional distributions are Beta(α, β) ($\alpha = \beta$) and identical, CMOS, the classification using two OS, attains an accuracy that is exactly identical to the optimal Bayes' bound.*

Proof. Without loss of generality, for the case when $\alpha = \beta = 2$ let the distributions of ω_1 and ω_2 be $B(x, 2, 2)$ and $B(x-\theta, 2, 2)$, where θ is the displacement for the second

[2]Because of the symmetry of the analysis shown below, we believe that the results are true for any positive integer value $\alpha = \beta = C$. However, the algebraic equations will be quite cumbersome for values of $C > 2$. Thus, in the following, we shall use the specific values of $\alpha = \beta = 2$.

distribution. Then, to get the Bayes' classifier, we argue that:

$$p(x|\omega_1)P(\omega_1) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} p(x|\omega_2)P(\omega_2) \implies 6x(1-x) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} 6(x-\theta)(1-(x-\theta))$$
$$\implies x \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \frac{\theta+1}{2}. \tag{4.29}$$

We now consider the classification with respect to the expected values of the moments of the 2-OS, o_1 and o_2. In order to prove our claim, we need to show that

$$x \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \frac{\theta+1}{2} \implies D(x,o_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(x,o_2). \tag{4.30}$$

If we examine Eqs. (4.27) and (4.28), we can see that Eq. (4.28) can be obtained by substituting $1-o_2$ for o_1 in Eq. (4.27) since:

$$-6(1-o_2)^3+9(1-o_2)^2-2=0 \implies -6o_2^3+9o_2^2-1=0. \tag{4.31}$$

Consequently, it is obvious that $o_2=\theta+o_1-1$, implying that the RHS of the claim given by Eq. (4.30) leads to the following:

$$D(x,o_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(x,o_2) \implies D(x,o_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(x,\theta+1-o_1)$$
$$\implies x-o_1 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \theta+1-o_1-x$$
$$\implies x \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \frac{\theta+1}{2}. \tag{4.32}$$

The result follows by observing that Eqs. (4.29) and (4.30) are identical comparisons. Hence the theorem. □

4.4.2 Data Generation: Beta Generation

As in the case of the Gamma distribution, the data is generated using the built-in function available in MatLab, namely *betarnd*(α,β,r), where α and β are the shape parameters, and where the function returns a square matrix with the dimension r. To be specific, *betarnd*$(2,2,10)$ will generate 100 values that follow the Beta distribution with 2 being the value for the shape parameters. For our experiments, we generated 1,000 points for each of the distributions, where the second distribution was displaced by a constant, θ.

4.4.3 Experimental Results: Beta Distribution ($\alpha = \beta$) - 2-OS

The CMOS has been rigorously tested for various Beta distributions with 2-OS with $\alpha = \beta = 2$. In the interest of brevity, a few typical results are given below. For each of the experiments, we generated 1,000 points for the classes ω_1 and ω_2 characterized by $B(x, 2, 2)$ and $B(x-\theta, 2, 2)$ respectively. We then invoked a classification procedure by utilizing the Bayesian and the CMOS strategies. In every case, CMOS was compared with the Bayesian classifier for different values of θ, as tabulated in Table 4.8. The results were obtained by executing each algorithm 50 times using a 10-fold cross-validation scheme.

θ	0.95	0.90	0.85	0.80	0.75	0.70	0.65	0.60
Bayesian	99.845	99.45	98.185	96.94	94.95	92.86	90.31	88.075
CMOS	99.845	99.45	98.185	96.94	94.95	92.86	90.31	88.075

Table 4.8: A comparison of the accuracy of the Bayesian and the 2-OS CMOS classifier for the Beta distribution $B(2, 2)$ for different values of θ.

The results given in this table justify the claim of Theorem 7. We conjecture that this claim is true for any $\alpha = \beta = t$, but it is presently considered as an unsolved problem.

4.4.4 Theoretical Analysis: Beta Distribution ($\alpha = \beta$)- k-OS

We have seen from Theorem 7 that the moments of the 2-OS are sufficient for the classification to attain a Bayes' bound. We shall now examine the scenario where the k-OS CMOS is invoked, and thus determine the strength of the proposed method.

Theorem 8. *For the 2-class problem in which the two class conditional distributions are Beta and identical as $B(x, \alpha, \beta)$ and $B(x - \theta, \alpha, \beta)$ where $\alpha = \beta = 2$, optimal Bayesian classification can be achieved by using symmetric pairs of the n-OS, i.e., the $n - k$ OS for ω_1 (represented by o_1) and the k OS for ω_2 (represented by o_2) if and*

only if $o_1 < o_2$. *If* $o_1 > o_2$, *optimal Bayesian classification can be achieved by using the* Dual *symmetric pairs of the n-OS, i.e., the k OS for* ω_1 *and the* $n-k$ *OS for* ω_2.

Proof. Our claim is that we can choose any of the symmetric pairs of the n-OS, i.e., the $n-k$ OS for ω_1 and the k OS for ω_2 to obtain an optimal classification. Let o_1 be the point for the percentile $\frac{n+1-k}{n+1}$ of the first distribution, and o_2 be the point for the percentile $\frac{k}{n+1}$ of the second distribution. Then:

$$\int_0^{o_1} 6x(1-x)dx = \frac{n+1-k}{n+1} \implies -2o_1^3 + 3o_1^2 - \frac{n+1-k}{n+1} = 0. \quad (4.33)$$

By a similar argument, if we ignore the displacement θ, the CMOS point for the $\frac{k}{n+1}$ percentile of the second distribution leads to the equation:

$$-2o_2^3 + 3o_2^2 - \frac{k}{n+1} = 0. \quad (4.34)$$

We have already shown in Eq. (4.29) that the Bayes' classifier is equivalent to the inequality $x \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \frac{\theta+1}{2}$. Thus, in order to prove our claim, we need to show that the same classification criterion results for any symmetric pairs of the n-OS. Thus, our claim is:

$$x \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \frac{\theta+1}{2} \implies D(x,o_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(x,o_2). \quad (4.35)$$

As in the case of 2-OS, if we substitute $o_1 = 1 - o_2$ in Eq. (4.33), the equation reduces to $-2o_2^3 + 3o_2^2 - \frac{k}{n+1} = 0$, proving the fact that for the distributions ω_1 and ω_2, the $\langle \frac{n+1-k}{n+1}, \frac{k}{n+1} \rangle$ CMOS positions (represented by o_1 and o_2 respectively) have the relation $o_2 = 1 - o_1$. Thus, the the RHS of the claim given by Eq. (4.35) simplifies to:

$$\begin{aligned} D(x,o_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(x,o_2) &\implies D(x,o_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(x,\theta+1-o_1) \\ &\implies x - o_1 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \theta + 1 - o_1 - x \\ &\implies x \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \frac{\theta+1}{2}, \end{aligned} \quad (4.36)$$

which is the condition sought for.

The arguments for the cases when the Dual condition has to be invoked follow in an identical manner and are omitted. The theorem follows. □

4.4.5 Experimental Results: Beta Distribution ($\alpha = \beta$) - k-OS

The CMOS method has also been tested for the Beta distribution for other k OS when $\alpha = \beta = 2$. In the interest of brevity, we merely cite one example where the distributions for ω_1 and ω_2 were characterized by $\beta(x, 2, 2)$ and $\beta(x - \theta, 2, 2)$ respectively. For each of the experiments, we generated 1,000 points for each class, and the testing samples were classified based on the selected *symmetric* pairs for values k and $n - k$ respectively. The results are displayed in Table 4.9.

Trial No.	Order(n)	Moments	OS_1	OS_2	CMOS	CMOS/ Dual CMOS
1	Two	$\langle \frac{2}{3}, \frac{1}{3} \rangle$	0.61304	$\theta + 0.38696$	87.3	CMOS
2	Four	$\langle \frac{4}{5}, \frac{1}{5} \rangle$	0.71286	$\theta + 0.28714$	87.3	CMOS
3	Four	$\langle \frac{3}{5}, \frac{2}{5} \rangle$	0.56707	$\theta + 0.43293$	87.3	CMOS
4	Six	$\langle \frac{6}{7}, \frac{1}{7} \rangle$	0.7621	$\theta + 0.23790$	87.3	CMOS
5	Six	$\langle \frac{5}{7}, \frac{2}{7} \rangle$	0.6471	$\theta + 0.3529$	87.3	CMOS
6	Six	$\langle \frac{4}{7}, \frac{3}{7} \rangle$	0.54776	$\theta + 0.45224$	87.3	CMOS
7	Eight	$\langle \frac{8}{9}, \frac{1}{9} \rangle$	0.79269	$\theta + 0.20731$	87.3	Dual CMOS
8	Eight	$\langle \frac{7}{9}, \frac{2}{9} \rangle$	0.69508	$\theta + 0.30492$	87.3	CMOS
9	Eight	$\langle \frac{5}{9}, \frac{4}{9} \rangle$	0.53711	$\theta + 0.46289$	87.3	CMOS

Table 4.9: A comparison of the accuracy of the Bayesian and the k-OS CMOS classifier for the Beta Distribution by using the symmetric pairs of the OS for different values of n. The value of θ was set to be 0.58. Note that in every case, the accuracy attained the Bayes' value whenever the conditions stated in Theorem 8 were satisfied.

To clarify the table, consider the cases in which the 8-OS were invoked for the classification. For 8-OS, the possible symmetric OS pairs could be $\langle 1, 8 \rangle$, $\langle 2, 7 \rangle$, and $\langle 4, 5 \rangle$ respectively. Wherever the condition $o_1 < o_2$ is satisfied, the CMOS attained the optimal Bayes' bound, as indicated by the results in the table (denoted by Trial Nos. 8 and 9). Now, consider the results presented in the row denoted by Trial No. 7. In this case where the CMOS positions were 0.79269 and $\theta + 0.20731$, the inequality of the condition imposed in Theorem 8 simplified to $0.79269 < 0.78731$, which is not valid. Observe that if $o_1 > o_2$, the symmetric pairs should be reversed to obtain the optimal Bayes' bound.

This concludes our study on the symmetric Beta distribution.

4.5 Conclusions

The pioneering work on using OS for classification (Classification by Moments of Order Statistics (CMOS)) was presented in Chapter 3 for the generic classifier and for the uni-dimensional Uniform distribution, by comparing the testing sample to a few samples distant from the mean. In that chapter, we showed that if these points are obtained by an OS criteria, they can attain the optimal Bayes' bound. In this chapter, we demonstrated that these results can be extended for a few *symmetric* distributions within the exponential family, namely the Doubly-Exponential, Gaussian and a form of the Beta distribution. We showed that the new scheme again has an accuracy that attains the Bayes' bound for symmetric distributions, and this theoretical assertion has been demonstrated by a rigorous experimental verification.

Chapter 5

Optimal "Anti-Bayesian" OS-based Parametric PR for Asymmetric Distributions in the Exponential Family

5.1 Introduction

The pioneering work on using OS for classification was presented in Chapter 3 for the Uniform distribution, where it was shown that optimal PR can be achieved in a counter-intuitive manner, diametrically opposed to the Bayesian paradigm, i.e., by comparing the testing sample to a few samples distant from the mean. In Chapter 4, we also showed that the results could be extended for a few *symmetric* distributions within the exponential family. In this chapter[1], we attempt to extend these results significantly by considering a spectrum of asymmetric distributions within the

[1] A preliminary version of some of the results of this chapter can be found in the *Proceedings of ICMLPR'12, the 2012 International Conference on Machine Learning and Pattern Recognition* held in Penang, in December 2012 [48]. More detailed descriptions of these results have been published in the journal *Pattern Recognition* [38].

exponential family, for some of which even the closed form expressions of the cumulative distribution functions are not available. Our new results include the Rayleigh, Gamma and certain Beta distributions.

5.1.1 Contributions of this Chapter

The novel contributions of this chapter are the following:

- We extend the "anti-Bayesian" generic paradigm of Section 3.2 for the classification of uni-dimensional asymmetric distributions within the exponential family, and show that the strategy requires a very few number of non-central training patterns to attain near-optimal accuracy;
- To justify these claims, we submit a formal analysis for the Raleigh, Gamma and some forms of Beta distributions by invoking the 2-OS CMOS and the k-OS CMOS. Further, for the k-OS CMOS, the condition at which the Dual CMOS has to be invoked is also thoroughly investigated.
- The claims are experimentally verified by a rigorous testing for the above-mentioned uni-dimensional distributions, and the results confirm the assertions unequivocally.

5.2 The Rayleigh Distribution

The Rayleigh distribution is a continuous probability distribution which is often observed when the overall magnitude of a vector is related to its directional components. Specifically, it can be used when two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components of a complex variable may have absolute values that are Rayleigh distributed. The pdf of the Rayleigh distribution, with parameter $\sigma > 0$ is:

$$\varphi(x, \sigma) = \frac{x}{\sigma^2} e^{-x^2/2\sigma^2}, \quad x \geq 0,$$

and the cumulative distribution function is:

$$\Phi(x) = 1 - e^{-x^2/2\sigma^2}, \quad x \geq 0.$$

The mean, the variance and the median of the Rayleigh distribution are $\mu(x) = \sigma\sqrt{\frac{\pi}{2}}$, $Var(x) = \frac{4-\pi}{2}\sigma^2$ and $Median(x) = \sigma\sqrt{ln(4)}$, respectively.

A plot of the pdf of the Rayleigh distribution for different values for the parameter σ is shown in Figure 5.1.

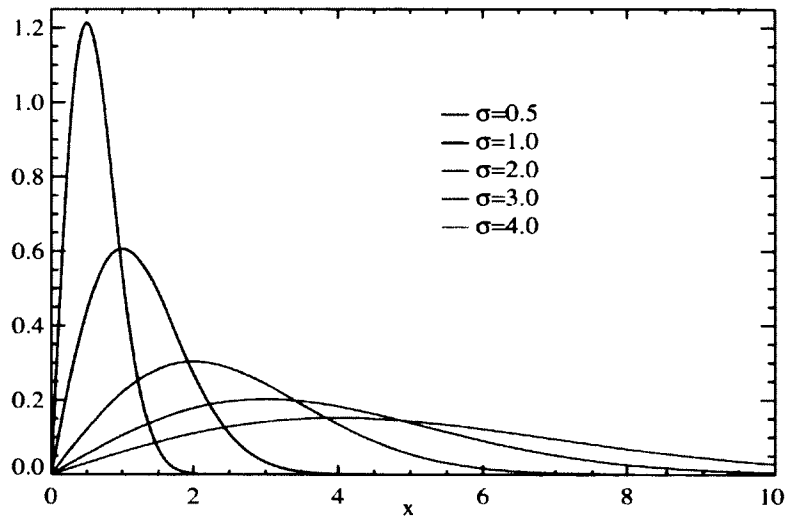


Figure 5.1: Rayleigh Distribution (taken from www.wikipedia.org.).

5.2.1 Theoretical Analysis: Rayleigh Distribution - 2-OS

The typical PR problem involving the Rayleigh distribution would consider two classes ω_1 and ω_2 where the class ω_2 is displaced by a quantity θ, and the values of σ are σ_1 and σ_2 respectively. As in the previous cases, we consider the scenario when $\sigma_1 = \sigma_2 = \sigma$. Consider the distributions: $f(x, \sigma) = \frac{x}{\sigma^2} e^{\frac{-x^2}{2\sigma^2}}$ and $f(x - \theta, \sigma) = \frac{x-\theta}{\sigma^2} e^{\frac{-(x-\theta)^2}{2\sigma^2}}$. In order to do the classification based on CMOS, we shall first derive the moments of the 2-OS for the Rayleigh distribution. By virtue of Eq. (3.3) and (3.4), the expected values of the first moments of the two OS can be obtained by determining the points where

the cumulative distribution function attains the values of $\frac{1}{3}$ and $\frac{2}{3}$ respectively. Let o_1 be the point for the percentile $\frac{2}{3}$ of the first distribution, and o_2 be the point for the percentile $\frac{1}{3}$ of the second distribution. Then:

$$\begin{aligned} \int_0^{o_1} \frac{x}{\sigma^2} e^{-x^2/2\sigma^2} dx = \frac{2}{3} &\implies 1 - e^{\frac{o_1^2}{2\sigma^2}} = \frac{2}{3} \\ &\implies o_1 = \sigma\sqrt{2\ ln(3)}. \end{aligned} \tag{5.1}$$

Using a similar argument, o_2 can be evaluated as:

$$o_2 = \theta + \sigma\sqrt{2ln\left(\frac{3}{2}\right)}. \tag{5.2}$$

We now derive the result concerning the efficiency of the CMOS when compared to the Bayesian classifier.

Theorem 9. *For the 2-class problem in which the two class conditional distributions are Rayleigh and identical, the accuracy obtained by CMOS, the classification using two OS, deviates from the optimal Bayes' bound as the solution of the transcendental equality* $ln\left(\frac{x}{x-\theta}\right) = \frac{-\theta^2+2\theta x}{2\sigma^2}$ *deviates from* $\frac{\theta}{2} + \frac{\sigma}{\sqrt{2}}\left(\sqrt{ln(3)} + \sqrt{ln\left(\frac{3}{2}\right)}\right)$.

Proof. Without loss of generality, let the distributions of ω_1 and ω_2 be $R(x, \sigma)$ and $R(x-\theta, \sigma)$, where σ is the identical scale parameter. Then, to get the Bayes' classifier, we argue that:

$$\begin{aligned} p(x|\omega_1)P(\omega_1) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} p(x|\omega_2)P(\omega_2) &\implies \frac{x}{\sigma^2} e^{\frac{-x^2}{2\sigma^2}} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{x-\theta}{\sigma^2} e^{\frac{-(x-\theta)^2}{2\sigma^2}} \\ &\implies \frac{x}{x-\theta} \underset{\omega_2}{\overset{\omega_1}{\lesseqgtr}} e^{\frac{-(x-\theta)^2}{2\sigma^2} + \frac{x^2}{2\sigma^2}} \\ &\implies ln\left(\frac{x}{x-\theta}\right) \underset{\omega_2}{\overset{\omega_1}{\lesseqgtr}} \frac{-\theta^2 + 2\theta x}{2\sigma^2}. \end{aligned} \tag{5.3}$$

The discriminant is then the solution to the transcendental equation:

$$ln\left(\frac{x}{x-\theta}\right) = \frac{-\theta^2 + 2\theta x}{2\sigma^2}. \tag{5.4}$$

We now consider the classification with respect to the expected values of the moments of the 2-OS, o_1 and o_2, where as per Eq. (5.1) and (5.2), $o_1 = \sigma\sqrt{2\ ln(3)}$

and $o_2 = \theta + \sigma\sqrt{2ln\left(\frac{3}{2}\right)}$. The discriminant enforced by the 2-OS classifier satisfies:

$$D(x, o_1) = D(x, o_2). \tag{5.5}$$

The condition imposed by Eq. (5.5) leads to the following:

$$\begin{aligned} D(x,o_1) = D(x,o_2) \quad &\Longrightarrow \quad D\left(x,\ \sigma\sqrt{2\ ln(3)}\right) = D\left(x,\ \theta + \sigma\sqrt{2ln\left(\frac{3}{2}\right)}\right) \\ &\Longrightarrow \quad 2x \ = \ \theta + \sigma\sqrt{2ln(3)} + \sigma\sqrt{2ln\left(\frac{3}{2}\right)} \\ &\Longrightarrow \quad x \ = \ \frac{\theta}{2} + \frac{\sigma}{\sqrt{2}}\left(\sqrt{ln(3)} + \sqrt{ln\left(\frac{3}{2}\right)}\right). \end{aligned} \tag{5.6}$$

The difference in the errors of the two classifiers is clearly related to differences in the corresponding discriminant functions specified by Eq. (5.4) and (5.6). The result follows. □

Remark:

Another way of comparing the approaches is by obtaining the error difference created by the CMOS classifier when compared to the Bayesian classifier. In Figure 5.2, the small area marked as "Error Difference" is the difference between the probability of error formed by the CMOS classifier when compared to the Bayesian counterpart, and we can evaluate this area by using the corresponding definite integrals. As Eq. (5.4) is transcendental in nature, the only way to find the Bayesian classifier is to resort to a numerical strategy, for example, by using a Taylor series expansion. The area depicting the differences in classification accuracy (in percentage) is reported in Table 5.1. Since the accuracy of the Taylor's expansion depends on the point around which the expansion is done, in Table 5.1, we have also recorded this point, i.e., the one around which the Taylor's expansion has been invoked for each specific scenario. From this table, we can see that the CMOS classifier is bounded by an error difference of less than 0.15%, which is truly, negligible.

Theorem 10. *For the 2-class problem in which the two class conditional distributions are Rayleigh and identical, CMOS, the accuracy obtained by CMOS, the*

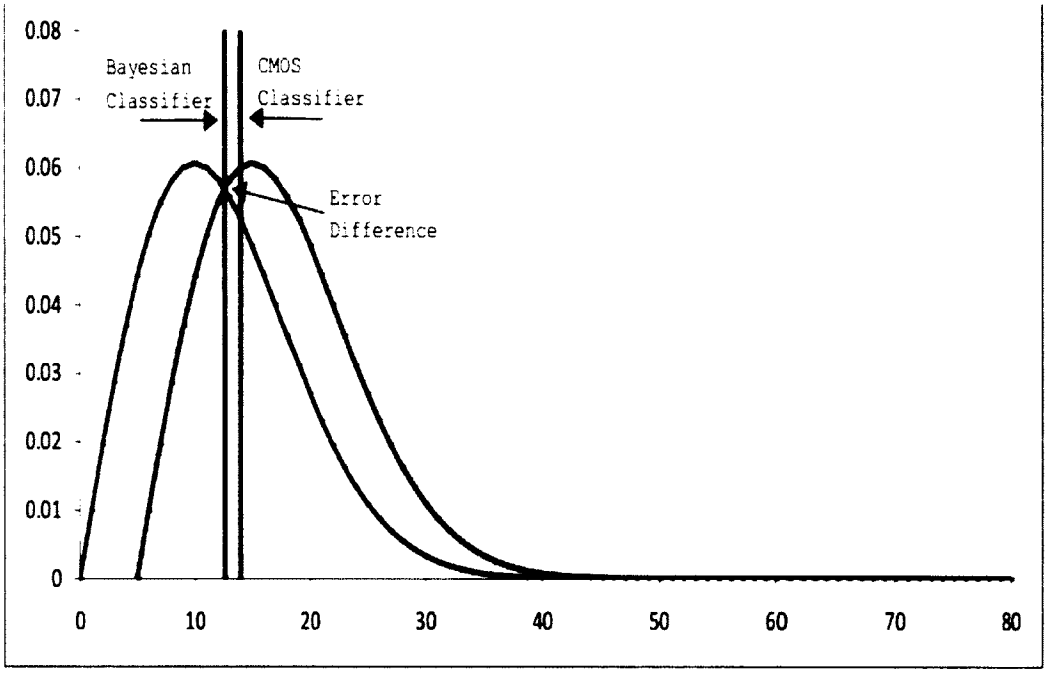


Figure 5.2: The differences of the error probability quantified by the differences between the areas under the curves of the resulting errors.

θ	1	1.5	2	2.5	3
a	3	4	5	5.3	6.5
Max. Bounded Error(in %)	0.15	0.06	0.05	0.001	0

Table 5.1: The maximum bounded error by the CMOS classifier when compared to the Bayesian classifier, for different values of θ of the Rayleigh Distribution. In each case, $\sigma = 2$, and the Taylor's expansion was invoked around the point a.

classification using two OS, deviates from the classifier which discriminates based on the distance from the corresponding **medians** *as* $\frac{\theta}{2} + \sigma\sqrt{ln(4)}$ *deviates from* $\frac{\theta}{2} + \frac{\sigma}{\sqrt{2}}\left(\sqrt{ln(3)} + \sqrt{ln\left(\frac{3}{2}\right)}\right)$.

Proof. As the curve of the Rayleigh distribution is not symmetric, for the present

analysis, we shall consider the scenario that the classification is done based on the median, which is the most central point of the distribution, other than the mean. In order to prove the theorem, we shall first show that when the class conditional distributions are Rayleigh and identical, the accuracy of the corresponding near-optimal discriminant obtained by a comparison to the corresponding medians is almost equal to the accuracy of the CMOS. Again, as in the case of Theorem 9, as the equations are transcendental, we can consider the classification based on the medians of the given distributions, namely $\nu_1 = \sigma\sqrt{ln(4)}$ and $\nu_2 = \theta + \sigma\sqrt{ln(4)}$, respectively. The classification will be based on the distances that the testing point has with respect to the respective medians. Thus,

$$\begin{aligned} D(x,\nu_1) < D(x,\nu_2) &\implies x - \sigma\sqrt{ln(4)} < \theta + \sigma\sqrt{ln(4)} - x \\ &\implies 2x < \theta + 2\sigma\sqrt{ln(4)} \\ &\implies x < \frac{\theta}{2} + \sigma\sqrt{ln(4)}. \end{aligned} \tag{5.7}$$

The discriminant function with regard to the medians of the distributions is: $x = \frac{\theta}{2} + \sigma\sqrt{ln(4)}$.

We now consider the classification with respect to the expected values of the moments of the 2-OS, o_1 and o_2, where as per Eq. (5.1) and (5.2), $o_1 = \sigma\sqrt{2\ ln(3)}$ and $o_2 = \theta + \sigma\sqrt{2ln\left(\frac{3}{2}\right)}$. The discriminant enforced by 2-OS CMOS is:

$$D(x,o_1) = D(x,o_2). \tag{5.8}$$

This equation simplifies to:

$$\begin{aligned} D(x,o_1) = D(x,o_2) &\implies D\left(x,\ \sigma\sqrt{2\ ln(3)}\right) = D\left(x,\ \theta + \sigma\sqrt{2ln\left(\frac{3}{2}\right)}\right) \\ &\implies 2x = \theta + \sigma\sqrt{2ln(3)} + \sigma\sqrt{2ln\left(\frac{3}{2}\right)} \\ &\implies x = \frac{\theta}{2} + \frac{\sigma}{\sqrt{2}}\left(\sqrt{ln(3)} + \sqrt{ln\left(\frac{3}{2}\right)}\right). \end{aligned} \tag{5.9}$$

The difference in the errors of the two classifiers is clearly related to differences in the corresponding discriminant functions specified by Eq. (5.7) and (5.9). Hence the theorem. □

Remark: As in Theorem 9, we can show that Eqs. (5.7) and (5.9) are almost identical by obtaining the error difference created by the CMOS classifier when compared to the classifier based on the corresponding medians for different values of θ. The area depicting the differences in classification accuracy (in percentage) is reported[2] in Table 5.2. From this table, we can see that the CMOS classifier is bounded by an error difference of less than 0.42%, which is again, negligible.

θ	1	1.5	2	2.5	3	3.5
Maximum Bounded Error(in %)	0.40	0.34	0.20	0.14	0	0

Table 5.2: Maximum bounded error by the CMOS classifier when compared to the classifier obtained with regard to the median, for different values of θ of the Rayleigh Distribution. In each case, $\sigma = 1$.

Corollary 1. *By virtue of the almost-identical nature of the two expressions for the Rayleigh distribution, the classification using the proximity to the median is almost indistinguishable from that of the Bayesian classifier.*

Proof. This result is an indirect implied consequence of Theorems 9 and 10. □

5.2.2 Data Generation: Rayleigh Generation

To experimentally verify our results, we made use of a Uniform $(0, 1)$ random variable generator to generate data values that follow a Rayleigh distribution. The expression $x = \sigma\sqrt{-2\ ln(1-u)}$, where σ is the parameter and u is a random variate from the Uniform distribution $U[0, 1]$, generates Rayleigh distributed values [7].

5.2.3 Experimental Results: Rayleigh Distribution - 2-OS

The CMOS classifier was rigorously tested for a number of experiments with various Rayleigh distributions having the identical parameter σ. In every case, the 2-OS

[2]Since the expressions are directly solvable, we do not need to resort to a Taylor's expansion in this case.

CMOS gave almost the same classification as that of the Bayesian classifier. The method was executed 50 times with the 10-fold cross validation scheme. The test results are tabulated in Table 5.3 and justify Theorem 9.

The results presented justify the claims of Theorems 9 and 10.

θ	3	2.5	2	1.5	1
Bayesian	99.1	97.35	94.45	87.75	78.80
CMOS	99.1	97.35	94.40	87.70	78.65

Table 5.3: A comparison of the accuracy of the Bayesian and the 2-OS CMOS classifier for the Rayleigh Distribution.

5.2.4 Theoretical Analysis: Rayleigh Distribution - k-OS

We have seen from Theorem 9 that for the Rayleigh distribution, the moments of the 2-OS are sufficient for a near-optimal classification. As in the case of the other distributions, we shall now consider the scenario when we utilize other k-OS. The formal result pertaining to this is given in Theorem 11.

Theorem 11. *For the 2-class problem in which the two class conditional distributions are Rayleigh and identical, a near-optimal Bayesian classification can be achieved by using symmetric pairs of the n-OS, i.e., the $n-k$ OS for ω_1 and the k OS for ω_2 if and only if $\sqrt{ln\left(\frac{n+1}{k}\right)} - \sqrt{ln\left(\frac{n+1}{n+1-k}\right)} < \frac{\theta}{\sigma\sqrt{2}}$. If this condition is violated, the CMOS classifier uses the* Dual *condition, i.e., the k OS for ω_1 and the $n-k$ OS for ω_2. In both these cases, the classification obtained by CMOS deviates from the optimal Bayes' bound as the solution of the transcendental equality $ln\left(\frac{x}{x-\theta}\right) = \frac{-\theta^2+2\theta x}{2\sigma^2}$ deviates from $\frac{\theta}{2} + \frac{\sigma}{\sqrt{2}}\left[\sqrt{ln\left(\frac{n+1}{k}\right)} + \sqrt{ln\left(\frac{n+1}{n+1-k}\right)}\right]$.*

Proof. First of all, we invoke the result of Corollary 1 that classification based on the proximity to the median is almost equivalent to the Bayesian classification. We shall now show the result for the k-OS, that the classification is almost identical to the classification based on the medians. Before proceeding further, we have to show that

the expected values of the first moment of the k-order OS for the Rayleigh distribution have the form $E[x_{k,n}] = \sigma\sqrt{2\ ln\left(\frac{n+1}{k}\right)}$. Let o_1 be the point for the percentile $\frac{n+1-k}{n+1}$ (the $(n-k)^{th}$-OS) of the first distribution, and o_2 be the point for the percentile $\frac{k}{n+1}$ (the k-OS) of the second distribution. Then:

$$\begin{aligned}\int_0^{o_1} \frac{x}{\sigma^2}\, e^{-x^2/2\sigma^2} dx \;=\; \frac{n+1-k}{n+1} \quad &\Longrightarrow \quad 1 - e^{\frac{o_1^2}{2\sigma^2}} \;=\; \frac{n+1-k}{n+1} \\ &\Longrightarrow \quad o_1 = \sigma\sqrt{2\ ln\left(\frac{n+1}{k}\right)}. \end{aligned} \tag{5.10}$$

Using a similar argument, o_2 can be evaluated as:

$$o_2 = \theta + \sigma\sqrt{2\ ln\left(\frac{n+1}{n+1-k}\right)}. \tag{5.11}$$

Our present claim is based on the classification in which we can choose any of the symmetric pairs of the n-OS, i.e., the $n-k$ OS for ω_1 and the k OS for ω_2, where these quantities are $\sigma\sqrt{2\ ln\left(\frac{n+1}{k}\right)}$ and $\theta + \sigma\sqrt{2\ ln\left(\frac{n+1}{n+1-k}\right)}$ respectively.

It is obvious that an OS value can correctly classify a testing point only when their positions have the correct ordering, i.e., $o_1 < o_2$. We can resolve this condition by solving this inequality as:

$$\begin{aligned} & \sigma\sqrt{\left(2\ ln\left(\frac{n+1}{k}\right)\right)} \;<\; \theta + \sigma\sqrt{\left(2\ ln\left(\frac{n+1}{n+1-k}\right)\right)} \\ \Longrightarrow \quad & \sigma\sqrt{2}\left[\sqrt{ln\left(\frac{n+1}{k}\right)} - \sqrt{ln\left(\frac{n+1}{n+1-k}\right)}\right] \;<\; \theta \\ \Longrightarrow \quad & \sqrt{ln\left(\frac{n+1}{k}\right)} - \sqrt{ln\left(\frac{n+1}{n+1-k}\right)} \;<\; \frac{\theta}{\sigma\sqrt{2}}. \end{aligned} \tag{5.12}$$

We have seen from Eq. (5.4) that the Bayesian classifier is the solution to the transcendental equation:

$$ln\left(\frac{x}{x-\theta}\right) = \frac{-\theta^2 + 2\theta x}{2\sigma^2}. \tag{5.13}$$

The discriminant enforced by the k-OS CMOS classifier is $D(x, o_1) = D(x, o_2)$ which can further be simplified to:

$$D(x, o_1) = D(x, o_2) \tag{5.14}$$

$$\Longrightarrow \quad D\left(x, \sigma\sqrt{\left(2\ ln\left(\frac{n+1}{k}\right)\right)}\right) = D\left(x, \theta + \sigma\sqrt{\left(2\ ln\left(\frac{n+1}{n+1-k}\right)\right)}\right)$$

$$\Longrightarrow \quad x - \left(\sigma\sqrt{\left(2\ ln\left(\frac{n+1}{k}\right)\right)}\right) = \left(\theta + \sigma\sqrt{\left(2\ ln\left(\frac{n+1}{n+1-k}\right)\right)}\right) - x$$

$$\Longrightarrow \quad x = \frac{\theta}{2} + \frac{\sigma}{\sqrt{2}}\left[\sqrt{ln\left(\frac{n+1}{k}\right)} + \sqrt{ln\left(\frac{n+1}{n+1-k}\right)}\right]. \tag{5.15}$$

The difference in the errors of the two classifiers is clearly related to differences in the corresponding discriminant functions specified by Eq. (5.13) and (5.14). The case when the *Dual* condition has to be invoked follows in an analogous manner and is omitted in the interest of brevity. Hence the theorem. □

Remark: As in the case of the 2-OS, if we examine the error bounded by the CMOS classifier with regard to the classifier which discriminates based on the distance from the corresponding medians for different values of θ, k and n, we can see that the classifiers are almost identical. This is demonstrated by the results tabulated in Table 5.4.

θ	x : 4-**OS**, $k = 2$	x : 6-**OS**, $k = 3$	x : 8-**OS**, $k = 4$
1	0.05	0.03	0.01
1.5	0.08	0.04	0.03
2	0.08	0.05	0.02
2.5	0.04	0.02	0.01
3	0	0	0

Table 5.4: Maximum bounded error (in %) by the CMOS classifier when compared to the classifier with respect to the medians of the distributions, for different values of θ, k and n of the Rayleigh Distribution. In each case, $\sigma = 2$.

5.2.5 Experimental Results: Rayleigh Distribution - k-OS

The CMOS method has been rigorously tested with different possibilities of the k-OS and for various values of n, and the test results are given in Table 5.5. For the distribution under consideration, the Bayesian approach provides an accuracy of 82.15%, and from the table, it is obvious that some of the considered k-OSs attains the optimal accuracy and the rest of the cases attain near-optimal accuracy. Also, we can see that the Dual CMOS has to be invoked if the condition stated in Theorem 11 is not satisfied.

No.	Order(n)	Moments	OS_1	OS_2	CMOS	CMOS/ Dual CMOS
1	Two	$(\frac{2}{3}, \frac{1}{3})$	$\sigma\sqrt{(2\ ln\ (\frac{3}{1}))}$	$\theta + \sigma\sqrt{(2\ ln\ (\frac{3}{2}))}$	82.05	CMOS
2	Four	$\left(\frac{5-i}{5}, \frac{i}{5}\right), 1 \leq i \leq \frac{n}{2}$	$\sigma\sqrt{(2\ ln\ (\frac{5}{1}))}$	$\theta + \sigma\sqrt{(2\ ln\ (\frac{5}{4}))}$	81.8	CMOS
3	Four	$\left(\frac{5-i}{5}, \frac{i}{5}\right), 1 \leq i \leq \frac{n}{2}$	$\sigma\sqrt{(2\ ln\ (\frac{5}{2}))}$	$\theta + \sigma\sqrt{(2\ ln\ (\frac{5}{3}))}$	82.0	CMOS
4	Six	$\left(\frac{7-i}{7}, \frac{i}{7}\right) 1 \leq i \leq \frac{n}{2}$	$\sigma\sqrt{(2\ ln\ (\frac{7}{1}))}$	$\theta + \sigma\sqrt{(2\ ln\ (\frac{7}{6}))}$	81.6	Dual CMOS
5	Six	$\left(\frac{7-i}{7}, \frac{i}{7}\right) 1 \leq i \leq \frac{n}{2}$	$\sigma\sqrt{(2\ ln\ (\frac{7}{2}))}$	$\theta + \sigma\sqrt{(2\ ln\ (\frac{7}{5}))}$	82.10	CMOS
6	Six	$\left(\frac{7-i}{7}, \frac{i}{7}\right) 1 \leq i \leq \frac{n}{2}$	$\sigma\sqrt{(2\ ln\ (\frac{7}{3}))}$	$\theta + \sigma\sqrt{(2\ ln\ (\frac{7}{4}))}$	82.15	CMOS
7	Eight	$\left(\frac{9-i}{9}, \frac{i}{9}\right), 1 \leq i \leq \frac{n}{2}$	$\sigma\sqrt{(2\ ln\ (\frac{9}{1}))}$	$\theta + \sigma\sqrt{(2\ ln\ (\frac{9}{8}))}$	81.55	Dual CMOS
8	Eight	$\left(\frac{9-i}{9}, \frac{i}{9}\right), 1 \leq i \leq \frac{n}{2}$	$\sigma\sqrt{(2\ ln\ (\frac{9}{2}))}$	$\theta + \sigma\sqrt{(2\ ln\ (\frac{9}{7}))}$	82.05	CMOS
9	Eight	$\left(\frac{9-i}{9}, \frac{i}{9}\right), 1 \leq i \leq \frac{n}{2}$	$\sigma\sqrt{(2\ ln\ (\frac{9}{4}))}$	$\theta + \sigma\sqrt{(2\ ln\ (\frac{9}{5}))}$	82.15	CMOS

Table 5.5: A comparison of the accuracy of the Bayesian(i.e., 82.15%) and the k-OS CMOS classifier for the Rayleigh Distribution by using the symmetric pairs of the OS for different values of n. The value of σ and θ were set to be 2 and 1.5 respectively. Note that in every case, CMOS attained near-optimal accuracy whenever the conditions stated in Theorem 11 were satisfied.

To clarify the table, consider the cases in which the 6-OS were invoked for the classification. For 6-OS, the possible symmetric OS pairs could be $\langle 1,6\rangle$, $\langle 2,5\rangle$, and $\langle 3,4\rangle$ respectively. Observe that the expected values for the first moment of the k-OS has the form $E[\mathbf{x}_{k,n}] = \sigma\sqrt{\left(2\ ln\ \left(\frac{n+1}{k}\right)\right)}$. For the cases where the condition $\sqrt{ln\ \left(\frac{n+1}{k}\right)} - \sqrt{ln\ \left(\frac{n+1}{n+1-k}\right)} < \frac{\theta}{\sigma\sqrt{2}}$, the accuracy attained is either optimal or near-optimal, as indicated by the results in the table (denoted by Trial Nos. 5 and 6). Now, consider the results presented in the row denoted by Trial No. 7. In this case

where the CMOS positions were $\sigma\sqrt{(2\ ln\ (\frac{7}{1}))}$ and $\theta + \sigma\sqrt{(2\ ln\ (\frac{7}{6}))}$, the inequality of the condition imposed in Theorem 11 simplified to $1.002339 < 0.88388$, which is not valid. Observe that if $\sqrt{ln\ (\frac{n+1}{k})} - \sqrt{ln\ (\frac{n+1}{n+1-k})} > \frac{\theta}{\sigma\sqrt{2}}$, the Dual CMOS should be invoked in which the symmetric pairs are reversed to obtain the near-optimal Bayes' bound.

This concludes our study on the CMOS for the Rayleigh distribution.

5.3 The Gamma Distribution

The Gamma distribution is a continuous probability distribution with two parameters - a, a shape parameter and b, a scale parameter. Another parametrization which is commonly used in Bayesian statistics has the parameters α, the shape parameter and the inverse scale parameter or the rate parameter, $\beta = \frac{1}{b}$. The pdf of the Gamma distribution with the parameters a and b is:

$$\frac{1}{\Gamma(a)\ b^a}\ x^{a-1}e^{\frac{-x}{b}};\ a > 0,\ b > 0, \tag{5.16}$$

with mean ab and variance ab^2. The pdf of the Gamma distribution with the parameters a and β is:

$$\frac{\beta^\alpha}{\Gamma(\alpha)}\ x^{\alpha-1}e^{-\beta x};\ \alpha > 0,\ \beta > 0, \tag{5.17}$$

with mean $\frac{\alpha}{\beta}$ and variance $\frac{\alpha}{\beta^2}$. Unfortunately, the cumulative distribution function does not have a closed form expression [30, 43, 58].

A plot of the pdf of the Gamma distribution for different values for the shape and scale parameters is shown in Figure 5.3.

5.3.1 Theoretical Analysis: Gamma Distribution - 2-OS

The typical PR problem invoking the Gamma distribution would consider two classes ω_1 and ω_2 where the class ω_2 is displaced by a quantity θ, and in the case analogous to the ones we have analyzed, the values of the scale and shape parameters are identical. As in the previous cases, we consider the scenario when $a_1 = a_2 = a$ and $b_1 = b_2 = b$.

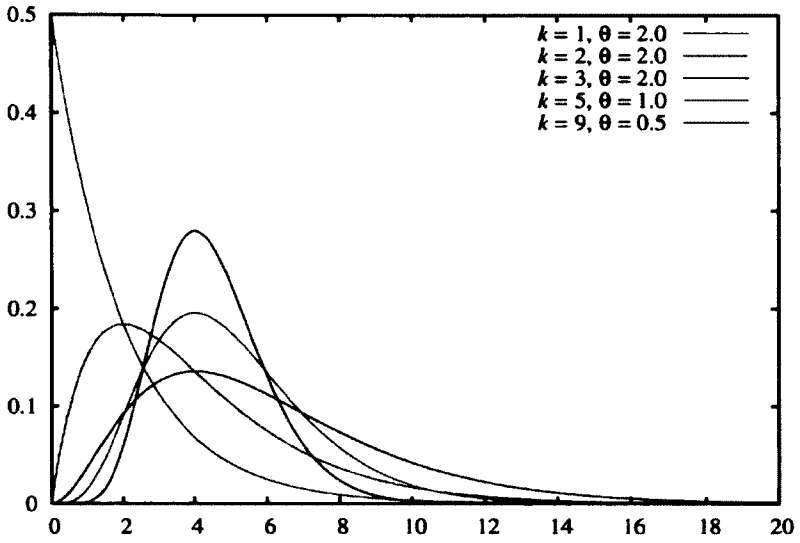


Figure 5.3: The Gamma Distribution for different parameters (from www.wikipedia.org.).

In the interest of simplicity[3], consider the distributions: $f(x, 2, 1) = x\ e^{-x}$ and $f(x - \theta, 2, 1) = (x - \theta)\ e^{-(x-\theta)}$.

We first derive the moments of the 2-OS, which are the points of interest for CMOS, for the Gamma distribution. By virtue of Eq. (3.3) and (3.4), the expected values of the first moments of the two OS can be obtained by determining the points where the cumulative distribution function attains the values of $\frac{1}{3}$ and $\frac{2}{3}$ respectively. Let o_1 be the point for the percentile $\frac{2}{3}$ of the first distribution, and o_2 be the point for the percentile $\frac{1}{3}$ of the second distribution. Then:

$$\int_0^{o_1} x\ e^{-x} dx = \frac{2}{3} \implies 1 - o_1 e^{-o_1} - e^{-o_1} = \frac{2}{3}$$
$$\implies ln(o_1) - 2o_1 = ln\left(\frac{1}{3}\right). \tag{5.18}$$

By a similar argument, the CMOS point for the $\frac{1}{3}$ percentile of the second distribution leads to the equation:

$$ln(o_2 - \theta) - 2(o_2 - \theta) = ln\left(\frac{1}{3}\right) - ln(\theta). \tag{5.19}$$

[3]Since we are resorting to a numerical strategy, the method of analysis given below will work for any other values of a and b.

We now prove that the classification with regard to the CMOS points are almost identical to the classification based on the median[4].

For the identical Gamma distributions with parameters 2 and 1, without loss of generality, let the distributions of ω_1 and ω_2 be $G(x, 2, 1)$ and $G(x - \theta, 2, 1)$, where θ is the displacement. As the curve of the Gamma distribution is not symmetric, we shall consider the scenario that the classification is done based on the median, which is the most central point of the distribution, other than the mean. The claim can be stated as in Theorem 12.

Theorem 12. *For the 2-class problem in which the two class conditional distributions are Gamma and identical with $a = 2$ and $b = 1$, the accuracy obtained by CMOS, the classification using two OS, deviates from the accuracy attained by the classifier with regard to the distance from the corresponding medians as the areas under the error curves deviate from the positions* $1.7391 + \frac{\theta}{2}$ *and* $1.6783 + \frac{\theta}{2}$.

Proof. The claim of this theorem is that CMOS classification can attain an accuracy which is almost identical to the one obtained with regard to the corresponding medians of the distributions.

As Eqs. (5.18) and (5.19) are of a transcendental nature, they cannot be simplified further, and hence it is not possible to obtain a closed form expression for the CMOS positions. The reason for this phenomenon is that the Gamma distribution lacks a closed form expression for its cumulative distribution function. Consequently, the only possible way by which we can proceed further to prove the claim is through a numerical formulation. The 2-OS CMOS positions o_1 and o_2 for $\Gamma(x, 2, 1)$ and $\Gamma(x - \theta, 2, 1)$ can be obtained by making use of the built-in functions available in standard software packages [11] as $o_1 = 2.2893$ and $o_2 = \theta + 1.1888$. Also, we can obtain the values of ν_1 and ν_2 for the same distributions as $\nu_1 = 1.6783$ and $\nu_2 = 1.6783 + \theta$ respectively. Then, the classifier with regard to the medians of the

[4]The Bayes' classifier, in this case when we are only using the 2-OS, is more distant than the CMOS, because of the skewed asymmetric form of the Gamma distribution. However, as we shall see later, other k-OS CMOS classifiers become more near-optimal.

distributions can be obtained as:

$$\begin{aligned} D(x,\nu_1) < D(x,\nu_2) &\implies D(x,1.6783) < D(x,1.6783+\theta) \\ &\implies x - 1.6783 < 1.6783 + \theta - x \\ &\implies x < 1.6783 + \frac{\theta}{2}. \end{aligned} \tag{5.20}$$

The discriminant function with regard to the medians of the distributions is thus:

$$x = 1.6783 + \frac{\theta}{2}. \tag{5.21}$$

We now consider the classification with respect to the expected values of the moments of the 2-OS. We can see that the discriminant enforced by 2-OS CMOS is $D(x,o_1) = D(x,o_2)$ which can further be simplified to:

$$\begin{aligned} D(x,o_1) = D(x,o_2) &\implies D(x,2.2893) = D(x,1.1888+\theta) \\ &\implies x - 2.2893 = 1.1888 + \theta - x \\ &\implies x = 1.7391 + \frac{\theta}{2}. \end{aligned} \tag{5.22}$$

The difference in the errors of the two classifiers is clearly related to differences in the corresponding discriminant functions specified by Eq. (5.23) and (5.22). □

Remark: As in the case of the Rayleigh distribution, we can show that the resulting classifiers are almost identical by considering the differences of the error probabilities quantified by the differences between the areas under the curves of the resulting errors. These error differences can be calculated by evaluating the corresponding definite integrals. Since closed form expressions for the integrals are not available, this has to be achieved numerically. The maximum area differences created by the CMOS classifier and the classifier based on the medians of the distributions for different values of θ are listed in Table 5.6. The claim of Theorem 12 is thus justified.

5.3.2 Data Generation: Gamma Generation

There are a number of data generation algorithms reported for the Gamma distribution, all of which make use of the Uniform random variate $U[0,1]$. In our experiments, data was generated using the built-in function available in MatLab, namely

θ	1	1.5	2	2.5	3
Max. Bounded Error(in %)	0.71	0.95	0.90	0.83	0.23

Table 5.6: Maximum bounded error by the CMOS classifier when compared to the classifier with regard to the medians, for different values of θ of the Gamma Distribution.

$gamrnd(a, b, sz)$, where a is the shape parameter, b is the scale parameter, and sz is the size of the array. To be specific, $gamrnd(2, 1, 10)$ will generate 100 values that follow the Gamma distribution with the shape parameter 2 and the scale parameter 1. For our experiments, we generated 1,000 points for each of the distributions, where the second distribution was displaced by a constant, θ.

5.3.3 Experimental Results: Gamma Distribution - 2-OS

The CMOS classifier was rigorously tested for a number of experiments for various Gamma distributions having the identical shape and scale parameters $a_1 = a_2 = 2$, and $b_1 = b_2 = 1$. In every case, the 2-OS CMOS gave almost the same classification as that of the classifier based on the central moments, namely, the mean and the median. The method was executed 50 times with the 10-fold cross validation scheme. The test results are tabulated in Table 5.7.

5.3.4 Theoretical Analysis: Gamma Distribution - k-OS

We have already seen in Theorem 12 that the 2-OS CMOS can obtain classification accuracy comparable to that obtained by comparing the testing sample with respect to the medians of the distributions. We shall now move on to examine the k-OS CMOS.

For the sake of the argument, let the distributions of ω_1 and ω_2 be $G(x, 2, 1)$ and $G(x - \theta, 2, 1)$, where θ is the displacement. Then, our claim for the k-OS can be stated as in Theorem 13.

n	Median	CMOS
4.5	94.825	95.01
4.0	94.25	94.49
3.5	92.74	92.915
3.0	90.765	90.425
2.5	86.51	85.985
2.0	80.145	79.54
1.5	72.64	72.34

Table 5.7: A comparison of the accuracy with respect to the median and the 2-OS CMOS classifier for the Gamma Distribution.

Theorem 13. *For the 2-class problem in which the two class conditional distributions are Gamma and identical, the classification obtained by using certain symmetric pairs of the n-OS, i.e., the $(n-k)^{th}$ OS for ω_1 (represented as o_1) and the k^{th} OS for ω_2 (represented as o_2) is arbitrarily close to the classification based on the medians if and only if $o_1 < o_2$. If $o_1 > o_2$, CMOS involves invoking the* Dual *n-OS pairs, i.e., the k^{th} OS for ω_1 and the $(n-k)^{th}$ OS for ω_2.*

Proof. We shall now extend the result of Theorem 12 for k-OS, so as to determine if the CMOS classification is almost identical to the classification based on the medians. Let o_1 be the point for the percentile $\frac{n+1-k}{n+1}$ (the $(n-k)^{th}$-OS) of the first distribution, and o_2 be the point for the percentile $\frac{k}{n+1}$ (the k-OS) of the second distribution. Our task is to compare the classification with respect to the CMOS positions and with the classifier obtained with regard to the medians of the distributions. The classifier with regard to the medians of the distributions can be obtained as:

$$\begin{aligned} D(x,\nu_1) < D(x,\nu_2) &\implies D(x,1.6783) < D(x,1.6783+\theta) \\ &\implies x - 1.6783 < 1.6783 + \theta - x \\ &\implies x < 1.6783 + \frac{\theta}{2}. \end{aligned} \tag{5.23}$$

Again, in order to compare the Bayesian, CMOS and median classifiers, as in the

2-OS case, we can provide a numerical comparison of the schemes by evaluating the differences of the error probabilities quantified by the differences between the areas under the curves of the resulting errors. If we proceed in this manner, we are to first obtain the values for the CMOS positions for different k-OS, and these are tabulated in Table 5.8.

n	Percentile	CMOS	n	Percentile	CMOS
2	$\frac{1}{2}$	1.6783	Corresponds to the Median		
3	$\frac{1}{3}$	1.1888	3	$\frac{2}{3}$	2.2893
5	$\frac{1}{5}$	0.8244	5	$\frac{2}{5}$	1.3764
5	$\frac{3}{5}$	2.0223	5	$\frac{4}{5}$	2.9943
7	$\frac{1}{7}$	0.6624	7	$\frac{2}{7}$	1.0584
7	$\frac{3}{7}$	1.4596	7	$\frac{4}{7}$	1.9183
7	$\frac{5}{7}$	2.5077	7	$\frac{6}{7}$	3.4356
9	$\frac{1}{9}$	0.5669	9	$\frac{2}{9}$	0.8855
9	$\frac{4}{9}$	1.5068	9	$\frac{5}{9}$	1.8627
9	$\frac{7}{9}$	2.8529	9	$\frac{8}{9}$	3.7568

Table 5.8: CMOS positions for Gamma distribution $\Gamma(2, 1)$ for different percentiles.

With these values on hand, we can now verify the claim that the CMOS classifier and its median-based counterpart are almost identical for different values of k, n and θ by computing the respective errors areas that they yield. For different values of θ, the areas are tabulated in Table 5.9 for certain specific CMOS pairs $(k, n - k + 1)$. From Table 5.9, the reader can observe that the classifiers are almost identical.

We thus conclude that a classification that is arbitrarily close to the one obtained by comparing to the medians can be achieved by using certain symmetric pairs of the n-OS, i.e., the $(n - k)^{th}$ OS for ω_1 (represented as o_1) and the k^{th} OS for ω_2 (represented as o_2).

The proof of the case when the *Dual* condition has to be invoked follows in an

θ	2	2.5	3	3.5	4
n	8	8	8	8	6
k	4	4	4	2	1
Max. Bounded Error	0.11	0.08	0.0040	0.01	0.01

Table 5.9: Maximum bounded error (in %) by the CMOS classifier when compared to the classifier with regard to the medians of the distributions, for different values of θ, k and n of the Gamma Distribution.

identical manner and is omitted here. □

Remark: Similar results are available for the comparison of CMOS and the corresponding Bayesian classifier. To get the Bayes' classifier, we argue that:

$$\begin{aligned} p(x|\omega_1)P(\omega_1) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} p(x|\omega_2)P(\omega_2) &\implies x\, e^{-x} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} (x-\theta)\, e^{-(x-\theta)} \\ &\implies \frac{x-\theta}{x} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} e^{-\theta} \\ &\implies x \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \frac{\theta}{1-e^{-\theta}}, \end{aligned} \tag{5.24}$$

whence the differences between the areas under the curves can be evaluated and seen to be almost negligible. The details are omitted here to avoid repetition.

5.3.5 Experimental Results: Gamma Distribution - k-OS

The CMOS method has been rigorously tested for numerous symmetric pairs of the k-OS and for various values of n, and the test results are given in Table 5.10. Experiments have been performed for different values of θ, and we can see that the CMOS attained a near-optimal Bayes' accuracy. Also, we can see that the Dual CMOS has to be invoked if the condition stated in Theorem 13 is not satisfied.

By way of example, consider the case in Trial # 10 when $\theta = 3.0$, where the condition $o_1 < o_2$ was not satisfied. Here, as the condition yielded an invalid inequality, i.e., $3.7568 < 3.5669$, the Dual CMOS has to be invoked by reversing the CMOS

No.	Classifier	Moments	$\theta = 4.5$	4.0	3.5	3.0	2.5	2.0
1	Bayes	-	97.06	95.085	93.145	90.68	86.93	81.53
2	Mean	-	96.165	94.875	92.52	88.335	83.105	77.035
3	Median	-	96.04	93.57	92.735	90.775	86.275	80.115
4	2-OS	$(\frac{2}{3}, \frac{1}{3})$	95.285	93.865	92.87	90.61	86.085	79.48
5	4-OS	$(\frac{4}{5}, \frac{1}{5})$	95.905	94.605	93.11	89.57	84.68	77.875 (D)
6	4-OS	$(\frac{3}{5}, \frac{2}{5})$	95.185	93.675	92.82	**90.855**	86.02	**80.32**
7	6-OS	$(\frac{6}{7}, \frac{1}{7})$	96.405	**95.01**	92.125	88.005	82.71 (D)	76.435 (D)
8	6-OS	$(\frac{5}{7}, \frac{2}{7})$	95.47	94.11	**93.135**	90.16	85.495	79.55
9	6-OS	$(\frac{4}{7}, \frac{3}{7})$	95.135	93.625	92.78	90.745	**86.135**	80.165
10	8-OS	$(\frac{8}{9}, \frac{1}{9})$	**96.815**	94.895	91.555	86.905 (D)	80.59 (D)	75.94 (D)
11	8-OS	$(\frac{7}{9}, \frac{2}{9})$	95.8	94.445	93.11	89.885	84.81	78.535
12	8-OS	$(\frac{5}{9}, \frac{4}{9})$	95.135	93.625	92.735	90.7	86.085	80.045

Table 5.10: A comparison of the k-OS CMOS classifier when compared to the Bayes' classifier and the classifier with respect to the median and mean for the Gamma Distribution for different values of n. In each column, the value which is near-optimal is rendered bold. The scenarios when we have invoked the *Dual* condition are specified by noting them using the notation "(D)".

values to obtain the near-optimal accuracy. Interestingly enough, if we examine the table, we can see that the Bayes' accuracy is the highest for all cases except for the scenario when $\theta = 3.0$. This result must, in fact, be be considered as an aberration.

This concludes the study of the Gamma distribution with regard to the CMOS classification.

5.4 The Beta Distribution

In Chapter 4, we discussed the symmetric Beta distribution when the values of the shape parameters α and β are identical. We now move on to the unimodal Beta distribution characterized by the shape parameters $\alpha > 1$ and $\beta > 1$, $\alpha \neq \beta$.

5.4.1 Theoretical Analysis: Beta Distribution ($\alpha > 1, \beta > 1$) - 2-OS

Consider the two classes ω_1 and ω_2 where the class ω_2 is displaced by a quantity θ. In this section, we consider the case when the shape parameters take the values $\alpha > 1$ and $\beta > 1$, and for the interest of preciseness[5], we consider the case when $\alpha = 2$ and $\beta = 5$. Then, the distributions are:

$$f(x, 2, 5) = 30x(1-x)^4 \tag{5.25}$$

and

$$f(x-\theta, 2, 5) = 30(x-\theta)(1-x+\theta)^4. \tag{5.26}$$

We first derive the moments of the 2-OS, namely o_1 and o_2 where o_1 represents the point for the percentile $\frac{2}{3}$ of the first distribution, and o_2 represents the point for the percentile $\frac{1}{3}$ of the second distribution. Then:

$$\int_0^{o_1} 30x(1-x)^4 dx \;=\; \frac{2}{3} \tag{5.27}$$

and

$$\int_0^{o_2} 30(x-\theta)(1-x+\theta)^4 dx \;=\; \frac{1}{3}. \tag{5.28}$$

These positions o_1 and o_2 can be obtained by making use of the built-in functions available in standard software packages [11] as $o_1 = 0.34249$ and $o_2 = \theta + 0.1954$. Thus, our aim is to show that the classification based on these points can attain near optimal accuracies when compared to the accuracy obtained by the classifier with regard to the medians, the most central points of the distributions.

Theorem 14. *For the 2-class problem in which the two class conditional distributions are Beta(α, β) ($\alpha > 1,\ \beta > 1$) and identical with $\alpha = 2$ and $\beta = 5$, the accuracy obtained by CMOS, the classification using two OS, deviates from the accuracy attained by the classifier with regard to the distance from the corresponding medians as the areas under the error curves deviate from the positions* $0.26445 + \frac{\theta}{2}$ *and* $0.2689 + \frac{\theta}{2}$.

[5]Any analysis will clearly have to involve specific values for α and β. The analyses for other values of α and β will follow the same arguments and are not included here.

Proof. For the Beta distributions under consideration, without loss of generality, let the distributions of ω_1 and ω_2 be $B(x, 2, 5)$ and $B(x - \theta, 2, 5)$. As already stated, the 2-OS CMOS positions and the medians of the distributions can be obtained using the standard software packages, whence we can determine that $o_1 = 0.34249$, $o_2 = \theta + 0.1954$, $\nu_1 = 0.26445$, and $\nu_2 = \theta + 0.26445$, where ν_1 and ν_2 are the medians of the distributions. The claim of this part is that CMOS classification can attain an accuracy which is almost identical to the one obtained with regard to the corresponding medians of the distributions.

Using the values of the medians of the distributions, the classifier can be obtained as:

$$\begin{aligned} D(x, \nu_1) < D(x, \nu_2) &\implies D(x, 0.26445) < D(x, 0.26445 + \theta) \\ &\implies x - 0.26445 < 0.26445 + \theta - x \\ &\implies x < 0.26445 + \frac{\theta}{2}. \end{aligned} \tag{5.29}$$

Thus, the discriminant function with respect to the medians of the distributions is:

$$x = 0.26445 + \frac{\theta}{2}. \tag{5.30}$$

If we consider the classifier with regard to the expected moments of the 2-OS, we can see that the discriminant enforced by 2-OS CMOS is $D(x, o_1) = D(x, o_2)$, which simplifies to:

$$\begin{aligned} D(x, o_1) = D(x, o_2) &\implies D(x, 0.34249) = D(x, 0.1954 + \theta) \\ &\implies x - 0.34249 = 0.1954 + \theta - x \\ &\implies x = 0.2689 + \frac{\theta}{2}. \end{aligned} \tag{5.31}$$

The difference in the errors of the two classifiers is clearly related to differences in the corresponding discriminant functions specified by Eq. (5.30) and (5.31). Hence the theorem. □

Remark:

1. As in the other asymmetric distributions, we can quantify the differences of the error probabilities by the differences between the areas under the curves of the

resulting errors of the considered Beta distributions. The maximum bounded error by the CMOS classifier when compared to the classifier with regard to the medians, for different values of θ, are tabulated in Table 5.11. From this table, we can conclude that the classifiers obtained with respect to the medians and the CMOS positions are almost indistinguishable. Clearly, CMOS, the classification using two OS, attains the near-identical bound obtained by comparison to the corresponding medians.

θ	0.30	0.35	0.4	0.45	0.5	0.55
Max. Bounded Error(in %)	0.3	0.25	0.17	0.03	0.16	0.21

Table 5.11: Maximum bounded error by the CMOS classifier when compared to the classifier with regard to the medians, for different values of θ of the Beta Distribution.

2. A similar comparison can be done by considering the efficiency of CMOS and comparing it with the optimal *Bayesian* classifier. Again, the accuracies of the two are almost indistinguishable, and the details are omitted here to avoid repetition.

5.4.2 Experimental Results: Beta Distribution ($\alpha > 1, \beta > 1$) - 2-OS

The CMOS has been rigorously tested for various Beta distributions with 2-OS. For each of the experiments, we generated 1,000 points for the classes ω_1 and ω_2 characterized by $B(x, 2, 5)$ and $B(x-\theta, 2, 5)$ respectively. We then performed the classification based on the CMOS strategy and with regard to the medians of the distributions. In every case, CMOS was compared with the accuracy obtained with respect to the medians for different values of θ, as tabulated in Table 5.12. The results were obtained by executing each algorithm 50 times using a 10-fold cross-validation scheme. The quality of the classifier is obvious.

θ	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8
Median	89.625	92.9	94.3	95.525	97.3	97.975	98.375	99.05	99.15
CMOS	89.475	92.775	94.525	95.75	97.3	98.05	98.375	99.2	99.225

Table 5.12: A comparison of the accuracy of the 2-OS CMOS classifier with the classification with respect to the medians for the Beta Distribution for different values of θ.

5.4.3 Theoretical Analysis: Beta Distribution ($\alpha > 1, \beta > 1$) - k-OS

We have seen in Theorem 14 that the 2-OS CMOS can attain a near-optimal classification when compared to the classification obtained with regard to the medians of the distributions. We shall now prove that the k-OS CMOS can also attain almost indistinguishable bounds for some symmetric pairs of the n-OS. The formal theorem follows.

Theorem 15. *For the 2-class problem in which the two class conditional distributions are Beta(α, β) ($\alpha > 1$, $\beta > 1$) and identical with $\alpha = 2$ and $\beta = 5$, a near-optimal classification can be achieved by using certain symmetric pairs of the n-OS, i.e., the $(n-k)^{th}$ OS for ω_1 (represented as o_1) and the k^{th} OS for ω_2 (represented as o_2) if and only if $o_1 < o_2$. If this condition is violated, the CMOS classifier uses the* Dual *condition, i.e., the k OS for ω_1 and the $n-k$ OS for ω_2.*

Proof. In order to prove this claim, we shall now extend the result of Theorem 14 for k-OS, that the classification is almost identical to the classification based on the medians. Let o_1 be the point for the percentile $\frac{n+1-k}{n+1}$ (the $(n-k)^{th}$-OS) of the first distribution, and o_2 be the point for the percentile $\frac{k}{n+1}$ (the k-OS) of the second distribution. We have to compare the CMOS classifier with the classifier obtained with regard to the medians of the distributions. To achieve this, we need to obtain the values for the CMOS positions for different k-OS, and these are tabulated in Table 5.13.

n	Percentile	CMOS	n	Percentile	CMOS
2	$\frac{1}{2}$	0.26445	Corresponds to the Median		
3	$\frac{1}{3}$	0.1954	3	$\frac{2}{3}$	0.3425
5	$\frac{1}{5}$	0.1399	5	$\frac{2}{5}$	0.2226
5	$\frac{3}{5}$	0.3095	5	$\frac{4}{5}$	0.4225
7	$\frac{1}{7}$	0.1140	7	$\frac{2}{7}$	0.1760
7	$\frac{3}{7}$	0.2344	7	$\frac{4}{7}$	0.2961
7	$\frac{5}{7}$	0.3684	7	$\frac{6}{7}$	0.4675
9	$\frac{1}{9}$	0.0984	9	$\frac{2}{9}$	0.1495
9	$\frac{4}{9}$	0.2409	9	$\frac{5}{9}$	0.2889
9	$\frac{7}{9}$	0.4072	9	$\frac{8}{9}$	0.4982

Table 5.13: CMOS positions for Beta distribution $B(2,5)$ for different percentiles.

With these values on hand, we can verify the claim that the CMOS classifier and its Bayesian counterpart are almost identical for different values of k, n and θ by computing the differences of the error probabilities quantified by the differences between the areas under the curves of the resulting errors of the respective distributions. The computed areas are depicted in Table 5.14 for certain CMOS pairs $(k, n-k+1)$ for different values of θ. From the tabulated values, we can see that the classifiers are almost identical.

The arguments are identical for the case when the Dual condition has to be invoked and are omitted here. The result follows. □

5.4.4 Experimental Results: Beta Distribution ($\alpha > 1, \beta > 1$) - k-OS

The CMOS method has been rigorously tested for certain symmetric pairs of the k-OS and for various values of n, and the test results are given in Table 5.15. Various

θ	0.35	0.45	0.55	0.65
n	8	4,6,8	8	8
k	4	2,3,4	4	4
Max. Bounded Error	0.02	0	0.03	0.02

Table 5.14: Maximum bounded error (in %) by the CMOS classifier when compared to the classifier with regard to the medians of the distributions, for different values of θ, k and n of the Beta Distribution.

experiments were performed for different values of θ, and from them, we can see that CMOS attained a near-optimal Bayes' accuracy. Also, we can see that the Dual CMOS has to be invoked if the condition stated in Theorem 15 is not satisfied.

No.	**Classifier**	**Moments**	$\theta =$ **0.35**	**0.45**	**0.55**	**0.65**	**0.75**	**0.85**
1	Mean	-	85.325	92.575	96.55	98.3	99.4	99.475
2	Median	-	86.675	92.775	95.525	97.975	99.05	99.275
3	2-OS	$(\frac{2}{3}, \frac{1}{3})$	86.2	92.575	95.75	98.05	99.2	99.275
4	4-OS	$(\frac{4}{5}, \frac{1}{5})$	85.375	92.525	96.225	98.225	99.325	99.475
5	4-OS	$(\frac{3}{5}, \frac{2}{5})$	86.475	**92.775**	95.6	98.05	99.125	99.275
6	6-OS	$(\frac{6}{7}, \frac{1}{7})$	85.2 (D)	92.425	**96.475**	**98.35**	99.45	99.625
7	6-OS	$(\frac{5}{7}, \frac{2}{7})$	86.125	92.625	96.0	98.075	99.2	99.275
8	6-OS	$(\frac{4}{7}, \frac{3}{7})$	86.55	**92.775**	95.525	97.975	99.125	**99.75**
9	8-OS	$(\frac{8}{9}, \frac{1}{9})$	84.225 (D)	92.225	96.225	98.35	**99.5**	99.375
10	8-OS	$(\frac{7}{9}, \frac{2}{9})$	85.675	92.5	96.175	98.15	99.325	99.375
11	8-OS	$(\frac{5}{9}, \frac{4}{9})$	**86.575**	**92.775**	95.525	97.975	99.125	99.275

Table 5.15: A comparison of the k-OS CMOS classifier when compared to the classifier with respect to means and medians for the Beta Distribution for different values of n. The scenarios when we have invoked the *Dual* condition are specified by noting them using the notation "(D)".

For example, if we examine Table 5.15, we see that CMOS attained the near-optimal value for certain k-OS when compared to the accuracy obtained with regard to the medians of the distributions. However, if we consider the case in Trial # 9 when $\theta = 0.35$, where the condition $o_1 < o_2 \implies 0.46753 < 0.46401$. In such cases, the Dual CMOS (CMOS values have to be reversed) has to be invoked in order to yield near-optimal accuracy.

This concludes the study of the Beta distribution with regard to the CMOS classification.

5.5 Conclusions

In this chapter, we have extended the results of Chapter 3 and Chapter 4 which earlier showed that optimal classification can be attained for various distributions of the exponential family by an "anti-Bayesian" approach by working with a *very few* points *distant* from the mean. In those chapters, we explained how this scheme, referred to as CMOS, operates by using these points determined by the respective *Order Statistics* of the distributions. Whereas the work of Chapter 4 concentrated on various *symmetric* distributions in the exponential family, in this chapter we have worked with non-symmetric distributions such as the Rayleigh, Gamma and other Beta Distributions. From a rigorous error-analysis and experiment verification, we have demonstrated that the CMOS can attain near-optimal accuracies for these distributions.

We proceed to investigate CMOS classification for multi-dimensional distributions in the next chapter.

Chapter 6

Optimal "Anti-Bayesian" Parametric PR for Multi-dimensional Distributions Using OS Criteria

6.1 Introduction

In Chapters 3, 4 and 5, we had discussed CMOS, our newly proposed "anti"-Bayesian PR strategy for generic classification, for symmetric distributions from the exponential family, and for asymmetric distributions from the exponential family respectively. In this chapter[1] we generalize the results of Chapters 4 and 5 for multi-dimensional distributions. The reader must observe that since we are speaking about the order statistics of a distribution, it implicitly and explicitly assumes that the points can be *ordered*. Consequently, the multi-dimensional generalization of CMOS, theoretically and with regard to implementation, is particularly non-trivial because there is no well-established method for achieving the ordering of multi-dimensional data specified in terms of its uni-dimensional components. To clarify this, consider two

[1] A preliminary version of some of the results of this chapter can be found in the *Proceedings of CORES'13, the 2013 Conference on Computer Recognition Systems* held in Milkow, Poland in May 2013 [49].

patterns, $X_1 = [x_{11}, x_{12}]^T = [2, 3]^T$ and $X_2 = [x_{21}, x_{22}]^T = [1, 4]^T$. If we only considered the first dimension, x_{21} would be the first OS since $x_{11} > x_{21}$. However, if we observe the second component of the patterns, we can see that x_{12} would be the first OS. It is thus, clearly, not possible to obtain the ordering of the *vectorial* representation of the patterns based on their individual components, which is the fundamental issue to be resolved before the problem can be tackled in any satisfactory manner for multi-dimensional features.

To resolve this, in this chapter we propose a strategy for the multi-dimensional generalization analogous to a Naïve-Bayes' approach, although it really is of an *anti*-Naïve-Bayes' paradigm.

Using such a Naïve-Bayes' approach, we demonstrate how a CMOS classifier can be both designed and implemented. In order to prove our claims, we provide analytical and experimental results for the two-dimensional Uniform, Doubly-Exponential, and Gaussian distributions, whence we show that the results are clearly conclusive. We also clearly specify the way by which one could extend these results for higher dimensions and for the other distributions in the exponential family that were discussed in [38, 51]. To avoid repetition, these results are omitted and merely alluded to.

6.1.1 Contributions of this Chapter

The novel contributions of this chapter are:

- We extend the theory of OS-based PR for multi-dimensional features. To accomplish this, we generalize the uni-dimensional CMOS schemes proposed in Chapters 3, 4 and 5;
- From the perspective of a Naïve-Bayes' generalization of the uni-dimensional scheme, we demonstrate that the proposed multi-dimensional approach attains the optimal bound for symmetric distributions, and a near-optimal accuracy for asymmetric distributions;
- To justify these claims, we also submit a formal analysis and the results of

various experiments which have been performed for a few distributions within the exponential family;

- Although the results have been derived for the two-dimensional scenario, without going into the explicit details, the chapter explains the way by which one can extend these results for higher dimensions, and for the other distributions in the exponential family that were discussed in Chapters 3, 4 and 5.

As in the case of the uni-dimensional results, the pioneering nature and novelty of these multi-dimensional results are also true, to the best of our knowledge.

The multi-dimensional OS-based classifier is based on its uni-dimensional counterpart developed in [38] and [51]. Our task is to now generalize these results for multi-dimensional distributions. As mentioned earlier, the generalization of CMOS for multi-dimensional classification problems is not so trivial. We have opted to do this by invoking a Naïve-Bayes' approach, which essentially implies that the first moments of the OS in each of the dimensions are uncorrelated[2].

6.2 Uniform Distribution

Classification using the 2-OS and k-OS CMOS can attain the optimal Bayes' bound for many uni-dimensional distributions [51]. Now, we extend these results to show that CMOS can also attain a similar optimal bound for their multi-dimensional counterparts. In order to prove this claim, we first consider two-dimensional distributions, and thereafter, in a straightforward manner, extend the result for higher dimensions.

In [51], we initiated this discussion by examining the Uniform distribution. The reason for this was that even though the distribution itself is rather trivial, the analysis provided us with an insight into the mechanism by which the problem can be tackled, which can then be extended for other distributions. We shall follow the same *modus operandus* here.

[2]Although the uncorrelation is *sufficient*, we are not certain whether the *independence* of the features is *necessary*. As far as we are concerned, this is still an open issue.

6.2.1 Order Statistics: Uniform Distributions

For a *prima facie* case, we consider two (overlapping) 2-dimensional Uniform distributions U_1 and U_2 in which both the features are in $[0,1]^2$ and $[h,1+h]^2$ respectively. Consequently, we see that the overlapping region of the distributions forms a square (see Fig. 6.1). In this case, it is easy to verify that the Bayesian classifier is the line that passes through the intersection points of the distributions. For the classification based on the moments of the 2-OS, because the features are independent for both dimensions, we can show that this is equivalent to utilizing the OS at position $\frac{2}{3}$ of the first distribution for both dimensions, and the OS at the position $h+\frac{1}{3}$ of the second distribution for both dimensions, as shown in Figure 6.1. In the figure, the lines $x_1 = \frac{2}{3}$ and $x_1 = h+\frac{1}{3}$ denote the corresponding boundaries of the first feature respectively. Similarly, $x_2 = \frac{2}{3}$ and $x_2 = h+\frac{1}{3}$ denote the corresponding boundaries for the second feature of the given distributions. The relevant points in this regard are $A = (a_1, a_2)$, $B = (b_1, b_2)$, $C = (c_1, c_2)$, and $D = (d_1, d_2)$. Observe that the Bayesian classifier is the diagonal of the intersecting square.

One can easily observe that, in this case, uncorrelation implies independence, and thus a testing point is classified to ω_1 if the value of its first feature is less than a_1 and is classified as ω_2 if the feature value is greater than c_1. The classification of the points that have the feature value as $a_1 < x_1 < c_1$, are equally likely and should thus be considered more carefully by considering the feature x_2. By virtue of the uni-dimensional result derived in Theorem 1 of [51], we see that for *all* values of the second dimension, the classifier for the first dimension in this region lies at the midpoint of **a** and **c**, which is exactly the position determined by the Bayes' classifier. Arguing in the same manner for the second dimension, we see that the values defined by the corresponding OS criteria will again be projected exactly onto the Bayes' classifier. We formally prove this assertion below.

6.2.2 Theoretical Analysis - 2-OS: Uniform Distributions

Consider the points $O_1 = \left(\frac{2}{3}, \frac{2}{3}\right)$ of the first distribution and $O_2 = \left(h+\frac{1}{3}, h+\frac{1}{3}\right)$ of the second distribution in the two-dimensional space. We now show, in Theorem

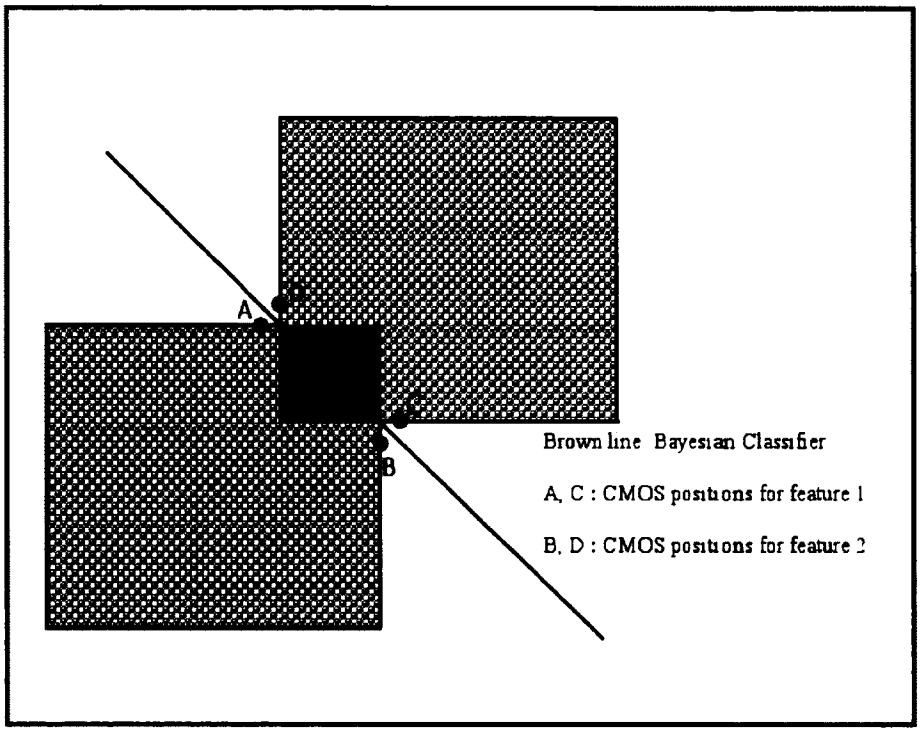


Figure 6.1: The optimal Bayes' classifier and the 2-OS CMOS for uniformly distributed 2-dimensional features. The coordinates of the axes are the respective features.

16, that if we compare the testing point $X = (x_1, x_2)$ with these points, the PR attains the optimal classification, i.e., that which is the result of comparing it with the corresponding means.

Theorem 16. *For the 2-class problem in which the two 2-dimensional class conditional distributions are Uniform and identical, CMOS, the classification using two OS, attains the optimal Bayes' bound.*

Proof. Without loss of generality, let the class conditional distributions for ω_1 and ω_2 be $U[0,1]^2$ and $U[h, 1+h]^2$, with means $M_1 = \left(\frac{1}{2}, \frac{1}{2}\right)$ and $M_2 = \left(h + \frac{1}{2}, h + \frac{1}{2}\right)$, respectively. From Theorem 1 of [51], we know that Bayesian classification for the unidimensional Uniform distribution is exactly identical to a classification with regard to the means. So, what remains to be proven is that a CMOS-based classification attains

exactly the same accuracy as that obtained when performing comparisons with the corresponding means in the 2-dimensional scenario. Thus, the claim is:

$$D(X, M_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(X, M_2) \implies D(X, O_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(X, O_2). \tag{6.1}$$

The LHS of Eq. (6.1) simplifies to:

$$\begin{aligned}
& D(X, M_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(X, M_2) \\
\implies & \sqrt{(x_1 - m_{11})^2 + (x_2 - m_{12})^2} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \sqrt{(x_1 - m_{21})^2 + (x_2 - m_{22})^2} \\
\implies & \sqrt{\left(x_1 - \frac{1}{2}\right)^2 + \left(x_2 - \frac{1}{2}\right)^2} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \sqrt{\left(x_1 - \left(h + \frac{1}{2}\right)\right)^2 + \left(x_2 - \left(h + \frac{1}{2}\right)\right)^2} \\
\implies & 2h\ (x_1 + x_2) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} 2h\ (h + 1) \\
\implies & x_1 + x_2 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} h + 1.
\end{aligned} \tag{6.2}$$

After some algebraic simplifications, we can see that the RHS of the claim given by Eq. (6.1) also reduces to the same expression since:

$$\begin{aligned}
& D(X, O_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(X, O_2) \\
\implies & \sqrt{(x_1 - o_{11})^2 + (x_2 - o_{12})^2} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \sqrt{(x_1 - o_{21})^2 + (x_2 - o_{22})^2} \\
\implies & \sqrt{\left(x_1 - \frac{2}{3}\right)^2 + \left(x_2 - \frac{2}{3}\right)^2} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \sqrt{\left(x_1 - \left(h + \frac{1}{3}\right)\right)^2 + \left(x_2 - \left(h + \frac{1}{3}\right)\right)^2} \\
\implies & 2(x_1 + x_2)\left(h - \frac{1}{3}\right) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} 2\left(h^2 + \frac{2}{3}h - \frac{3}{9}\right) \\
\implies & 2(x_1 + x_2)\left(h - \frac{1}{3}\right) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} 2\left(h - \frac{1}{3}\right)(h + 1) \\
\implies & x_1 + x_2 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} h + 1.
\end{aligned} \tag{6.3}$$

It is interesting to observe that in the above expressions, the radicals and higher order terms of the variables on both sides are identical and so cancel each other and disappear. This is because they appear *individually* and not as *joint-product* terms.

The result follows by observing that Eq. (6.2) and (6.3) are identical comparisons.

□

Remark: From Figure 6.1, one can observe that from a geometric perspective, the *perpendicular bisector* of the line joining the points $\left(\frac{2}{3}, \frac{2}{3}\right)$ and $\left(h + \frac{1}{3}, h + \frac{1}{3}\right)$ is exactly the Bayesian classifier. This is, intuitively, quite appealing.

6.2.3 Experimental Results - 2-OS: Uniform Distributions

The CMOS method for 2-dimensional Uniform distributions U_1 (in $[0, 1]$ in both dimensions) and U_2 (in $[h, 1 + h]$ in both dimensions) has been rigorously tested, and the results are given in Table 6.1. For each of the experiments, we generated 1,000 points for the classes ω_1 and ω_2. In every case, the 2-OS CMOS gave *exactly* the same classification as that of the Bayesian classifier. The method was executed 50 times with the 10-fold cross validation scheme.

h	0.95	0.90	0.85	0.80	0.75	0.70	0.65	0.60
Bayesian	99.845	99.505	98.875	98.045	97.15	95.555	94.14	91.82
CMOS	99.845	99.505	98.875	98.045	97.15	95.555	94.14	91.82

Table 6.1: Classification of Uniformly distributed 2-dimensional classes by the CMOS 2-OS method for different values of h. The accuracy obtained in every case is exactly the Bayes' bound.

6.2.4 Theoretical Analysis - k-OS: Uniform Distributions

We shall now discuss the efficiency of the k-OS CMOS. In Theorem 2 of [51], we proved that the k-OS CMOS can attain the optimal bound for uni-dimensional Uniform distributions provided that the condition imposed by the OSs being not "reversed" is satisfied. The same result can be obtained for 2-dimensional Uniformly distributed features. As in the case of the 2-OS CMOS, by enforcing the fact that the features are independent, we can, indeed, prove that the k-OS CMOS also attains the optimal Bayesian bound.

Theorem 17. *For the 2-class problem in which the two class conditional distributions are Uniform and identical as $U[0,1]^2$ and $U[h,1+h]^2$, optimal Bayesian classification can be achieved by using symmetric pairs of the n-OS, i.e., the $n-k$ OS for ω_1 (represented by O_1) and the k OS for ω_2 (represented by O_2) if and only if $k > \frac{(n+1)(1-h)}{2}$ for both the features. If this condition is violated, the CMOS classifier uses the* Dual *condition, i.e., the k OS for ω_1 and the $n-k$ OS for ω_2 for both the features.*

Proof. We know that the symmetric CMOS pairs of the n-OS, i.e., $n-k$ OS for ω_1 and the k OS for ω_2 have the form $\frac{n-k+1}{n+1}$ and $h+\frac{k}{n+1}$ respectively. We claim that the classification with regard to these points can also attain the optimal Bayes' bound. Thus the claim is again:

$$D(X,M_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(X,M_2) \implies D(X,O_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(X,O_2). \tag{6.4}$$

In Theorem 16, we have shown that the LHS of the claim given in Eq. (6.4) simplifies to:

$$x_1 + x_2 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} h+1. \tag{6.5}$$

What remains to be proven is that the RHS of the claim also leads to the same condition. This can be seen to be true since:

$$\begin{aligned}
& D(X,O_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(X,O_2) \\
\implies & \sqrt{(x_1-o_{11})^2+(x_2-o_{12})^2} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \sqrt{(x_1-o_{21})^2+(x_2-o_{22})^2} \\
\implies & \sqrt{\left(x_1-\frac{n+1-k}{n+1}\right)^2+\left(x_2-\frac{n+1-k}{n+1}\right)^2} \\
\underset{\omega_2}{\overset{\omega_1}{\lessgtr}} & \sqrt{\left(x_1-\left(h+\frac{k}{n+1}\right)\right)^2+\left(x_2-\left(h+\frac{k}{n+1}\right)\right)^2} \\
\implies & 2(x_1+x_2)\left(h-1+\frac{2k}{n+1}\right) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} 2\left(h-1+\frac{2k}{n+1}\right)(h+1) \\
\implies & x_1+x_2 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} h+1.
\end{aligned} \tag{6.6}$$

Again, we can see that as the radicals and higher order terms on both sides are not joint-product terms, they cancel each other. The result follows by observing that Eq. (6.5) and (6.6) are identical comparisons. □

6.2.5 Experimental Results - k-OS: Uniform Distributions

The k-OS CMOS method for 2-dimensional Uniform distributions U_1 (in $[0,1]^2$) and U_2 (in $[h,1+h]^2$) has been rigorously tested, and the results are given in Table 6.2. For each of the experiments, we generated 1,000 points for the classes ω_1 and ω_2. In every case, the k-OS CMOS gave exactly the same classification as that of the Bayesian classifier for different values of k and n. The method was executed 50 times with the 10-fold cross validation scheme.

$h \longrightarrow$	**0.95**	**0.90**	**0.85**	**0.80**	**0.75**	**0.70**	**0.65**
$\langle \frac{2}{3}, \frac{1}{3} \rangle$	99.92	99.58	98.86	97.94	96.78	95.69	93.73
$\langle \frac{4}{5}, \frac{1}{5} \rangle$	99.92	99.58	98.86	97.94	96.78	95.69	93.73
$\langle \frac{6}{7}, \frac{1}{7} \rangle$	99.92	99.58	98.86	97.94	96.78	95.69 (D)	93.73 (D)
$\langle \frac{5}{7}, \frac{2}{7} \rangle$	99.92	99.58	98.86	97.94	96.78	95.69	93.73
$\langle \frac{4}{7}, \frac{3}{7} \rangle$	99.92	99.58	98.86	97.94	96.78	95.69	93.73
$\langle \frac{8}{9}, \frac{1}{9} \rangle$	99.92	99.58	98.86	97.94	96.78 (D)	95.69 (D)	93.73 (D)
$\langle \frac{7}{9}, \frac{2}{9} \rangle$	99.92	99.58	98.86	97.94	96.78	95.69	93.73
$\langle \frac{5}{9}, \frac{4}{9} \rangle$	99.92	99.58	98.86	97.94	96.78	95.69	93.73

Table 6.2: Classification of Uniformly distributed 2-dimensional classes by the k-OS CMOS method for different values of h, which is where the second distribution starts in each dimension. The scenarios when we have invoked the *Dual* condition are specified by noting them using the notation "(D)".

If we examine Table 6.2, we can see that k-OS CMOS attained the optimal Bayes' bound for all the cases where the condition is strictly enforced. But, for the cases where the condition failed, the Dual condition holds and so the CMOS positions should be reversed so as to attain the optimal accuracy. For example, consider the classification with the CMOS positions, $\langle \frac{8}{9}, \frac{1}{9} \rangle$ for $h = 0.75$. As stated earlier, for any symmetric pair, the condition which is to be enforced is that the $\frac{n-k+1}{n+1}^{th}$ percentile should be less than the $\frac{k}{n+1}^{th}$ percentile for *every* dimension. But, for the symmetric

pairs $\langle \frac{8}{9}, \frac{1}{9} \rangle$, this is not true, and hence, to obtain optimal classification, the pairs should be reversed.

6.2.6 Multi-dimensional Extension: Uniform Distributions

We have now shown that the CMOS can attain exactly the same classification as that of the Bayes' classifier for two-dimensional Uniform distributions. With some insight, one can see that this result can easily be extended to identical multi-dimensional Uniform distributions too. Since the multi-dimensional distribution naturally imposes the independence for the Uniform scenario, we can extend the result of Theorems 16 and 17 to obtain a classifier for the higher-dimensional problem.

Theorem 18. *For the 2-class problem in which the two class conditional distributions are Uniform and identical as $U[0,1]^d$ and $U[h,1+h]^d$, the classifier*

$$x_1 + x_2 + ... + x_d \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \frac{d}{2}(h+1),$$

obtained by using symmetric pairs of the n-OS, i.e., the $n-k$ OS for ω_1 and k OS for ω_2, leads to an optimal Bayesian classification if and only if $k > \frac{(n+1)(1-h)}{2}$ for all the features. If this condition is violated, the CMOS classifier uses the Dual *condition, i.e., the k OS for ω_1 and the $n-k$ OS for ω_2 for both the features.*

Proof. The proofs for higher order multi-dimensional distributions follow due to the independence and due to the arguments analogous to those used in Theorems 16 and 17. However, on simplification, since the radical terms do not appear as cross products, the discriminant becomes:

$$\sum_{i=1}^{d} x_i \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \frac{d}{2}(h+1). \tag{6.7}$$

The details are omitted here in the interest of brevity. □

6.3 Doubly-Exponential Distribution

We earlier worked with the 2-class problem in which the class conditional distributions are uni-dimensional Doubly-Exponential and identical. We then demonstrated that

the optimal Bayesian classification can be achieved by using symmetric pairs of the n-OS, i.e., the $n-k$ OS for the first distribution and the k-OS for the second distribution if and only if $ln\left(\frac{2k}{n+1}\right) > \frac{c_1-c_2}{2}$ where c_1 and c_2 are the respective means of the distributions. Now, we shall extend this result for the multi-dimensional Doubly-Exponential distributions. For this, as in the case of the Uniform distribution, we first discuss two-dimensional distributions, and then generalize it for other higher dimensions.

6.3.1 Order Statistics: Doubly-Exponential Distributions

Let ω_1 and ω_2 be the two classes where the features follow two-dimensional Doubly-Exponential distributions. Then, since the random vectors have independent components[3], the pdfs can be represented as:

$$f_1(X) = \frac{\lambda_{11}}{2}e^{-\lambda_{11}|x_1-\mathbf{c_{11}}|} \cdot \frac{\lambda_{12}}{2}e^{-\lambda_{12}|x_2-\mathbf{c_{12}}|}, \quad -\infty < x_1 < \infty, -\infty < x_2 < \infty, \text{ and}$$

$$f_2(X) = \frac{\lambda_{21}}{2}e^{-\lambda_{21}|x_1-\mathbf{c_{21}}|} \cdot \frac{\lambda_{22}}{2}e^{-\lambda_{22}|x_2-\mathbf{c_{22}}|}, \quad -\infty < x_1 < \infty, -\infty < x_2 < \infty,$$

where $C_1 = (c_{11}, c_{12})$ and $C_2 = (c_{21}, c_{22})$ are the respective means of the distributions, and the values λ_{11}, λ_{12}, λ_{21} and λ_{22} are the corresponding parameters of the distributions in the respective dimensions.

In [51], we had derived the k-OS CMOS positions for the uni-dimensional Doubly-Exponential distribution as:

$$o_1 = c_1 - \frac{1}{\lambda_1} ln\left(\frac{2k}{n+1}\right), \tag{6.8}$$

and

$$o_2 = c_2 + \frac{1}{\lambda_2} ln\left(\frac{2k}{n+1}\right). \tag{6.9}$$

As the individual features of the Doubly-Exponential distribution are independent, the CMOS positions are computed directly using *these* independent univariate

[3]This independence is a consequence of the fact that the exponential terms can be factored so that each factor only possesses a *single* variable.

distributions, and thus have the corresponding forms as those of the positions obtained for the uni-dimensional distributions. Consequently, for the two dimensional distributions for classes ω_1 and ω_2, the CMOS positions O_1 and O_2 are:

$$O_1 = \left[c_{11} - \frac{1}{\lambda_{11}} ln\left(\frac{2k}{n+1}\right),\ c_{12} - \frac{1}{\lambda_{12}} ln\left(\frac{2k}{n+1}\right)\right]^T \tag{6.10}$$

and

$$O_2 = \left[c_{21} + \frac{1}{\lambda_{21}} ln\left(\frac{2k}{n+1}\right),\ c_{22} + \frac{1}{\lambda_{22}} ln\left(\frac{2k}{n+1}\right)\right]^T. \tag{6.11}$$

To proceed, we now need to show that the classification with respect to these CMOS positions attains the optimal Bayes' bound when the various distributions are identical and symmetrically placed.

6.3.2 Theoretical Analysis: Doubly-Exponential Distributions

In this section, we claim that CMOS can attain the optimal bound for two-dimensional identical and symmetrically placed Doubly-Exponential distributions. Without loss of generality, we consider the distributions to have the means $(0,0)$ and (c,c) respectively and with identical λ ($\lambda_{11} = \lambda_{12} = \lambda_{21} = \lambda_{22} = \lambda$). Then the distributions are:

$$f_1(X) = \frac{\lambda}{2}e^{-\lambda x_1} \cdot \frac{\lambda}{2}e^{-\lambda x_2}, \quad -\infty < x_1 < \infty, -\infty < x_2 < \infty, \text{ and}$$

$$f_2(X) = \frac{\lambda}{2}e^{-\lambda|x_1-c|} \cdot \frac{\lambda}{2}e^{-\lambda|x_2-c|}, \quad -\infty < x_1 < \infty, -\infty < x_2 < \infty.$$

We shall now formally prove the claim.

Theorem 19. *For the 2-class problem in which the two class conditional distributions are two-dimensional Doubly-Exponential, identical and symmetric, optimal Bayesian classification can be achieved by using symmetric pairs of the n-OS, i.e., the $n-k$ OS for ω_1 and the k OS for ω_2 if and only if $ln\left(\frac{2k}{n+1}\right) > \frac{C_1-C_2}{2}$ for both the features. If this condition is violated, the CMOS classifier uses the* Dual *condition, i.e., the k OS for ω_1 and the $n-k$ OS for ω_2 for both the features.*

Proof. We first invoke Theorem 3 of [51], where we have shown that when the class conditional distributions are Doubly-Exponential and identical, the optimal Bayes'

bound can be attained by a comparison to the corresponding *means*. Similarly, for two-dimensional distributions, we can see that the result is valid since[4]:

$$\begin{aligned}
& p(X|\omega_1)\ P(\omega_1) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} p(X|\omega_2)\ P(\omega_2) \\
\Longrightarrow\quad & \frac{\lambda}{2}\, e^{-\lambda|x_1-c_{11}|} \cdot \frac{\lambda}{2}\, e^{-\lambda|x_2-c_{12}|} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{\lambda}{2}\, e^{-\lambda|x_1-c_{21}|} \cdot \frac{\lambda}{2}\, e^{-\lambda|x_2-c_{22}|} \\
\Longrightarrow\quad & x_1 + x_2 - c_{21} - c_{22} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} -x_1 - x_2 + c_{11} + c_{12} \\
\Longrightarrow\quad & x_1 + x_2 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \frac{c_{11} + c_{12} + c_{21} + c_{22}}{2}.
\end{aligned} \tag{6.12}$$

This inequality reduces to $x_1 + x_2 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} c$ for the distributions with means $(0,0)$ and (c,c) respectively.

What remains to be proven is that the CMOS can attain the optimal Bayes' bound which is, in turn, equivalent to the accuracy obtained by performing comparisons with the corresponding means. Hence, the claim can be written as:

$$D(X,C_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(X,C_2) \Longrightarrow D(X,O_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(X,O_2). \tag{6.13}$$

The LHS of this claim simplifies to:

$$\begin{aligned}
D(X,C_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(X,C_2) \quad \Longrightarrow \quad & \sqrt{x_1^2 + x_2^2} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \sqrt{(x_1-c)^2 + (x_2-c)^2} \\
\Longrightarrow \quad & 2c(x_1 + x_2) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} 2c^2. \\
\Longrightarrow \quad & x_1 + x_2 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} c.
\end{aligned} \tag{6.14}$$

In order to prove our claim, we need to show that the RHS of the claim also

[4]The argument could have just as well been shown if we worked with vectorial expressions. But we have chosen to work with the corresponding scalar expressions so that the generalization is obvious. Also, the expressions will not be so straightforward if the λ's are different in each dimension.

reduces to the same inequality. This is true since:

$$
\begin{aligned}
& D(X,O_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(X,O_2) \\
\Longrightarrow \quad & \sqrt{\left(x_1 + ln\left(\frac{2k}{n+1}\right)\right)^2 + \left(x_2 + ln\left(\frac{2k}{n+1}\right)\right)^2} \\
\underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \quad & \sqrt{\left(\left(c + ln\left(\frac{2k}{n+1}\right)\right) - x_1\right)^2 + \left(\left(c + ln\left(\frac{2k}{n+1}\right)\right) - x_2\right)^2} \\
\Longrightarrow \quad & (x_1 + x_2)\left(4ln\left(\frac{2k}{n+1}\right) + 2c\right) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} 2c^2 + 4cln\left(\frac{2k}{n+1}\right) \\
\Longrightarrow \quad & 2(x_1 + x_2)\left(c + 2ln\left(\frac{2k}{n+1}\right)\right) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} 2c\left(c + 2ln\left(\frac{2k}{n+1}\right)\right) \\
\Longrightarrow \quad & x_1 + x_2 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} c. \qquad (6.15)
\end{aligned}
$$

The proof follows from the identical inequalities obtained in Eqs. (6.14) and (6.15) for the LHS and the RHS of the claim given in Eq. (6.13), which is again valid because the higher order and radical terms do not appear as cross products. Hence the theorem. □

6.3.3 Experimental Results: Doubly-Exponential Distributions

The CMOS classifier has been rigorously tested for a number of experiments with various Doubly-Exponential distributions having means C_1 and C_2. In every case where the condition $ln\left(\frac{2k}{n+1}\right) > \frac{C_1 - C_2}{2}$ was satisfied, the k-OS CMOS gave exactly the same classification as that of the Bayesian classifier. The method was executed 50 times with the 10-fold cross validation scheme. The test results are depicted in Table 6.3. The mean of the first distribution was the origin and the mean of the second distribution was $\langle c, c \rangle$. As the reader can see, CMOS attained the Bayes' bound for all the cases where the condition specified in Theorem 19 is enforced.

Now, consider the results presented in the row denoted by Trial No. 4. In this case, the testing attained the Bayes' accuracy for the symmetric OS pairs $\langle \frac{2}{3}, \frac{1}{3} \rangle$ and $\langle \frac{5}{7}, \frac{2}{7} \rangle$, but the Dual pairs had to be used for the pairs $\langle \frac{4}{5}, \frac{1}{5} \rangle$ and $\langle \frac{8}{9}, \frac{1}{9} \rangle$, since these values violated the condition imposed by Theorem 19.

No.	c	w.r.t Mean	$\langle \frac{2}{3}, \frac{1}{3} \rangle$	$\langle \frac{4}{5}, \frac{1}{5} \rangle$	$\langle \frac{5}{7}, \frac{2}{7} \rangle$	$\langle \frac{8}{9}, \frac{1}{9} \rangle$
1	3	96.55	96.55	96.55	96.55	96.55
2	2.5	95.5	95.5	95.5	95.5	95.5
3	2	92	92	92	92	92
4	1.5	89.3	89.3	89.3 (D)	89.3	89.3 (D)

Table 6.3: Classification of Doubly-Exponentially distributed 2-dimensional classes by the CMOS k-OS method for different means. The results reported are just a small subset of the results obtained. The rest are omitted in the interest of brevity. The scenarios when we have invoked the *Dual* condition are specified by noting them using the notation "(D)".

6.3.4 Multi-Dimensional Extension: Doubly-Exponential Distributions

The results of Theorem 19 proved that the k-OS CMOS can attain the optimal Bayes' bound for 2-dimensional Doubly-Exponential distributions. This result can easily be extended for multi-dimensional distributions. As the features are again independent because of the explicit factorizability, we can perform the classification with respect to the $\langle \frac{n+1-k}{n+1}, \frac{k}{n+1} \rangle$ positions of each of the features for the given distributions whenever the two distributions are identical and symmetrical.

Theorem 20. *For the 2-class problem in which the two class conditional distributions are d-dimensional Doubly-Exponential, identical and symmetric, the optimal Bayesian classifier is:*

$$x_1 + x_2 + ... + x_d = \frac{d}{2} \cdot c,$$

and is exactly the CMOS classifier obtained by using symmetric pairs of the n-OS, i.e., the $n-k$ OS for ω_1 and the k OS for ω_2, if and only if $ln\left(\frac{2k}{n+1}\right) > \frac{C_1 - C_2}{2}$ for all the features. If this condition is violated, the CMOS classifier uses the Dual *condition, i.e., the k OS for ω_1 and the $n-k$ OS for ω_2 for all the features.*

Proof. The proof can be achieved by a straightforward extension of the arguments of Theorem 19. One can see that the simplifications for both the classifiers result to:

$$\sum_{i=1}^{d} x_i \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \frac{d}{2} \cdot c, \tag{6.16}$$

where the means of the distributions are $C_1 = [0, 0, ..., 0]^T$ and $C_2 = [c, c, ..., c]^T$. The details of the proof are omitted here to avoid repetition. Hence the theorem. □

6.4 Gaussian Distribution

In this section, we intend to work with multi-dimensional Gaussian distribution. Earlier in [51], we showed that CMOS can attain optimal classification for uni-dimensional Gaussian distribution. The moments of the OS for the Normal distribution can be determined from the generalized expression:

$$E[\mathbf{x}_{k,n}^r] = \frac{n!}{(k-1)!(n-k)!} \int_{-\infty}^{+\infty} x^r \Phi^{k-1}(x)(1-\Phi(x))^{n-k}\varphi(x)dx,$$

where $\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$ and $\Phi(x) = \int_{-\infty}^{x} \varphi(t)dt$, and when we are considering the k^{th}-OS from n samples. From this expression, the expected values of the first moment of the 2-OS can be determined as:

$$E[\mathbf{x}_{1,2}] = \mu - \frac{\sigma}{\sqrt{2\pi}}, \quad \text{and} \tag{6.17}$$

$$E[\mathbf{x}_{2,2}] = \mu + \frac{\sigma}{\sqrt{2\pi}}, \tag{6.18}$$

as shown in [1]. As before, we initially deal with the two-dimensional Gaussian distribution, and then extend the result for higher dimensions.

6.4.1 Order Statistics: Gaussian Distributions

Let ω_1 and ω_2 be the two classes where the features follow two-dimensional Gaussian distributions. As the pdfs of the Gaussian distributions are not factorizable, we need

to assume the independence (i.e., uncorrelation) of the features. Then, the pdfs can be represented as:

$$f_1(X) = \frac{1}{\sqrt{2\pi}\sigma_{11}} e^{\frac{-(x_1-\mu_{11})^2}{2\sigma_{11}^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma_{12}} e^{\frac{-(x_2-\mu_{12})^2}{2\sigma_{12}^2}}, \quad -\infty < x_1 < \infty, -\infty < x_2 < \infty, \text{ and}$$

$$f_2(X) = \frac{1}{\sqrt{2\pi}\sigma_{21}} e^{\frac{-(x_1-\mu_{21})^2}{2\sigma_{21}^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma_{22}} e^{\frac{-(x_2-\mu_{22})^2}{2\sigma_{22}^2}}, \quad -\infty < x_1 < \infty, -\infty < x_2 < \infty,$$

where $M_1 = (\mu_{11}, \mu_{12})$ and $M_2 = (\mu_{21}, \mu_{22})$ are the respective means and $\sigma_1 = (\sigma_{11}, \sigma_{12})$ and $\sigma_2 = (\sigma_{21}, \sigma_{22})$ are the corresponding standard deviations of the distributions.

As the individual features of the Gaussian distribution are independent, the CMOS positions are computed directly using *these* independent univariate distributions, and thus have the same forms as those of the positions obtained for the uni-dimensional distributions. Consequently, for the two dimensional distributions for the classes ω_1 and ω_2, the CMOS positions O_1 and O_2 are:

$$O_1 = \left[\mu_{11} - \frac{\sigma_{11}}{\sqrt{2\pi}}, \ \mu_{12} - \frac{\sigma_{12}}{\sqrt{2\pi}}\right]^T \tag{6.19}$$

and

$$O_2 = \left[\mu_{21} - \frac{\sigma_{21}}{\sqrt{2\pi}}, \ \mu_{22} - \frac{\sigma_{22}}{\sqrt{2\pi}}\right]^T. \tag{6.20}$$

To proceed, we now need to show that the classification with respect to these CMOS positions attains the optimal Bayes' bound when the various distributions are identical and symmetrically placed.

6.4.2 Theoretical Analysis: Gaussian Distributions

In this section, we demonstrate that CMOS can attain the optimal bound for two-dimensional identical and symmetrically placed Gaussian distributions. Without loss of generality, we consider the distributions to have the means $(0, 0)$ and (μ, μ) respectively, and with identical standard deviations, σ ($\sigma_{11} = \sigma_{12} = \sigma_{21} = \sigma_{22} = \sigma$). Then the distributions are:

$$f_1(X) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x_1^2}{2\sigma^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x_2^2}{2\sigma^2}}, \quad -\infty < x_1 < \infty, -\infty < x_2 < \infty, \text{ and}$$

$$f_2(X) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x_1-\mu)^2}{2\sigma^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x_2-\mu)^2}{2\sigma^2}}, \quad -\infty < x_1 < \infty, -\infty < x_2 < \infty.$$

The claim we stated above is now proven.

Theorem 21. *For the 2-class problem in which the two 2-dimensional class conditional distributions are Gaussian, identical and symmetric, CMOS, the classification using two OS, attains the optimal Bayes' bound.*

Proof. Earlier, in [51], we had shown that when the class conditional distributions are Gaussian and identical, the optimal Bayes' bound can be attained by a comparison to the corresponding *means*. So, for two-dimensional distributions also, we can see that the result is valid since:

$$\begin{aligned}
& p(X|\omega_1)\ P(\omega_1) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} p(X|\omega_2)\ P(\omega_2) \\
\Longrightarrow \quad & \frac{1}{\sqrt{2\pi}\sigma_{11}} e^{\frac{-(x_1-\mu_{11})^2}{2\sigma_{11}^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma_{12}} e^{\frac{-(x_2-\mu_{12})^2}{2\sigma_{12}^2}} \\
\underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \quad & \frac{1}{\sqrt{2\pi}\sigma_{21}} e^{\frac{-(x_1-\mu_{21})^2}{2\sigma_{21}^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma_{22}} e^{\frac{-(x_2-\mu_{22})^2}{2\sigma_{22}^2}} \\
\Longrightarrow \quad & x_1^2 + x_2^2 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} (x_1-\mu)^2 + (x_2-\mu)^2 \\
\Longrightarrow \quad & x_1 + x_2 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \mu,
\end{aligned} \tag{6.21}$$

where $(0,0)$ and (μ,μ) are the means of the distributions.

We shall now prove that the CMOS can attain the optimal Bayes' bound, which is again equivalent to the accuracy obtained by comparing with the corresponding means. This claim can be written as:

$$D(X,M_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(X,M_2) \Longrightarrow D(X,O_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(X,O_2). \tag{6.22}$$

The LHS of this claim simplifies to:

$$\begin{aligned}
D(X,M_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(X,M_2) \quad \Longrightarrow \quad & \sqrt{x_1^2+x_2^2} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \sqrt{(x_1-\mu)^2+(x_2-\mu)^2} \\
\Longrightarrow \quad & 2\mu(x_1+x_2) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} 2\mu^2. \\
\Longrightarrow \quad & x_1+x_2 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \mu.
\end{aligned} \tag{6.23}$$

In order to prove our claim, we need to show that the RHS of the claim also reduces to the same inequality. This is true since:

$$
\begin{aligned}
& D(X, O_1) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} D(X, O_2) \\
\Longrightarrow \quad & \sqrt{\left(x_1 + \frac{\sigma}{\sqrt{2\pi}}\right)^2 + \left(x_2 + \frac{\sigma}{\sqrt{2\pi}}\right)^2} \\
\underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \quad & \sqrt{\left(\mu + \frac{\sigma}{\sqrt{2\pi}} - x_1\right)^2 + \left(\mu + \frac{\sigma}{\sqrt{2\pi}} - x_2\right)^2} \\
\Longrightarrow \quad & 4(x_1 + x_2)\left(\frac{\sigma}{\sqrt{2\pi}}\right) + 2\mu(x_1 + x_2) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} 2\mu^2 + 4\mu\left(\frac{\sigma}{\sqrt{2\pi}}\right) \\
\Longrightarrow \quad & 2(x_1 + x_2)\left(\mu + \frac{2\sigma}{\sqrt{2\pi}}\right) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} 2\mu\left(\mu + \frac{2\sigma}{\sqrt{2\pi}}\right) \\
\Longrightarrow \quad & x_1 + x_2 \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} \mu. \qquad (6.24)
\end{aligned}
$$

The proof follows from the identical inequalities obtained in Eqs. (6.23) and (6.24) for the LHS and the RHS of the claim given in Eq. (6.22). Hence the theorem[5]. □

6.4.3 Experimental Results: Gaussian Distributions

The CMOS method for 2-dimensional Gaussian distributions ω_1 (centered at $(0, 0)$) and ω_2 (centered at (μ, μ)) has been rigorously tested, and the results are given in Table 6.4. For each of the experiments, we generated 1,000 points for the classes ω_1 and ω_2. The method was executed 50 times with the 10-fold cross validation scheme. In every case, the 2-OS CMOS gave exactly the same classification as that of the Bayesian classifier.

6.4.4 Multi-dimensional Extension: Gaussian Distributions

The result that we obtained in Theorem 21 can easily be generalized for higher dimensions. As the features are assumed to be independent with identical variances in

[5]The amazing reason for this claim is that the radicals and exponents again appear independently and not as cross product terms.

μ	1	1.5	2	2.5	3	3.5	4	4.5
Bayesian	75.985	85.485	91.93	96.13	98.335	99.34	99.81	99.95
CMOS	75.985	85.485	91.93	96.13	98.335	99.34	99.81	99.95

Table 6.4: Classification of 2-dimensional Gaussian Distributions by the CMOS 2-OS method for different means $(0,0)$ and (μ,μ). The accuracy obtained in every case is exactly the Bayes' bound.

each dimension, the classification can be performed with regard to the 2-OS CMOS positions of each of the features for the identical and symmetrical distributions. The result that is true is stated below.

Theorem 22. *For the 2-class problem in which the two class conditional distributions are d-dimensional Gaussian, identical and symmetric, the optimal Bayesian classifier has the form:*

$$x_1 + x_2 + ... + x_d = \tfrac{d}{2} \cdot \mu,$$

and this is again the classifier obtained by using symmetric 2-OS CMOS positions.

Proof. We have earlier seen that the 2-OS CMOS positions for identical and symmetric Gaussian distributions are $o_1 = \mu_1 - \frac{\sigma}{\sqrt{2\pi}}$ and $o_2 = \mu_2 + \frac{\sigma}{\sqrt{2\pi}}$. For the multi-dimensional classification problem, as the features are independent, the task can be performed based on the 2-OS CMOS positions for all of the features. This will result in the classifier:

$$\sum_{i=1}^{d} x_i = \frac{d}{2} \cdot \mu. \tag{6.25}$$

The proof is straightforward and the algebra is omitted here to avoid repetition. □

6.5 Rayleigh Distribution

In [38], we had earlier worked with uni-dimensional Rayleigh distributions in which CMOS attained near-optimal classification with regard to the classifiers based on the

medians of the distribution. We also showed that the error difference created by the CMOS classifier when compared to the Bayesian classifier is negligible by considering the differences of the error probabilities quantified by the differences between the areas under the curves of the resulting errors. In this section, we work with the two-dimensional Rayleigh distribution and intend to show that the CMOS can attain near-optimal classification by performing the task based on the moments of the OS.

6.5.1 Order Statistics: Rayleigh Distributions

Let ω_1 and ω_2 be the two classes where the features follow two-dimensional Rayleigh distributions. Earlier, in [51], we derived the 2-OS CMOS positions for the uni-dimensional Rayleigh distribution as:

$$o_1 = \sigma\sqrt{2\ ln(3)}, \tag{6.26}$$

and

$$o_2 = \theta + \sigma\sqrt{2ln\left(\frac{3}{2}\right)}. \tag{6.27}$$

In order to extend this result for a two-dimensional case, as before, we assume a Naïve-Bayes' approach, in which the first moments of the OS in each of the dimensions are uncorrelated. Then, the pdfs can be represented as:

$$f(X,\sigma) = \frac{x_1}{\sigma_{11}^2}e^{\frac{-x_1^2}{2\sigma_{11}^2}} \cdot \frac{x_2}{\sigma_{12}^2}e^{\frac{-x_2^2}{2\sigma_{12}^2}}, \quad -\infty < x_1 < \infty, -\infty < x_2 < \infty, \text{ and}$$

$$f(X-\theta,\sigma) = \frac{x_1-\theta_1}{\sigma_{21}^2}e^{\frac{-(x_1-\theta_1)^2}{2\sigma_{21}^2}} \cdot \frac{x_2-\theta_2}{\sigma_{22}^2}e^{\frac{-(x_2-\theta_2)^2}{2\sigma_{22}^2}}, \quad -\infty < x_1 < \infty, -\infty < x_2 < \infty,$$

where $\sigma_1 = (\sigma_{11}, \sigma_{12})$ and $\sigma_2 = (\sigma_{21}, \sigma_{22})$ are the corresponding standard deviations of the distributions. Consequently, for the two dimensional distributions for the classes ω_1 and ω_2, the CMOS positions O_1 and O_2 are respectively:

$$O_1 = \left[\sigma_{11}\sqrt{2\ ln(3)},\ \sigma_{12}\sqrt{2\ ln(3)}\right]^T \tag{6.28}$$

and

$$O_2 = \left[\theta_1 + \sigma_{21}\sqrt{2ln\left(\frac{3}{2}\right)},\ \theta_2 + \sigma_{22}\sqrt{2ln\left(\frac{3}{2}\right)}\right]^T. \tag{6.29}$$

6.5.2 Theoretical Analysis: Rayleigh Distributions

In [38], we showed that CMOS can attain a near-optimal classification for Rayleigh distributed distributions. In order to prove this claim, we showed that the error difference created by the CMOS classifier when compared to the Bayesian classifier is negligible by considering the differences of the error probabilities quantified by the differences between the areas under the curves of the resulting errors. As in the uni-dimensional case, we can now compute the differences in the corresponding analogous *volumes* created by the classifiers for the respective distributions. The "ceiling" of the volume is rather complex because it involves the difference between the corresponding three-dimensional surfaces. However, we can easily obtain an upper bound for this volume by considering the smallest bounding rectanguloid. In Figure 6.2, we display this upper bound of the error for the CMOS classifier when compared to the Bayesian classifier, which can be seen to be the product of the differences in the positions of the CMOS classifier in both the dimensions, and the height difference of the distributions at the corresponding positions. This volume is the small yellow rectanguloid marked in the figure.

The upper bound of the error can be numerically evaluated and is given in Table 6.5. Notice that this bound is minimized for various values of k for the CMOS. In each case, the value of k which leads to the minimum error is also listed.

θ	x : 4-**OS**, $k = 2$	x : 6-**OS**, $k = 3$	x : 8-**OS**, $k = 4$
1	0	0	0
1.5	0	0	0
2	0	0	0
2.5	0	0	0
3	0	0	0

Table 6.5: Upper bound of the error (in %) by the CMOS classifier when compared to the classifier with respect to the medians of the distributions, for different values of θ, k and n of the 2-dimensional Rayleigh Distribution. In each case, $\sigma = 2$.

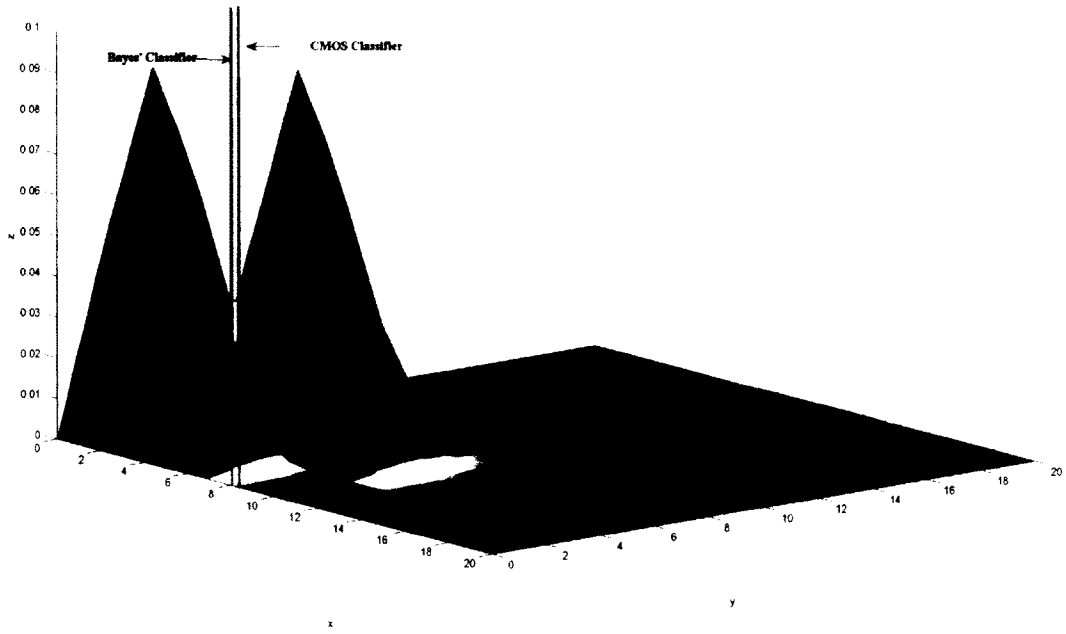


Figure 6.2: The upper bound of the differences of the error probabilities quantified by the differences between the areas under the curves of the resulting errors.

From this table, we can conclude that the error probability quantified by the differences between the volumes under the curves of the resulting errors is negligible and this justifies our claim that CMOS can attain a near-optimal bound. This claim is experimentally proven in Section 6.5.3.

6.5.3 Experimental Results: Rayleigh Distributions

The CMOS method has been rigorously tested with different possibilities of the k-OS and for various values of n, and the test results are given in Table 6.6. For each of the experiments, we generated 1,000 points for the classes ω_1 and ω_2. The method was executed 50 times with the 10-fold cross validation scheme.

No.	Order(n)	Moments	$\theta = 2$	$\theta = 1.5$	$\theta = 1.3$	$\theta = 1.2$
1	Median	$[(\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{1}{2})]$	98.25	95.3	93.15	89.75
2	Two	$[(\frac{2}{3},\frac{1}{3}),(\frac{2}{3},\frac{1}{3})]$	98.35	95.3	92.95	89.6
3	Four	$[(\frac{4}{5},\frac{1}{5}),(\frac{4}{5},\frac{1}{5})]$	98.65	95.3	92.7	90.6
4	Four	$[(\frac{3}{5},\frac{2}{5}),(\frac{3}{5},\frac{2}{5})]$	98.25	95.2	93.15	89.6
5	Six	$[(\frac{6}{7},\frac{1}{7}),(\frac{6}{7},\frac{1}{7})]$	98.75	95.15	92.2 (D)	90.35 (D)
6	Six	$[(\frac{5}{7},\frac{2}{7}),(\frac{5}{7},\frac{2}{7})]$	98.45	95.2	92.85	90.05
7	Six	$[(\frac{4}{7},\frac{3}{7}),(\frac{4}{7},\frac{3}{7})]$	98.25	95.2	93.15	89.75
8	Eight	$[(\frac{8}{9},\frac{1}{9}),(\frac{8}{9},\frac{1}{9})]$	98.8	95.2 (D)	92.25 (D)	89.9 (D)
9	Eight	$[(\frac{7}{9},\frac{2}{9}),(\frac{7}{9},\frac{2}{9})]$	98.65	95.3	92.85	90.45
10	Eight	$[(\frac{5}{9},\frac{4}{9}),(\frac{5}{9},\frac{4}{9})]$	98.25	95.25	93.1	89.7

Table 6.6: Classification of Rayleigh distributed 2-dimensional classes by the CMOS k-OS method for different values of θ. The scenarios when we have invoked the *Dual* condition are specified by noting them using the notation "(D)".

6.6 Other Multi-dimensional Distributions

In Sections 6.2 - 6.5, we dealt with some of the symmetric multi-dimensional distributions of the exponential family such as the Uniform, Doubly-Exponential and the Gaussian distribution and a specific asymmetric two-dimensional distribution, i.e., the Rayleigh distribution. We have proved that the CMOS can attain the optimal Bayes' bound for all the symmetric distributions and a near-optimal bound for the asymmetric distribution. One can see that the strategy is analogous to a Naïve-Bayes' approach, although it, really, is of an *anti*-Naïve-Bayes' paradigm. For the distributions in which the class-conditional densities are not factorizable, it is important to assume that the features are uncorrelated.

One can see that the same argument can be generalized for any higher-dimensional distribution to attain optimal/near-optimal classification. This was mentioned in Sections 6.2 - 6.4, but we omit further discussion here because the strategy is identical.

6.7 Conclusions

The results of this chapter have built on the works of Chapters 4 and 5, in which we proposed a new paradigm named CMOS, Classification by the Moments of Order Statistics, which specifically utilized the properties of the *Order Statistics* (OS) quantifiers of the distributions of the features in a PR problem.

Unlike our initial works, in which we proposed the foundational theory of CMOS for the uni-dimensional Uniform distribution (Chapter 3), and which were subsequently extended for various symmetric and asymmetric uni-dimensional distributions within the exponential family, in this chapter, we generalized these results for *multi-dimensional* distributions. Such a generalization is particularly interesting and non-trivial because there is no well-established method for achieving the ordering of multi-dimensional data specified in terms of its uni-dimensional components. The strategy that we have proposed can be seen to be analogous to a Naïve-Bayes' approach, although it really is of an *anti*-Naïve-Bayes' paradigm. The multi-dimensional analytical and experimental generalizations were done for various two-dimensional distributions, and the way by which they can be extended for higher dimensions was formally submitted. As far as we know, the results for classification using the OS for such a multi-dimensional scenario are both pioneering and novel.

In the next chapter, we will extend these results for real-life data, and also investigating how they can be used to develop new Border Identification algorithms. The formal work actually leads us to what we shall refer to as the "ultimate" PRS because it reduces the size of the "Reference" set to only a single element from each class.

Chapter 7

"Ultimate" Prototype Reduction Schemes Obtained using CMOS

7.1 Introduction

In this chapter, we will consider the task of extending CMOS to resolve the problem of designing new PRSs and hopefully, the "ultimate" PRSs. As explained in Chapter 2, a PRS is a generic method for reducing the number of training vectors, without affecting the performance of the classifier built on the reduced design set [17, 21, 28, 54]. Instead of considering all the training patterns for the classification, a subset of the whole set is selected based on certain criteria. The training is then performed on this reduced set, which is also called the "Reference" set. More recent advances have involved the use of Border Identification (BI) algorithms [8, 13, 14, 33] to choose these prototypes from the so-called "border" points of the various classes.

Traditionally, a good PRS can reduce the size of the training set to a small percentage (for example, 10%) of the original set. But how small can one make this reduced set? Is it possible to, at least conceptually, reduce the set of prototypes to contain *only a single element* from each class. The aim of this chapter is to investigate this issue both conceptually and from a practical perspective. Indeed, we shall demonstrate that we can push and attain the limit on the field of PRSs to obtain a classification accuracy comparable to the optimal, by condensing the information in

the data set into a *single training* point. Apart from showing that such a PRS exists and is attainable, we shall also show that the design and implementation of such a mechanism relies on the recently-introduced paradigm of Order Statistics (OS)-based classifiers.

One should, of course, mention that the new point obtained by invoking the PRS is not necessarily a member of the original data set. Rather, it can be an artificially created point, representative of the training set, as perceived from the perspective of the data sets OSs.

We now consider another facet of a typical PRS-based PR solution which we did not consider in Chapter 2. Whenever a practitioner designs a PRS, he works with the premise that *all* features are crucial for the classification. The problem that is "dual" to the PRS problem is the following: Apart from reducing the size of the "Reference" set, is it possible to also reduce the number of features utilized within the latter. This chapter addresses both of these issues simultaneously. To be specific, we state that the OS-based PRS scheme that we propose has the fascinating property that it can be rendered operational by using the information in a *single feature* of the *single data point* obtained using an OS-based computation. Indeed, in each of these cases, the accuracy of this approach is very close to the optimal Bayes' bound and is almost comparable to that of the SVM. In a nutshell, this is the fundamental contribution of this chapter, and we are not aware of any reported comparable results.

From an overall perspective, we now discuss how we are to achieve our goal to reduce the cardinality of the OS-based PRS to be unity for each class. First of all, we know that PRSs can be broadly classified as being "selective" or "creative" [26]. A "selective" PRS yields as its output a set of prototypes which are *chosen* from the original training points. As opposed to this, a "creative" PRS *creates* a set of artificial points which may not be found in the original training set, and these points are thereafter used in the classification. This chapter considers both these strategies.

We first study the task of designing "selective" OS-based PRSs in Section 7.3. Since, at this juncture, we are not willing to assume a distributional form for the

corresponding features, we are forced to work with the non-parametric representation that the training data captures. By working with the multi-dimensional non-parametric form of the data, and by thereafter invoking an OS-based paradigm, we are able to obtain a *single* prototype with which we can accomplish efficient classification. This *single* prototype is, as a vector, a "created" point, although, in every single dimension, the value is "selected" from the actual training sample that is closest to the value specified by the OS value.

Two versions of this strategy have been proposed, namely, the first which considers the entire vectorial form of the resultant prototype (in Section 7.3.1), and the second which invokes a majority vote by considering the OS-based classification of the individual features. The latter, which is a *Scalar*-Based Selective PRS, has been described in Section 7.3.2. It is worth mentioning that the classification results obtained by both these methods – both of which involve only a *single* prototype – are quite satisfactory, and are comparable, though understandably, marginally inferior, to those obtained from a Naïve Bayes or SVM strategy.

After investigating selective PRSs, we subsequently consider the task of designing "creative" OS-based PRSs in Section 7.4. In this case, we assume a distributional form for the corresponding features, and so we proceed to work with the parametric representation that the training data captures. By working with a multi-dimensional parametric form of the data, and by thereafter invoking an OS-based paradigm, we succeed in obtaining a *single* prototype in the "Reference" set, which can be used for classification. This process has been explained in Section 7.4.1. As in the non-parametric case, we have also developed a *Scalar*-Based Creative PRS in Section 7.4.2. Again, it is worth mentioning that the classification results obtained from both these parametric strategies (i.e., the vector, and the majority-voted individual-feature based) are quite satisfactory, and comparable, though marginally inferior, to those obtained from a Naïve Bayes or SVM strategy.

The final concluding contribution is actually far more ambitious. It consists of using only a *single* feature of a *single* prototype. In this case, in Section 7.5, we have designed a "creative" PRS scheme which merely includes the OS-based points of a single feature, where the $\frac{n-k+1}{n+1}^{th}$ percentile of *this* feature of the first class, and the

$\frac{k}{n+1}^{th}$ percentile of *this* feature of the second class, are the corresponding "prototypes". It is clear that the accuracy of this *scalar*-based OS will be inferior to that of the corresponding *vector*-based OS. However, astonishingly enough, the accuracy does not degrade significantly – the resultant classifier still yields an accuracy that is acceptable considering the fact that one requires only a single *scalar* comparison to achieve the classification.

The reader must observe that the intent of this chapter is not to compare the resultant classification accuracies with those obtained from an entire ensemble of classification methodologies. Rather, our aim is to show that we can obtain very efficient classification by merely using a single (vector or scalar) prototype which is either selected or created. Thus, we have compared our proposed scheme with only *three* standard algorithms which have been universally considered as benchmarks. We believe that the results presented here conclusively demonstrate the power of our contribution.

7.1.1 Contributions of this Chapter

The novel contributions of this chapter are:

- We propose a "selective" PRS which can be metaphorically perceived to be the "Ultimate" selective PRS because, by using a non-parametric paradigm, it reduces the size of the "Reference" set to be a *single* pattern from each class, which is thereafter utilized in the classification;

- We also propose a "creative" PRS which can be considered to be the "Ultimate" creative PRS because, by invoking a parametric paradigm, it also reduces the size of the "Reference" set to be a *single* pattern from each class;

- In both of the above cases, we have also shown that it is possible to derive a majority-based PRS which fuses the classification results of the various features of the *single* d-dimensional prototype. The classification accuracies of these fused scalar schemes are marginally worse than those of the corresponding vector-based algorithms;

- We have also shown that it is possible to derive a single scalar prototype, i.e., one which involves only a a *single* feature of a *single* d-dimensional vector. The classification accuracy of this single-scalar PRS is marginally worse than that of the vector-based methods;

- In every case, we demonstrate, by testing the algorithms on real-life data sets from the UCI repository, that the new PRS-based classification schemes yield accuracies comparable to the traditional NB classifiers, and even the SVM, even though the computations needed are, really, of an atomic magnitude.

We conclude this section by remarking that, to the best of our knowledge, analogous results have been unreported in the literature.

7.2 Experimental Data Sets

7.2.1 Artificial Data Sets

For a *prima facie* testing of artificial data, we generated two classes that obeyed Gaussian distributions. To do this, we made use of a Uniform $(0, 1)$ random variable generator to generate data values that follow a Gaussian distribution. The expression $\mathbf{z} = \sqrt{-2ln(u_1)}\ cos(2\pi u_2)$ is known to yield data values that follow $N(0, 1)$ [7]. Thereafter, by using the technique described in [16], one can generate Gaussian random vectors which possess any arbitrary mean and covariance matrix. The means of the classes were $[2\ \ 2\ \ 2\ \ 2\ \ 2]^T$ and $[-2\ \ -2\ \ -2\ \ -2\ \ -2]^T$ respectively, and the covariances of the two classes were identical and had the form[1]:

$$\Sigma = \begin{bmatrix} a^2 & b & 0 & a & \alpha ab \\ b & 2a+3b & 0 & b & a \\ 0 & 0 & 1 & 0 & 0 \\ a & b & 0 & 2a+3b & b \\ \alpha ab & a & 0 & b & b^2 \end{bmatrix}$$

[1]In our experiments, we set $a = 5$, $b = 4$, and $\alpha = 0.4$.

This rendered the classes to have an optimal linear classifier. With regard to the cardinality of the data set, each of the classes had 200 instances in the corresponding 5-dimensional space.

7.2.2 Real-Life Setup

The data sets [15] used in this study have two classes, and the number of attributes varies from four up to thirty two. The data sets are described in Table 7.1.

Data set	No. Instances	No. Attributes	No. Classes	Attribute Type
WOBC	699	9	2	Integer
WDBC	569	32	2	Real
Diabetes	768	8	2	Integer, Real
Hepatitis	155	19	2	Categorical, Integer, Real
Iris	150	4	3	Real
Statlog (Heart)	270	13	2	Categorical, Real
Statlog (Australian Credit)	690	14	2	Categorical, Integer, Real
Vote	435	16	2	Categorical, Integer

Table 7.1: The Real-life data sets used in our experiments.

The **Wisconsin Original Breast Cancer (WOBC)** data set contains information regarding breast cancer. The goal is the prediction of benign and malignant tumors, which leads to a binary classification problem. This database was obtained from the University of Wisconsin Hospitals in Madison, from Dr. William H. Wolberg [35]. The database has some missing values, and they are handled by the ReplaceMissingValues filter of Weka 3.6.4 (a data mining software developed by the University of Waikato).
The **Wisconsin Diagnostic Breast Cancer (WDBC)** Data Set contains information regarding breast cancer. The PR goal is the prediction of benign and malign tumors (again leading to a binary classification problem). The features are computed from digitized images of a Fine Needle Aspirate of a breast mass, which describe characteristics of the cell nuclei present in the image [15]. The original attribute *ID number* was ignored in our experiments.
The **Pima Indians Diabetes** data set was donated by the National Institute of

Diabetes and Digestive and Kidney Diseases in 1990. The data set contains 768 instances, of females of at least 21 years old of Pima Indian heritage.

The **Hepatitis** data set contains information of 155 hepatitis patients and consists of the counts of bilirubin, phosphate, albumin, etc. The two classes are denoted by "DIE" and "LIVE" which represents the survivors and the patients for whom the Hepatitis proved terminal.

The **Iris** data set [12] includes the measurements (in centimeters) of the features, namely, the sepal length, sepal width, petal length and petal width, respectively, for 50 flowers from each of the 3 species of the Iris family. The species are the Iris Setosa, Iris Versicolor, and Iris Virginica. Iris Setosa is linearly separable from the other two classes. As we are discussing 2-class problems here, we consider the classification of the other two classes for the comparison of the classifiers.

The **Statlog (Australian Credit)** data set has 690 instances which are the details of the credit card applications.

The **Statlog (Heart)** data set is a heart disease database with 270 instances and has measurements like blood pressure, cholesterol, blood sugar, type of chest pain etc. The objective of the analysis is to determine the presence or absence of heart disease.

The **Congressional Voting Records** data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by Congressional Quarterly Almanac (CQA). There are two classes for this data set – Democrat and Republican. The data set has 435 instances with 16 attributes.

7.3 OS-based "Selective" PRSs Using a Non-parametric Perspective

In this section, we discuss the problem of designing a "Selective" OS-based PRS. Since we are ultimately going to select a training sample, at this juncture we take the position that we are not willing to assume a *distributional form* for the corresponding

features. Consequently, we are forced to work with the non-parametric representation that the training data captures. This implies that one has to resort to a non-parametric avenue in which we are able to compute the corresponding prototypes by approximating the distribution using a multi-dimensional kernel. Although a generalized kernel could be used for this phase by invoking a Parzen window methodology, in the interest of simplicity, for a *prima facie* case, we have opted to use a simplistic bin-based approach. Once the histogram of the features has been obtained in each dimension, the training sample that lies closest to the point representing the $\frac{n-k+1}{n+1}^{th}$ percentile of the first distribution and the $\frac{k}{n+1}^{th}$ percentile of the second distribution of the given data sets are *selected* to be the prototypes of interest. Indeed, by using these *selected* patterns as vector prototypes – *a single one from each class* – one can now achieve classification. One should observe that this *single* prototype is, as a vector, a "created" point, although, in every single dimension, the value is "selected" from the actual training sample that is closest to the value specified by the OS value.

Although the specific value of k is not so crucial [38, 50, 51], in this chapter, as mentioned earlier, we have set $k = 1$, implying that we have, in each dimension, worked with the pattern that falls at the $\frac{2}{3}$ percentile of the first distribution and the pattern that falls at the $\frac{1}{3}$ percentile of the second.

To obtain the final PRS, we can envision two methodologies, namely where the computations are vector-based or scalar-based. Each of them are described below.

7.3.1 The *Vector*-based Selective OS-based PRS

The *Vector*-based selective OS-based PRS is obtained by comparing the testing sample with the prototype procured by the above process. Such a comparison can be achieved using any metric, but for the sake of simplicity, we have utilized the well-known Euclidean norm. In the interest of completeness, the formal algorithm of this approach is provided as Algorithm 9.

The proposed method has been rigorously tested on the various artificial and real-life data sets obtained from the UCI repository [15] described above. It has also been compared with other well-known schemes including the Naïve Bayes, SVM, and the

Algorithm 9 `Vector_Based_Selective_PRS`(T, TP)

Input:

T: The training set, comprising of elements T_1 and T_2 from classes ω_1 and ω_2 respectively.

TP: the testing set

Output:

Classification for TP

Method:

Training

for Class ω_1 with the Training set T_1 **do**
  for i = 1 to d **do**
    Build histogram for Feature 'i'
    Select the point at $\frac{2}{3}$ percentile and assign to o_{1i}
  end for
end for
for Class ω_2 with the Training set T_2 **do**
  for i = 1 to d **do**
    Build histogram for Feature 'i'
    Select the point at $\frac{1}{3}$ percentile and assign to o_{2i}
  end for
end for
End_Training

Testing

for all $X \in TP$ **do**
  if $DIST(O_1, X) < DIST(O_2, X)$ **then**
    Assign X to class ω_1
  else
    Assign X to class ω_2
  end if
end for
End_Testing
End Algorithm

kNN. In order to obtain the results, the algorithms were executed 50 times with the 10-fold cross validation scheme. The results are tabulated in Table 7.2. To ensure standardization[2], the performance of the benchmark classifiers[3] are taken from [24, 25, 39, 42]. By examining the table of results, we can see that the proposed algorithm can achieve a comparable classification when compared to the other traditional classifiers, which is particularly impressive because once the *single* prototype has been computed after the training phase, the testing is done by exactly two vector-based computations (one for each class), comparing the testing sample with the resultant prototypes. For example, for the Breast Cancer data set, we can see that the new approach yielded an accuracy of 95.06% which should be compared to the accuracies of the SVM (96.99%), NB (96.40%) and the kNN (96.60%). The reader will observe that the classification accuracies for all the data sets is commendable except for the "Diabetes" set. This is because, for this data set, the approximation of the distributions using simplistic histograms in the d-dimensional space is rather crude. Superior results are obtained in this case when we resort to obtaining the OS-based points using the criteria explained in Section 7.4.1.

7.3.2 The *Scalar*-based Selective OS-based PRS

In the *Scalar*-based selective OS-based PRS, the patterns are treated as a group of scalars and a classification is performed for each dimension. Thereafter, the final determination of the identity of the testing sample is achieved based on a majority vote. In the interest of simplicity, this vote is done by weighting all the features equally. The formal algorithm for this is found in Algorithm 10.

The scalar-based selective CMOS has been tested on the various artificial and real-life data sets and the results are tabulated in Table 7.3. If we examine the table, one can see that the approach yields a near optimal accuracy for the all the

[2]The standard results will be the same for all the following subsections, and so this fact will not be repeated.

[3]Unfortunately, the results reported in these publications do not always contain the standard deviations of the corresponding accuracies. We have thus included them only for our schemes. This comment will also be true for all the results reported in this chapter and the next.

Data Set	NB	NN	SVM	Selective CMOS - Vector
Artificial Set	98.85	92.90	98.75	94.45 ± 5.35
WOBC	96.40	96.6	96.99	95.06 ± 2.58
WDBC	92.97	96.66	97.71	90.82 ± 5.37
Diabetes	73.11	71.90	73.84	67.24 ± 5.95
Hepatitis	83.19	82.58	84.54	76.67 ± 4.06
Iris	95.13	96.0	96.67	92.50 ± 9.56
Statlog (Australian Credit)	87.4	85.9	85.51	84.21 ± 9.82
Statlog (Heart)	83.00	84.4	85.6	83.93 ± 5.92
Vote	94.29	90.23	94.33	91.62 ± 4.95

Table 7.2: Classification of the Artificial and Real-life data sets using the Vector-based *Selective* CMOS.

data sets except the Diabetes data set, which, as before has a poor accuracy for all the classifiers, and for which the histogram leads to a very crude approximation. For example, if we consider the Hepatitis data set, the proposed approach yields an accuracy of 81% while the traditional classifiers yields 84.54% (SVM), 82.58% (NN) and 83.19% (NB), which is still quite astonishing considering that all the information in the entire training set has been crystallized into a single prototype *distant from the mean.*

We now move on to present the vector and scalar-based "Creative" PRSs in which the Reference set has only a single element.

7.4 A CMOS-based "Creative" PRS Using a Parametric Perspective

We now consider the task of designing a "creative" OS-based PRS, where we again aim to attain the goal that the cardinality of the Reference set is unity. Since we are now

Data Set	NB	NN	SVM	Selective CMOS - Scalar
Artificial Set	98.85	92.90	98.75	85.83 ± 5.16
WOBC	96.40	96.6	96.99	94.52 ± 2.96
WDBC	92.97	96.66	97.71	87.25 ± 4.53
Diabetes	73.11	71.90	73.84	43.41 ± 9.10
Hepatitis	83.19	82.58	84.54	81.00 ± 4.05
Iris	95.13	96.00	96.67	78.80 ± 9.92
Statlog (Australian Credit)	87.40	85.90	85.51	48.19 ± 8.03
Statlog (Heart)	83.00	84.40	85.6	62.67 ± 9.96
Vote	94.29	90.23	94.33	85.52 ± 5.65

Table 7.3: Classification of the Artificial and Real-life data sets using the Scalar-based *Selective* CMOS.

willing to permit the option of assuming a distributional form for the corresponding features, we have chosen to resolve this fundamental issue by invoking a strategy analogous to a Naïve-Bayes' approach, although it, really, is of an *anti*-Naïve-Bayes' paradigm. As a Naïve-Bayes' strategy requires the uncorrelation of the features, if we consider a k-OS CMOS, we need to determine, for every feature, the $\frac{n-k+1}{n+1}^{th}$ percentile of the first distribution and the $\frac{k}{n+1}^{th}$ percentile of the second distribution. From an anti-Naïve-Bayes'perspective, we can obtain the corresponding values of all of the features by assuming a Gaussian[4] distribution for all the features. The OS-based PRS that we thus propose here again consists of the *single created* point in the d-dimensional space characterized by the location of the $\frac{n-k+1}{n+1}^{th}$ percentile of the first distribution and the $\frac{k}{n+1}^{th}$ percentile of the second distribution. As shown in [38], for the value of $k = 1$, the 2-OS CMOS positions for the classes that follow a Gaussian

[4]Any other member of the exponential family described in [38] could have just as well been used. We have chosen to use the Gaussian distribution because it is more general than the others and involves the means and the variances of the features.

distributions can be expressed as:

$$\begin{aligned} o_1 &= \mu_1 - \frac{\sigma}{\sqrt{2\pi}}, \text{and} \\ o_2 &= \mu_2 + \frac{\sigma}{\sqrt{2\pi}}. \end{aligned} \tag{7.1}$$

We thus opt to use these expressions to obtain the corresponding CMOS positions, whence the vector and scalar-based PRS schemes are derived.

7.4.1 The *Vector*-based "Creative" OS-based PRS

For this approach also, we consider the possibility of perceiving the training set as vectors or as scalars. The *Vector*-based "Creative" OS-based PRS considers the final prototype as a vector which has been artificially created as a new pattern by resorting to Eq. (7.1). The testing sample is then compared with the *single* OS-based prototype, and the identity is determined with regard to how distant it is from the latter. Since the individual variances are known, this distance is computed using the Mahalanobis distance, referred to as M_Dist. The formal algorithm for this approach is given in Algorithm 11.

The vector-based *Creative* CMOS has been tested for the same data sets as before, and the results are tabulated in Table 7.4. From the table, we can conclude that the new approach is comparable with the other well-used and well-established classifiers. This approach achieves "almost" optimal classification when compared to the traditional classifiers. For example, if we consider the classification of the Breast Cancer data set, we see that Algorithm 11 achieves 96.94% accuracy as opposed to the 96.99% of SVM, 96.40% of NB and 96.6% of NN. Indeed, one can see that the difference in the accuracies is almost negligible. For the other data sets too, this approach attains a near-optimal classification when compared to the traditional classifiers, even though there is only a single element in the Reference set and the testing involves only two vector comparisons.

Data Set	NB	NN	SVM	Creative CMOS - Vector
Artificial Set	98.85	92.90	98.75	98.85 ± 1.72
WOBC	96.40	96.6	96.99	96.94 ± 2.36
WDBC	92.97	96.66	97.71	93.43 ± 2.87
Diabetes	73.11	71.90	73.84	73.76 ± 4.38
Hepatitis	83.19	82.58	84.54	76.67 ± 5.90
Iris	95.13	96.00	96.67	95.60 ± 5.35
Statlog (Australian Credit)	87.40	85.90	85.51	94.97 ± 2.43
Statlog (Heart)	83.00	84.40	85.6	86.52 ± 6.40
Vote	94.29	90.23	94.33	93.62 ± 3.31

Table 7.4: Classification of the Artificial and Real-life data sets using the Vector-based *Creative* CMOS.

7.4.2 The *Scalar*-based "Creative" OS-based PRS

In Algorithm 11, each pattern was considered as a vector, and the distance calculations were based on the Mahalanobis metric. As in the case of the selective scheme described in Section 7.3.2, a similar classification can be achieved by considering the various feature values as scalars and by accomplishing the task by computing the majority vote. Algorithm 12 provides a formal algorithm for this approach.

The scalar-based creative CMOS has also been tested on the various artificial and real-life data sets and the results are tabulated in Table 7.5. Again, an examination of the table shows that the classification results are near-optimal. For example, if we consider the Vote data set, the proposed approach yields an accuracy of 93.43% while the traditional classifiers yields 94.33% (SVM), 90.24% (NN) and 94.29% (Naive Bayes (NB)). Observe that the prototype-based NN performs even better than the traditional NN which involves the entire training set, which is quite astonishing considering that all the information in the entire training set has been crystallized into a single newly-created prototype.

Data Set	NB	NN	SVM	Creative CMOS - Scalar
Artificial Set	98.85	92.90	98.75	84.93 ± 5.45
WOBC	96.40	96.6	96.99	94.35 ± 2.39
WDBC	92.97	96.66	97.71	89.46 ± 3.66
Diabetes	73.11	71.90	73.84	76.74 ± 3.77
Hepatitis	83.19	82.58	84.54	82.67 ± 8.64
Iris	95.13	96.00	96.67	94.40 ± 6.37
Statlog (Australian Credit)	87.40	85.90	85.51	84.59 ± 3.92
Statlog (Heart)	83.00	84.40	85.60	83.33 ± 5.69
Vote	94.29	90.23	94.33	89.62 ± 4.24

Table 7.5: Classification of the Artificial and Real-life data sets using the Scalar-based *Creative* CMOS.

7.5 Classification Based On One Selected Feature

In this section we have embarked on an even more ambitious goal which consists of seeing if we could do the classification by using only a *single* feature of a *single* prototype. To achieve this goal, we have operated with the philosophy proposed in Section 7.4 and designed a "creative" vector PRS. But rather than use all the components of the vector in the classification, we have merely chosen the OS-based points of a *single feature*, where the $\frac{n-k+1}{n+1}^{th}$ percentile of *this* feature of the first class, and the $\frac{k}{n+1}^{th}$ percentile of *this* feature of the second class, are the corresponding "prototypes" (where we have, as usual, used the value of $k = 1$). Algorithm 13 describes the scheme formally.

The proposed approach of Algorithm 13 has been tested on the artificial and real-life data sets described earlier and the results are tabulated in Table 7.6. If we closely investigate the table, one can see that the method attains a comparable classification when compared to the traditional classifiers. Specifically, for the Diabetes data set, if the classification is performed based on the OS positions of the feature *Plasma Glucose*

Concentration, an accuracy of 73.63% is attained as opposed to the accuracy of 73.84% attained by SVM[5]. The reader should not be surprised that the accuracies are not always so outstanding. Indeed, it is clear that the accuracy of this *scalar*-based OS will be inferior to that of the corresponding *vector*-based OS. However, astonishingly enough, the accuracy does not degrade significantly – the resultant classifier still yields an accuracy that is acceptable considering the fact that one requires only two *scalar* comparisons to achieve the classification.

Data set	SVM	Dimension	Feature	CMOS
Artificial Set	98.75	3	A3	98.55 ± 1.88
WOBC	96.99	2	Uniformity of Cell Size	93.04 ± 2.59
WDBC	97.71	27	Worst Compactness	91.29 ± 3.62
Diabetes	73.84	2	Plasma Glucose Concentration	73.63 ± 4.11
Hepatitis	84.54	12	Ascites	85.47 ± 6.89
Iris	96.67	4	Petal Width	95.50 ± 4.68
Statlog (Australian Credit)	85.51	8	A9	84.84 ± 4.06
Statlog (Heart)	85.60	2	Chest Pain Type	78.52 ± 7.21
Vote	94.33	4	Physician-fee-freeze	95.95 ± 2.44

Table 7.6: Classification of the Artificial and Real-life data sets using the Scalar-based *Creative* CMOS involving only a single dimension.

7.6 Conclusions

In this chapter, we have demonstrated the power and potential of CMOS to yield single-element PRSs which are either "selective" or "creative", where in each case we resort to a non-parametric or a parametric paradigm respectively. All of these schemes have led to what we believe, "ultimate" PRSs, because they all require only a single element from each class.

All of these solutions have been used to achieve classification for artificial and real-life data sets from the UCI Machine Learning Repository, where we have followed an

[5]The result for the Vote data set for this scheme is actually higher than that obtained for the SVM. This must, really, be considered as an abberation.

approach that is similar to the Naïve-Bayes' strategy although it is essentially of an anti-Naïve-Bayes' paradigm. All of the reported algorithms yielded an acceptable accuracy when compared to many of the established benchmark methods. It is even more fascinating to see that our paradigm performs favorably by using the information in a *single feature* of such a *single data point.*

In the next chapter, Chapter 8, which represents the final chapter of *our contributions*, we will demonstrate how the principle behind the CMOS paradigm can be utilized to formulate and implement novel BI algorithms.

Algorithm 10 `Scalar_based_Selective_PRS`(T, TP)

Input:

T: The training set, comprising of elements T_1 and T_2 from classes ω_1 and ω_2 respectively.

TP: the testing set

Output:

Classification for TP

Method:

Training

1: **for** Class ω_1 with the Training set T_1 **do**
2: **for** i = 1 to d **do**
3: Build histogram for Feature 'i'
4: Select the point at $\frac{2}{3}$ percentile and assign to o_{1i}
5: **end for**
6: **end for**
7: **for** Class ω_2 with the Training set T_2 **do**
8: **for** i = 1 to d **do**
9: Build histogram for Feature 'i'
10: Select the point at $\frac{1}{3}$ percentile and assign to o_{2i}
11: **end for**
12: **end for**
End_Training

Testing

1: **for all** $X \in TP$ **do**
2: **for** i = 1 to d **do**
3: Compare component x_i with o_{1i} and o_{2i}
4: Determine the class of x_i
5: **end for**
6: Assign X to the class based on a majority vote on the classification obtained for all x_i's
7: **end for**
End_Testing
End Algorithm

Algorithm 11 `Vector_based_Creative_PRS`(T, TP)

Input:

T: The training set, comprising of elements T_1 and T_2 from classes ω_1 and ω_2 respectively.

TP: the testing set

Output:

Classification for TP

Method:

Training

for i = 1 to d **do**
  Estimate mean of T_1 as m_{1i} and mean of T_2 as m_{2i}
  Estimate the standard deviations of T_1 and T_2 as σ_{1i} and σ_{2i}
end for
for i = 1 to d **do**
  Determine the i^{th} component of O_1, $o_{1i} = m_{1i} - \frac{\sigma_{1i}}{\sqrt{2\pi}}$
  Determine the i^{th} component of O_2, $o_{2i} = m_{2i} + \frac{\sigma_{2i}}{\sqrt{2\pi}}$
end for

End_Training

Testing

for all $X \in TP$ **do**
  if $M\_Dist(O_1, X) < M\_Dist(O_2, X)$ **then**
    Assign X to class ω_1
  else
    Assign X to class ω_2
  end if
end for

End_Testing

End Algorithm

Algorithm 12 `Scalar_based_Creative_PRS`(T, TP)

Input:

T: The training set, comprising of elements T_1 and T_2 from classes ω_1 and ω_2 respectively.

TP: the testing set

Output:

Classification for TP

Method: Training

1: **for** i = 1 to d **do**
2: Estimate mean of T_1 as m_{1i} and mean of T_2 as m_{2i}
3: Estimate the standard deviations of T_1 and T_2 as σ_{1i} and σ_{2i}
4: **end for**
5: **for** i = 1 to d **do**
6: Determine the i^{th} component of O_1, $o_{1i} = m_{1i} - \frac{\sigma_{1i}}{\sqrt{2\pi}}$
7: Determine the i^{th} component of O_2, $o_{2i} = m_{2i} + \frac{\sigma_{2i}}{\sqrt{2\pi}}$
8: **end for**

End_Training

Testing

1: **for all** $X \in TP$ **do**
2: **for** i = 1 to d **do**
3: **if** $M\_Dist(o_{1i}, x_i) < M\_Dist(o_{2i}, x_i)$ **then**
4: Assign X to class ω_1 for dimension i
5: **else**
6: Assign X to class ω_2 for dimension i
7: **end if**
8: **end for**
9: Assign X to the class based on a majority vote on the classification obtained for all x_i's
10: **end for**

End_Testing
End Algorithm

Algorithm 13 `Classification_by_Selected_Feature`(T, TP)

Input:

T: The training set, comprising of elements T_1 and T_2 from classes ω_1 and ω_2 respectively.

TP: the testing set

Output:

Classification for TP

Method:

Training

for i = 1 to d **do**
  Estimate mean of T_1 as m_{1i} and mean of T_2 as m_{2i}
  Estimate standard deviations of T_1 and T_2 as σ_{1i} and σ_{2i}
end for
for i = 1 to d **do**
  Determine $o_{1i} = m_{1i} - \frac{\sigma_{1i}}{\sqrt{2\pi}}$
  Determine $o_{2i} = m_{2i} + \frac{\sigma_{2i}}{\sqrt{2\pi}}$
end for
for i = 1 to d **do**
  for all $X \in T$ **do**
    if $M\_Dist(o_{1i}, x_i) < M\_Dist(o_{2i}, x_i)$ **then**
      Assign X to class ω_1 for dimension i
    else
      Assign X to class ω_2 for dimension i
    end if
  end for
end for
for all $X \in T$ **do**
  Determine the dimension (feature) i that correctly classifies it
end for
Select the best dimension (feature) i based on the results for the entire set, and store it as D

End_Training

Testing

for all $X \in TP$ **do**
  Using M_Dist, classify x_D with respect to o_{1D} and o_{2D}
end for

End_Testing

End Algorithm

Chapter 8

A Novel "Anti"-Bayesian Border Identification Algorithm Applicable for Real-Life Data

8.1 Introduction

As explained in Chapter 2, Border Identification (BI) algorithms, which are a subset of PRSs, work with a Reference set which only contains "border" points. Specializing this criterion, the current-day BI algorithms, designed by Duch, Foody, and Li *et al.* (see Section 2.3.3), attempt to select a Reference set which contains border patterns derived, in turn, from the set of training patterns. These algorithms also, in effect, yield reduced training sets. As opposed to the latter, we are interested in determining border patterns which, in some sense, are closer to the true optimal classifier, and which can thus better classify the entire testing set. We remark that when the task is accomplished, such a Reference set would contain the patterns drawn from different classes and they would lie near the optimal classifier. Contrary to a Bayesian intuition, these border patterns have the ability to accurately classify the testing patterns, as we shall presently demonstrate. Our method is a combination of NN computations

and (Mahalanobis) *multi*-dimensional[1] distance computations which yields the border points that are subsequently used for the purpose of classification. The characterizing component of our algorithm, referred to as ABBI, is that classification can be done by processing the obtained border points by themselves without invoking, for example, a subsequent SVM phase.

How then can one determine the border points themselves? This, indeed, depends on the model of computation - for example, whether we are working within the parametric or non-parametric model. This chapter deals with the former model, where the information about the classes is crystallized in the class-conditional *distributions* and their respective parameters[2], where the training samples are used to estimate the *parameters* of these models. In this chapter, we have shown how the border points can be obtained by utilizing the information gleaned from the estimated distributions. Observe that with regard to classification and testing, all of these computations can be considered to be of a "pre-processing" nature, and so the final scheme would merely be of a Nearest Neighbor (NN)-sort. The details of how this is achieved is described in detail in this chapter.

The goal of this chapter is not to merely present new BI algorithms. Rather, put in the context of the OS-based PR schemes and criteria studied in Chapters 3 -7, we would like to design and implement a new BI algorithm which, essentially, mirrors the phenomena crystallized by the OS. In other words, we would like to design novel BI algorithms that utilize the patterns which are neither close to the mean nor on the exact borders of the classes, and if possible to be "close" to the location of the $\frac{n-k+1}{n+1}^{th}$ percentile of the first distribution and the $\frac{k}{n+1}^{th}$ percentile of the second distribution. While this goal is ambitious, we demonstrate that it is achievable.

Apart from its straightforward significance in classification, philosophically, the BI problem can be abstracted for more complex situations. It is true that in the most simplistic sense, the borders could consist of the samples that could enable the

[1] We also have some initial results in which the distance and optimizations are done using lower-dimensional projections, the results of which are subsequently fused using an appropriate fusion technique.

[2] Although conclusive analogous results for the non-parametric model are not currently available, our initial investigations seem to indicate that this is a rich avenue for research.

inter-class discrimination. But we suggest that this concept can be abstracted to much higher and finer levels. From this abstract perspective, the "borders" could be the *types* of measurements used in a PR system – i.e., one which could be perceived as a classification problem in its own right – where one has to determine *which* measurements are pertinent to the problem domain and which measurements are not[3]. Indeed, in an even more abstract sense, a feature selection methodology can also be aptly modeled as a BI algorithm. While all of these scenarios present research potentials for the (distant) future, in the interest of simplicity, we shall propose the determination and use of BI methods to efficiently and accurately design classifiers that only utilize the "border" points of the respective classes to achieve classification. The way by which our new BI algorithm relates to the OS-based PR schemes described in Chapters 3 - 7 is explained in Section 8.2.1.

8.2 A Novel Two-Class "Anti-Bayesian" BI Scheme

8.2.1 The Formal Algorithm

The problem of determining the border points for the parametric model of computation can be solved for fairly complex scenarios. We shall achieve this progressively. In this section, we shall first consider the two-class problem.

When one examines the existing BI schemes, one observes that the information that has been utilized to procure the border patterns is primarily (and indeed, essentially) distance-based. In other words, the distances between the patterns are evaluated independently, and the border patterns are obtained based on such distances. The patterns obtained in this manner are considered as the new training set, which reduces these BI schemes to be special types of PRSs, but with the border patterns being the Reference set. However, as these border patterns are only the "Near" ones, they do not possess sufficient information to train an efficient classifier. We shall now rectify this.

We now mention a second major handicap associated with the traditional BI

[3]This study is beyond the scope of this thesis and is a future research avenue.

schemes. Once they have computed the border points associated with the specific classes, the traditional schemes operate by determining a "classifier" based on the new set. In other words, they have to determine a classifying *boundary* (linear, quadratic or SVM-based) to achieve this classification. As the reader will observe, in our work, we attempt to circumvent this entire phase. Indeed, in our proposed strategy, we merely achieve the classification using the final *small* subset of border points – which entails a significant reduction in computation.

The reader should also observe that this final decision would involve NN-like computations with a *few* points. The intriguing feature of these few points is that they lie close to the boundary and not to the mean, implying an "anti-Bayesian" philosophy [38, 50, 51].

In order to obtain the border patterns of the distributions ω_1 and ω_2 in an "anti-Bayesian" approach, we make use of the axiom that the patterns that have nearest neighbors from *other* classes *along* with the patterns of the same class fall into a common region - which is, by definition, the overlapped region.

The proposed algorithm has 4 parameters, namely, J, J_1, J_2 and K. First of all, J denotes the number of border points that have to be selected from each class. We understand that in the process of selecting the border points, the training set must be "examined" so as to ignore the patterns which are not relevant for the classification. As this decision is taken based on the border points in and of themselves, we conclude that the patterns which are in the overlapping region are not able to provide an accurate decision, and so these points have to be ignored. Thus, for any X, those patterns with J_2 or more NNs out of the J_1 NNs, which are not from the same class as X, are ignored.

To be more specific, in order to eliminate the overlapping points, we first determine J_1-NNs of every pattern X. If J_2 or more of these NN patterns are from the same class, this pattern X is added to the new training set. Once this step is achieved, we are left with the training points which are not overlapping with any other classes. Thereafter, we evaluate the (Mahalanobis) distance[4](MD) of every pattern of the new training

[4]Any well-defined norm, appropriate for the data distribution, can be used to quantify this distance.

set with respect to the *mean* of both the classes. Both of these phases distinguish our particular strategy. The patterns which are closest to being equidistant from both the classes, and which are not determined to be overlapping with respect to the other classes, are added to the Border set.

The process of determining the (Mahalanobis) distances with respect to both the classes, is repeated for all the patterns of the new training set, and a decision is made for each pattern based on the difference between these distances.

The two-dimensional view of this philosophy is depicted in Figures 8.1 - 8.3. The border patterns obtained by applying this method are also given in the figure, where the border patterns of class ω_1 are specified by rectangles, and those of class ω_2 are specified by circles. We now make the following observations:

1. If we examine Figure 8.1, we can see that the border patterns that are specified by rectangles and circles are precisely those that lie close to the OS-based positions – reasonably close to the $\frac{10}{11}^{th}$ percentile of the first class and $\frac{1}{11}^{th}$ percentile of the second class. As the classes are fairly separable (they have a Bhattacharyya Distance, $BD = 4.52$), any of the symmetric OS pairs can attain a reasonable classification.

2. However, if the classes are semi-overlapped, then the "more interior" symmetric percentiles, such as the $\langle \frac{2}{3}, \frac{1}{3} \rangle$ can perform a near-optimal classification. This can be seen in Figure 8.2. The patterns in this figure have more overlap (the $BD = 1.69$), and the border points chosen are the ones which lie just outside the overlapping region. From the OS perspective, these points are reasonably close to the ones that lie at the $\frac{2}{3}^{th}$ percentile of the first class and $\frac{1}{3}^{th}$ percentile of the second class.

3. The same argument is valid for Figure 8.3. In the OS-based classification, we have seen that if the classes have a large overlap as in Figure 8.3 (in this case, $MD = 0.78$), then the Dual CMOS has to be invoked to achieve a reasonable classification. In such a case, one can see that the border patterns are the similar to the patterns obtained by invoking a Dual CMOS.

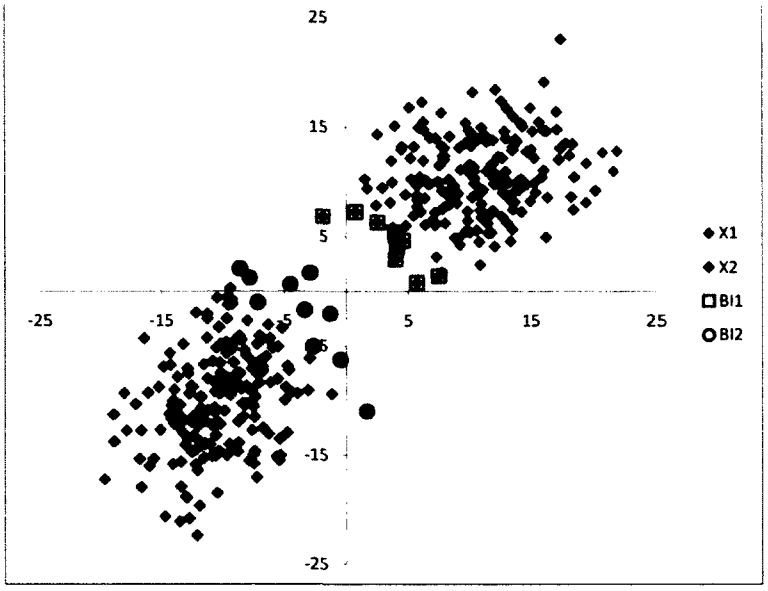


Figure 8.1: The border patterns for fairly separable classes obtained by ABBI.

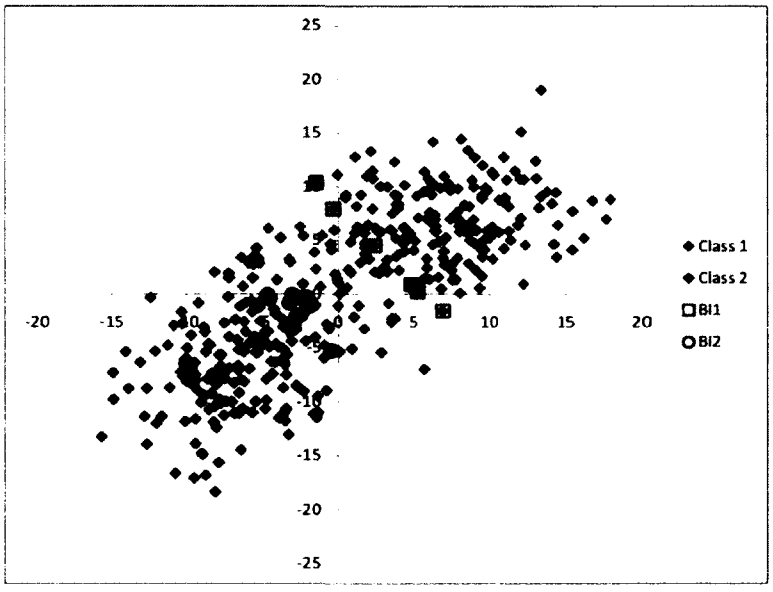


Figure 8.2: The border patterns for semi-overlapped classes obtained by ABBI.

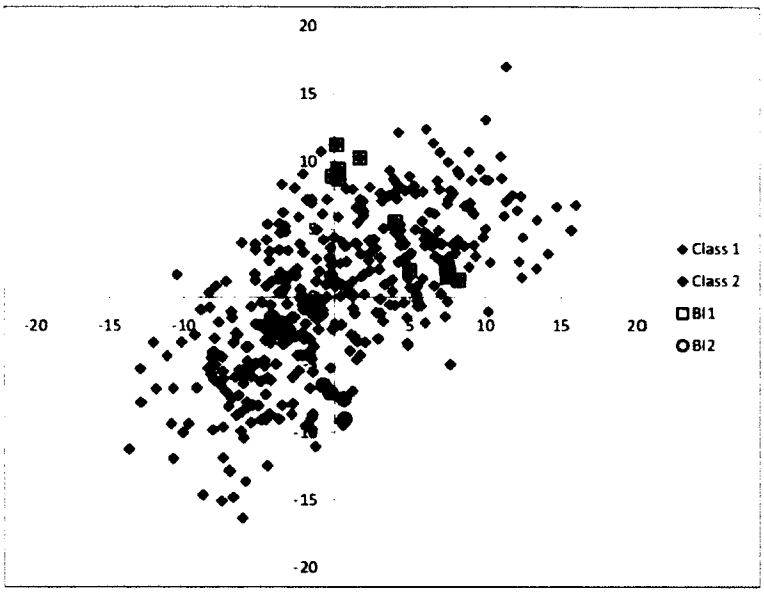


Figure 8.3: The border patterns for overlapped classes obtained by ABBI.

The algorithm for obtaining the border patterns, ABBI, is formally given in Algorithm 14.

Contrary to traditional BI algorithms, ABBI requires only a small number of border patterns for the classification. For example, consider the Breast Cancer data set which contains 699 patterns. Traditional BI algorithms will obtain a border set of around 150 patterns for this data set. Furthermore, once these methods have obtained the border points, they will have to generate a classifier for the new reduced set to achieve the classification. As opposed to this, our method requires only 20 border patterns, and the classification is based on the *five* NN border patterns of the testing pattern.

8.3 Experimental Results

The proposed method ABBI has been tested on various data sets that include artificial and real-life data sets obtained from the UCI repository [15]. ABBI has also been compared with other well-known methods which include the NB, SVM, and the kNN.

Algorithm 14 ABBI(ω_1, ω_2)

Input:

Data from two classes; ω_1, ω_2, whose means are $\mathbf{M}_1$ and $\mathbf{M}_2$ respectively.

Parameters: J_1, J, J_2, K: Small numbers

Assumption:

Dist computes the distance between two vectors.

DistDiff computes the difference in distances obtained with respect to μ_1 and μ_2

Assumption:

NTR_1 and NTR_2 are the new training sets which do not contain points in the overlapped region.

Output:

The classification based only on the Border points

Method:

1: $NTR_1 \leftarrow \emptyset$
2: $NTR_2 \leftarrow \emptyset$
3: Divide points of ω_1 into training and testing sets, TRP_1 and T_1 respectively
4: Divide points of ω_2 into training and testing sets, TRP_2 and T_2 respectively
5: **for all** $X \in TRP_1$ **do**
6: Compute J_1 NNs of X
7: If J_2 or more NNs are from class ω_1, $NTR_1 \leftarrow NTR_1 \cup X$
8: **end for**
9: **for all** $X \in TRP_2$ **do**
10: Compute J_1 NNs of X
11: If J_2 or more NNs are from class ω_2, $NTR_2 \leftarrow NTR_2 \cup X$
12: **end for**
13: **for all** $X \in NTR_1$ **do**
14: Dist(X, M_1)
15: Dist(X, M_2)
16: **end for**
17: **for all** $X \in NTR_2$ **do**
18: Dist(X, M_1)
19: Dist(X, M_2)
20: **end for**
21: **for all** $X \in NTR_1$ **do**
22: DistDiff(X)
23: **end for**
24: **for all** $X \in NTR_2$ **do**
25: DistDiff(X)
26: **end for**
27: Add J points with minimum DistDiff from NTR_1 and NTR_2 to BI
28: Classify testing points using a K-NN based on the points in BI.

End Algorithm

In order to obtain the results, ABBI algorithm was executed 50 times with the 10-fold cross validation scheme.

8.3.1 Artificial Data Sets

For a *prima facie* testing of artificial data, we generated two classes that obeyed Gaussian distributions. To do this, we made use of a Uniform [0, 1] random variable generator to generate data values that follow a Gaussian distribution. The expression $\mathbf{z} = \sqrt{-2ln(u_1)}\, cos(2\pi u_2)$ is known to yield data values that follow $N(0,1)$ [7]. Thereafter, by using the technique described in [16], one can generate Gaussian random vectors which possess any arbitrary mean and covariance matrix. In our experiments, since this is just for a *prima facie* case, we opted to perform experiments for two-dimensional and three-dimensional data sets. The respective means of the classes were $[\mu_{11}, \mu_{12}]^T$ and $[\mu_{21}, \mu_{22}]^T$ for the two-dimensional data, and $[\mu_{11}, \mu_{12}, \mu_{13}]^T$ and $[\mu_{21}, \mu_{22}, \mu_{23}]^T$ for the three-dimensional data. Further, the corresponding covariance matrices of the two-dimensional classes had the forms:

$$\Sigma_1 = \begin{bmatrix} a^2 & \alpha ab \\ \alpha ab & b^2 \end{bmatrix}, \qquad \Sigma_2 = \begin{bmatrix} b^2 & \alpha ab \\ \alpha ab & a^2 \end{bmatrix}$$

The covariance matrices for the three-dimensional classes had the forms:

$$\Sigma_1 = \begin{bmatrix} a^2 & 0 & \alpha ab \\ 0 & 1 & 0 \\ \alpha ab & 0 & b^2 \end{bmatrix}, \qquad \Sigma_1 = \begin{bmatrix} b^2 & 0 & \alpha ab \\ 0 & 1 & 0 \\ \alpha ab & 0 & a^2 \end{bmatrix}$$

With regard to the cardinality of the data set, each of the classes had 200 instances in the corresponding two and three-dimensional space.

For the distance computations, we used the Mahalanobis distance (MD), which is based on the means and the covariance matrices Σ_1 and Σ_2. It is based on the correlations between the variables using which different patterns can be identified and analyzed. The MD of a multivariate vector $X = (X_1, X_2, ..., X_N)^T$ from a group of values with mean $M = (\mu_1, \mu_2, ..., \mu_N)^T$ and covariance matrix Σ is defined as

$$MD(X) = \sqrt{(X - M)^T\, \Sigma^{-1}\, (X - M)}.$$

In order to analyze the results and to evaluate the difficulty of the problem tackled, we used the Bhattacharyya distance (BD) as the measure for the comparison, which

can be considered to be a measure of the *separability* between the classes. For the multivariate Gaussian distribution, the Bhattacharyya distance (BD) can be evaluated as

$$BD = \frac{1}{8}\,(M_1 - M_2)^T\,\Sigma^{-1}\,(M_1 - M_2) + \frac{1}{2}\,ln\frac{\det\,\Sigma}{\sqrt{\det\,\Sigma_1\,\det\,\Sigma_2}},$$

where M_i and Σ_i are the respective means and the covariances, and $\Sigma = \frac{\Sigma_1+\Sigma_2}{2}$.

In order to not make the chapter too cumbersome, the *specific* details of the values of the μ's, a, b and α (for the means and covariances), are not included here[5]. However, what is crucial to guarantee "repeatability", are the respective values of the BD for each experimental setting, and *these* are clearly specified in every single row.

8.3.2 Experimental Results: Artificial Data Sets

The experimental results obtained for two dimensional artificial data sets can be seen in Table 8.1 and those for three dimensional artificial data sets can be seen in Table 8.2.

BD	1NN	3NN	SVM	ABBI
4.52	100	100	100	100
2.94	99.10	99.20	99.25	99.25
1.69	95.30	96.50	97.00	96.40
0.78	84.15	86.05	88.25	88.0
0.45	73.55	75.45	81.50	80.55

Table 8.1: Results of the classification of various two dimensional artificial data sets where the BDs are given in the first column. For all these cases, the values of J_1, J_2, J and K are 10, 9, 10 and 5 respectively.

By examining Tables 8.1 and 8.2, one can see that ABBI can achieve remarkable classification when compared to that attained by the benchmark classifiers. For example, if we consider the case where the classes are separated by a BD of 1.66 in Table

[5]The inclusion of these values is actually not too relevant inasmuch as we have reported the Bhattacharyya Distance in each case.

Class Nature	Average BD	1NN	3NN	SVM	ABBI
Separated	6.08	100	100	100	100
Semi-overlapped	2.64	96.92	97.67	97.81	95.67
Overlapped	2.42	94.50	95.50	96.50	94.72
Highly overlapped	1.43	83.50	87.23	88.79	85.20

Table 8.2: Results of the classification of various three dimensional artificial data sets where the average BD are given in the first column.

8.1, ABBI can achieve a classification accuracy of 95.38%, while the 3NN achieves 97.25%. This is quite fascinating when we consider the fact the ABBI performs the classification based *only on 5 samples* from the selected 10 samples from each class, whereas the classification of NN involves the entire training set.

8.3.3 Real-life Data Sets

The data sets that we used in Chapter 7 are again tested with the new approach.

8.3.4 Experimental Results: Real-life Data Sets

The experimental results obtained using the ABBI algorithm are given in Table 8.3.

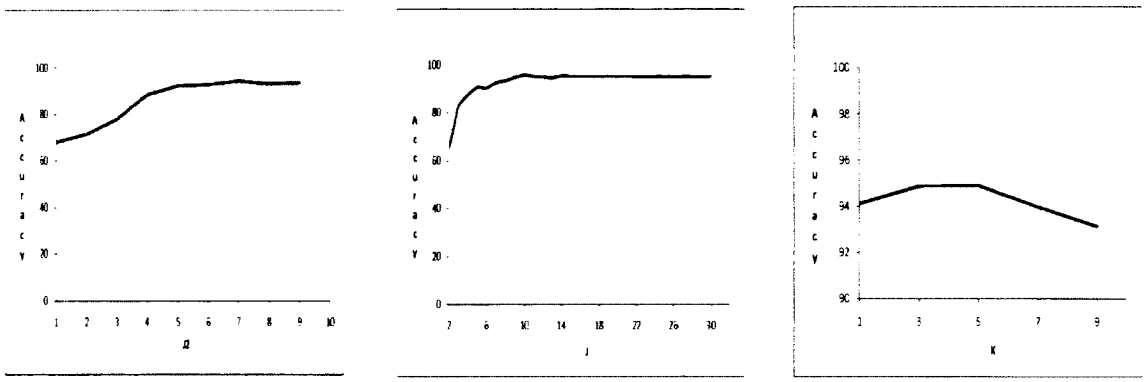
From the table of results, one can see that the proposed algorithm achieves a comparable classification when compared to the other traditional classifiers, which is particularly impressive because only a very few samples are involved in the process. For example, for the WOBC data set, we can see that the new approach yielded a accuracy of 95.80% which should be compared to the accuracies of the SVM (95.99%), NB (96.40%) and the kNN (96.60%). Similarly, for the Iris data set, ABBI can achieve an accuracy of 94.53%, which is again comparable to the performance of SVM (96.67%), NB (96.00%), and NN (95.13%).

As we have mentioned earlier, each data set possesses unique values for the parameters J_1, J_2, J, and K, which can be determined during the training phase. The

Data set	kNN	NB	SVM	ABBI
WOBC	96.60	96.40	95.99	95.80 ± 2.76
WDBC	96.66	92.97	97.71	92.39 ± 3.58
Diabetes	75.26	73.1098	76.70	72.30 ± 4.01
Hepatitis	82.58	83.19	82.51	80.27 ± 1.31
Iris	95.13	96.00	96.67	95.33 ± 4.81
Statlog (Australian Credit)	85.90	87.40	85.51	78.85 ± 4.22
Statlog (Heart)	84.40	83.00	85.60	82.50 ± 7.99
Vote	94.2857	90.23	94.33	90.76 ± 4.34

Table 8.3: The classification results for the various algorithms for Real-life data sets.

accuracies obtained for different values of J_1, J_2, J, and K for the WOBC data set are plotted in Fig.8.4. From this analysis, we conclude that the best values of J, J_1, J_2 and K for the breast cancer data set are 10, 8, 5, and 5 respectively.



(a) J_2 vs Accuracy (b) J vs Accuracy (c) K vs Accuracy

Figure 8.4: Accuracy of WOBC data set against different values of J, J_2, and K.

The best values for the parameters J, J_1, J_2, and K for various data sets is given in Table 8.4.

Data set	J_1	J_2	J	K
WOBC	8	10	5	5
WDBC	8	10	5	5
Diabetes	10	10	9	5
Hepatitis	5	5	5	3
Iris	10	10	9	5
Vote	8	10	7	5

Table 8.4: Parameter values for different data sets.

8.4 Conclusions

In this chapter, we have proposed a novel BI algorithm which involves the border patterns selected with respect to a new definition of the term "border". In line with the newly proposed OS-based anti-Bayesian classifiers of Chapters 3 - 7, we created the "border" set by selecting those patterns which are close to the true border of the alternate class. The classification is achieved with regard to *these* border patterns alone, and the size of this set is very small, in some cases, as small as five from each class. The resultant accuracy is comparable to that attained by other well-established classifiers. The superiority of this method over other BI schemes is that it yields a relatively small border set, and as the classification is based on the border patterns themselves (without involving an additional auxiliary classifier), it is computationally inexpensive.

Chapter 9

Conclusions, Summary & Future Work

9.1 Conclusions

When we initially embarked on this research work, the "long term" goal was to get a deeper understanding of how "Border Identification" (BI) was central to the field of Pattern Recognition (PR). The literature had reported numerous studies on the significance and determination of borders but we felt that the scientific "rationale" behind working with border points was not crystallized.

The literature also had scores of papers on Prototype Reduction Schemes (PRSs). The relationship between PRSs and BI algorithms was also not clearly described in the existing literature.

The main breakthrough in our research came when we discovered that all the existing work in PR dealt with the *distributions* of the features for the individual classes but that, in every case, all the work done only considered indicators such as means, variances etc. Apart from these indicators, we observed that a distribution has many other characterizing indicators, for example, those related to its Order Statistics (OS). It was quite astonishing that although some of them are quite unrelated to the traditional moments themselves, they had not been used in achieving PR. The main question that we then considered was whether these indicators/indices possess

any potential in PR. The entire research "took off" when we got a deeper insight into what the OS really signified and when we discovered *the methodology* by which the OS-based indices could be used in PR. It was, indeed, quite astonishing when we discovered that these OS-based classifiers operated in a completely "anti-Bayesian" manner (because they only involve points *distant* from the mean) and that they could yet attain the Bayesian gold standard for a classifier.

We can thus, as a concluding remark, confidently state that in this thesis, we have, in a pioneering and novel manner:

- Introduced the theory of optimal anti-Bayesian PR using the OS of the features rather than the distributions of the features themselves. Our novel methodology, has been referred to as Classification by Moments of Order Statistics (CMOS);
- Proven the properties of CMOS for many uni-dimensional and multi-dimensional distributions within the exponential family namely the Uniform, Doubly-Exponential, Gaussian, and the theoretical results have been verified by rigorous experimental testing;
- Extended the latter results significantly by considering asymmetric distributions within the exponential family like the Rayleigh, Gamma, and Beta distributions (for some of which even the closed form expressions of the cumulative distribution functions are not available) for which a near-optimal accuracy has been achieved;
- Shown how the field of PRSs are related to the theory of OS-based classifiers;
- Extended the OS-based classifiers to design PRSs which contain only a single element for each class;
- Presented a sound and formal theoretical foundation for the families of BI algorithms;
- Used these concepts to also design a new BI algorithm, referred to as ABBI.

All of these results have been earlier unreported in the literature, and have led to refereed publications in *Pattern Recognition*, one of the finest journals in the field, and in the Proceedings of a few refereed conferences.

9.1.1 Summary of Work Done

Chapter 1: We began this manuscript by providing the main reasons that led us to study this field and presented the motivations that inspired us, and the objectives that we intended to achieve.

Chapter 2: In this chapter, we provided a thorough literature survey on various aspects concerning the different phases of a PR system including feature extraction, training, classification, testing and system evaluation, and a brief analysis about a classifier's accuracy, the Receiver Operating Characteristics (ROC) curve, and the Area Under the Curve (AUC). We also provided a clear overview on the types of classifiers which are currently in use. The chapter also presented a comprehensive survey of the concept of PRSs [17, 54] and various traditional and present day BI algorithms. Thereafter, the chapter discussed the difference between prototypes and border patterns, and continues to provide a detailed study of the families of BI algorithms, which, in turn, can be considered to be a subset of the PRS in which the reduced set contains only the border patterns.

Chapter 3: This chapter provided a novel "anti-Bayesian" approach for resolving pattern classification problems. Specifically, we showed that the optimal Bayes' bound can be attained by an "anti-Bayesian" strategy, which we named as Classification by Moments of Order Statistics (CMOS). We proved that the classification can be attained by working with a *very few* (indeed, sometimes two) points *distant* from the mean. Further, if we determine these points by the *Order Statistics* of the distributions, we showed that the method can attain the optimal Bayes' bound. We discussed the conditions for a generic uni-dimensional classifier, which were proven and experimentally justified for the uni-dimensional Uniform distribution.

Chapter 4: In this chapter, we demonstrated that the CMOS can be applied on various uni-dimensional distributions in the Exponential family. The detailed study

proved that the CMOS can attain the optimal Bayes' bound for various symmetric distributions like the Doubly-Exponential, Gaussian, and some Beta distributions. Theoretical analysis and experimental results were also provided to show the strength of the proposed method.

Chapter 5: This chapter extended the application of CMOS for various asymmetric distributions namely the Rayleigh, Gamma, and some other Beta distributions in the Exponential family. We showed that the CMOS can attain near-optimal classification for these asymmetric distributions and the error probability of CMOS, when compared to Bayes' bound, is truly negligible. We provided theoretical analysis and experimental results for all these distributions.

Chapter 6: In this chapter, CMOS was extended for multi-dimensional distributions. We provided theoretical analysis and experimental results for various two-dimensional symmetric and asymmetric distributions, and also paved the way to proceed for the analysis of other higher-order multi-dimensional distributions. Again, the theoretical analysis and experiments showed that the CMOS can attain the optimal Bayes' bound for symmetric multi-dimensional distributions, and near-optimal classification for asymmetric multi-dimensional distributions.

Chapter 7: This chapter presented a family of "Ultimate" PRSs based on the anti-Bayesian OS criteria. We derived single-element PRSs which are either "selective" or "creative", where in each case we presented a non-parametric or a parametric paradigm respectively. We also suggested a single-value-single-feature PRS in which only a single value of a single feature from both the classes are involved in the classification. We tested all the methods on the real life data sets which are taken from the UCI Machine Learning Repository [15].

Chapter 8: In this chapter, we proposed novel BI algorithms built on the concepts of OS-based classification. The algorithms were designed as per the new definition that we provided for the term "border". Again, the algorithm was tested for both artificial and real-life data sets, which confirmed the power of the scheme.

9.2 Future Work

We believe that our research can be extended to various problems that have not been studied in this thesis. These potentially new research directions are briefly listed below.

- Throughout our work, we tackled only the problem of binary classification. But it is clear that the solutions should be further extended for multi-class problems. While all the avenues for extending OS-based two-class solutions to multi-class problems are considered open, we propose the following as potential solutions:

    1. The first way by which this problem can be tackled would be by considering the problem as $\binom{C}{2}$ 2-class problems, where the OS points for these pairs are used in each classifier.
    2. The second strategy for resolving the multi-class problem would be by modeling it as C "one-versus-others" problems. The solution in this case would, necessarily, be more complex because the alternate classes would not have the same distributional form as the class being considered. This would, certainly, be a very interesting avenue for future research.

- The current research basically concentrated on supervised learning. The application of CMOS for unsupervised learning is still unresolved. We suggest that a possible solution would involve k-OS or "Fuzzy" k-OS clustering algorithms, which work on principles analogous to the traditional k-means or "Fuzzy" k-means clustering schemes.

- In our work, we have not handled the problem of unbalanced data. The application of CMOS when the classes are unbalanced is again a new avenue for further research. In this case, we believe that the modifications needed to extend our current solutions will be marginal, because we may have to just work with the OS points of the corresponding unbalanced classes. As of now, this is but a conjecture.

Bibliography

[1] M. Ahsanullah and V. B. Nevzorov. *Order Statistics: Examples and Exercises.* Nova Science Publishers, Inc, 2005.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning.* Academic Press, 2009.

[3] C. L. Chang. Finding Prototypes for Nearest Neighbor Classifiers. In *IEEE Transactions on Computing*, volume 23, pages 1179–1184, 1974.

[4] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. In *Machine Learning*, pages 273–297, 1995.

[5] H. A. David. *Order Statitics.* John Wiley & Sons Inc., 1981.

[6] P. A. Devijver and J. Kittler. On the Edited Nearest Neighbor Rule. In *Fifth International Conference on Pattern Recognition*, pages 72–80, December 1980.

[7] Luc Devroye. *Non-Uniform Random Variate Generation.* Springer-Verlag, New York, 1986.

[8] W. Duch. Similarity Based Methods: A General Framework for Classification, Approximation and Association. *Control and Cybernetics*, 29(4):937–968, 2000.

[9] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley-Interscience Publication, 2006.

[10] R. O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* A Wiley Interscience Publication, 2000.

[11] John W. Eaton. GNU Octave, 1994.

[12] R.A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annual Eugenics*, 7:179–188, 1936.

[13] G. M. Foody. Issues in Training Set Selection and Refinement for Classification by a Feedforward Neural Network. In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, pages 409–411, 1998.

[14] G. M. Foody. The Significance of Border Training Patterns in Classification by a Feedforward Neural Network using Back Propogation Learning. *International Journal of Remote Sensing*, 20(18):3549–3562, 1999.

[15] A. Frank and A. Asuncion. UCI Machine Learning Repository `http://archive.ics.uci.edu/ml` as of April 18, 2013, 2010.

[16] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, San Diego, second edition, 1990.

[17] S. Garcia, J. Derrac, J. Ramon Cano, and F. Herrera. Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):417–435, 2012.

[18] G. W. Gates. The Reduced Nearest Neighbor Rule. In *IEEE Transactions on Information Theory*, volume 18, pages 431–433, 1972.

[19] Z. Grudzien and D. Szynal. Characterizations of Distributions by Moments of Order Statistics when the Sample Size is Random. *Applications Mathematicae*, 23:305–318, 1995.

[20] P. E. Hart. The Condensed Nearest Neighbor Rule. In *IEEE Transactions on Information Theory*, volume 14, pages 515–516, 1968.

[21] `http://sci2s.ugr.es/pr/` as of April 18, 2013.

[22] `http://www.cis.hut.fi/research/som_lvq_pak.shtml` as of April 18, 2013.

[23] `http://www.data-compression.com/vq.shtml` as of April 18, 2013.

[24] `http://www.is.umk.pl/projects/datasets.html` as of April 18, 2013.

[25] Asha Gowda Karegowda, M.A.Jayaram, and A.S.Manjunath. Cascading K-means Clustering and k-Nearest Neighbor Classifier for Categorization of Diabetic Patients. *International Journal of Engineering and Advanced Technonlogy*, 01:147–151, 2012.

[26] S. Kim and B. J. Oommen. A brief Taxonomy and Ranking of Creative Prototype Reduction Schemes. *Pattern Analysis and Applications*, 6:232–244, 2003.

[27] S. Kim and B. J. Oommen. Enhancing Prototype Reduction Schemes with LVQ3-type Algorithms. *Pattern Recognition - The Journal of the Pattern Recognition Society*, 36:1083–1093, 2003.

[28] S. Kim and B. J. Oommen. On Using Prototype Reduction Schemes and Classifier Fusion Strategies to Optimize Kernel-Based Nonlinear Subspace Methods. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 27, pages 455–460, 2005.

[29] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 1995.

[30] P. R. Krishnaih and M. Haseeb Rizvi. A Note on Moments of Gamma Order Statistics. *Technometrics*, 9:315–318, 1967.

[31] L. I. Kuncheva. *Combining Pattern Classifiers - Methods and Algorithms*. John Wiley & sons, INC., Publication, 2004.

[32] L. I. Kuncheva, J. C. Bezdek, and R. P. W. Duin. Decision Templates for Multiple Classifier Fusion: An Experimental Comparison. *Pattern Recognition - The Journal of the Pattern Recognition Society*, 34:299–314, 2001.

[33] G. Li, N. Japkowicz, T. J. Stocki, and R. K. Ungar. Full Border Identification for Reduction of Training Sets. In *Proceedings of the Canadian Society for Computational Studies of Intelligence, 21st Conference on Advances in Artificial Intelligence*, pages 203–215, 2008.

[34] G. D. Lin. Characterizations of Continuous Distributions via Expected Values of Two Functions of Order Statistics. *Sankhya: The Indian Journal of Statistics*, 52:84–90, 1990.

[35] O. L. Mangasarian and W. H. Wolberg. Cancer Diagnosis via Linear Programming. *SIAM News*, 23(5):1 & 18, 1990.

[36] K. W. Morris and D. Szynal. A Goodness-of-fit for the Uniform Distribution Based on a Characterization. *Journal of Mathematical Science*, 106:2719–2724, 2001.

[37] S. Nadarajah. Explicit Expressions for Moments of Order Statistics. *Statistics and Probability Letters*, 78:196–205, 2008.

[38] B. J. Oommen and A. Thomas. Optimal Order Statistics-based "Anti-Bayesian" Parametric Pattern Classification for the Exponential Family. *Pattern Recognition*, 2013. Accepted for Publication.

[39] Y. Peng, G. Kou, and Y. Shi. Knowledge-Rich Data Mining in Financial Risk Detection. In *Computational Science*, pages 534–542, 2009.

[40] G. L. Ritter, H. B. Woodruff, S. R. Lowry, and T. L. Isenhour. An Algorithm for a Selective Nearest Neighbor Rule. In *IEEE Transactions on Information Theory*, volume 21, pages 665–669, 1975.

[41] D. Ruta and B. Gabrys. An Overview of Classifier Fusion Methods. *Computing and Information Systems*, 7:1–10, 2000.

[42] Gouda I. Salama, M.B.Abdelhalim, and Magdy Abd elghany Zeid. Breast Cancer Diagnosis on Three Different Datasets using Multi-classifiers. *International Journal of Computer and Information Technology*, 01:36–43, 2012.

[43] Pandu R. Tadikamalla. An Approximation to the Moments and the Percentiles of Gamma Order Statistics. *Sankhya: The Indian Journal of Statistics*, 39:372–381, 1977.

[44] T. G. Tape. *Interpreting Diagnostic Tests.* (This item can be found at `http://gim.unmc.edu/dxtests/ROC1.htm` as of April 18, 2013).

[45] S. Theodoridis and K. Koutroumbas. *Pattern Recognition.* Springer, 2007.

[46] A. Thomas and B. J. Oommen. Optimal "Anti-Bayesian" Parametric Pattern Classification for the Exponential Family Using Order Statistics Criteria. In *Image Analysis and Recognition*, volume 7324 of *Lecture Notes in Computer Science*, pages 11–18. Springer Berlin / Heidelberg, 2012.

[47] A. Thomas and B. J. Oommen. Optimal "Anti-Bayesian" Parametric Pattern Classification Using Order Statistics Criteria. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 7441 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin / Heidelberg, 2012. This was a Plenary/Keynote Talk at the Conference.

[48] A. Thomas and B. J. Oommen. Order Statistics-based "Anti-Bayesian" Parametric Classification for Asymmetric Distributions in the Exponential Family. In *International Conference on Machine Learnnig and Pattern Recognition*, volume 72 of *WASET 2012*, pages 187–195, 2012.

[49] A. Thomas and B. J. Oommen. Classification of Multi-dimensional Distributions using Order Statistics Criteria. In *Conference on Computer Recognition Systems*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2013. Accepted for Publication.

[50] A. Thomas and B. J. Oommen. Order Statistics-based Parametric Classification for Multi-dimensional Distributions. 2013. Submitted for Publication.

[51] A. Thomas and B. J. Oommen. The Fundamental Theory of Optimal "Anti-Bayesian" Parametric Pattern Classification Using Order Statistics Criteria. *Pattern Recognition*, 46:376–388, 2013.

[52] I. Tomek. Two Modifcations of CNN. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-6(6):769 – 772, Nov. 1976.

[53] Y. Too and G. D. Lin. Characterizations of Uniform and Exponential Distributions. *Academia Sinica*, 7(5):357–359, 1989.

[54] I. Triguero, J. Derrac, S. Garcia, and F. Herrera. A Taxonomy and Experimental Study on Prototype Generation for Nearest Neighbor Classification. *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, 42:86–100, 2012.

[55] Vladimir Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

[56] Q. Xie, C.A. Laszlo, and R. K. Ward. Vector Quantization Techniques for Nonparametric Classifier Design. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-15(12):1326 – 1330, Dec. 1993.

[57] L. Xu, A. Krzyzak, and C. Y. Seun. Methods of Combining Multiple Classifiers and their Application to Handwriting Recognition. In *IEEE Transactions on Systems, Man and Cybernetics*, volume 22, pages 418–435, 1992.

[58] D. H. Young. Moment Relations for Order Statistics of the Standardized Gamma Distribution and the Inverse Multinomial Distribution. *Biometrika*, 58:637–640, 1971.

[59] D. Zhang and G. Lu. Segmentation of Moving Objects in Image Sequence: A Review. Technical report, Monash University, Australia, 2001.