Learning From Data Locally and Globally

Huang, Kaizhu

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy
in
Department of Computer Science & Engineering

July, 2004

Learning From Data Locally and Globally

submitted by

Huang, Kaizhu

for the degree of Doctor of Philosophy
at the Chinese University of Hong Kong

Abstract

I mainly consider the task of learning classifiers from data in this thesis. In this context, I propose a common framework that combines two different and important paradigms in machine learning: global learning and local learning. Traditional global learning approaches focus on describing phenomena by attempting to estimate a distribution from data. Based on the estimated distribution, the global learning methods can then perform inferences, conduct marginlizations, and make predictions. Although enjoying a long and distinguished history and containing many good features, e.g., a relatively simple optimization, and the flexibility in incorporating global information such as structure information and invariance etc, these learning approaches usually have to assume a specific type of distribution a prior. Therefore, they are widely argued for lacking the generality. On the other hand, local learning methods do not estimate a distribution from data. Instead, they focus on extracting only the local information, which is directly related to the learning task, i.e., the classification in this thesis. Recent progress following this trend has demonstrated that local learning approaches, e.g., Support Vector Machines (SVM), outperform the global learning methods in many aspects.

Despite of the success, local learning approaches actually discard plenty of important global information on data, e.g., the structure information. Therefore, this restricts the learning performance of this types of learning schemes.

In this thesis, I thus develop a hybrid model named Maxi-Min Margin Machine (M^4), which successfully combines two largely differently but complementary paradigms. Within this new framework, I propose a hybrid model named Maxi-Min Margin Machine (M^4). This model is demonstrated to contain both appealing features in global learning and local learning. It not only captures the global structure information from data, but it also provides a task-oriented scheme for the learning purpose and inherits the superior performance from local learning. As a major contribution, M^4 successfully unifies many important learning models, including Support Vector Machines, Minimax Probability Machine (MPM), and Fisher Discriminant Analysis. Another compelling feature of M^4 is that it can be cast as a Sequential Second Order Cone Programming problem, yielding a polynomial time complexity.

In addition, directly motivated from the Maxi-Min Margin Machine, I also develop a regression model named Local Support Vector Regression (LSVR). LSVR is demonstrated to provide a systematic and automatic scheme to locally and flexibly adapt the margin, which is globally fixed in the standard Support Vector Regression (SVR), a state-of-the-art regression model. Therefore, it can tolerate the noise adaptively. The proposed LSVR is promising in the sense that it not only adequately considers the local information of the data in approximating functions, but more importantly, it includes special cases, which enjoy a physical meaning very much similar to the standard SVR. Both theoretical and empirical investigations demonstrate the advantages of this new model.

Another important contribution of this thesis is that I also develop a novel global learning model called Minimum Error Minimax Probability Machine (MEMPM). Although still within the framework of global learning, this model

does not need to assume any specific distribution beforehand and represents a distribution-free Bayes optimal classifier in a worst-case scenario. This thus makes the model distinguished from the traditional global learning models, especially the traditional Bayes optimal classifier. One promising feature of MEMPM is that it can derive an explicit accuracy bound under a mild condition, leading to a good generalization performance for future data.

The fourth critical contribution of this thesis is the development of the Biased Minimax Probability Machine (BMPM) model. In spite of the fact that it is a special case of MEMPM, I make this model distinguished because BMPM provides the first systematic and rigorous approach for a kind of important learning tasks, namely, the biased learning or imbalanced learning. Different from traditional imbalanced (biased) learning methods, BMPM can quantitatively and explicitly incorporate a bias for one class and consequently emphasizes the more important classes. A series of experiments demonstrate that BMPM is very promising in imbalanced learning and medical diagnosis.

摘要

本論文提出了一種機器學習的新思路，即局部且全局的學習方法。這種新的學習框架成功的組合了傳統的僅僅基於局部或全局的學習方法。在此新的框架下，我們於是提出了一個組合模型，最大最小邊界機 (Maxi-Min Margin Machine 簡稱 M^4) 。此模型成功的統一了機器學習領域眾多重要的模型，包括現今最為流行和成功的支持向量機 (Support Vector Machine 簡稱 SVM) ，最大最小概率機 (Minimax Probability Machine 簡稱 MPM)，以及被廣泛應用的線性判別分析 (Fisher Discriminant Analysis 簡稱 FDA) 。 M^4 不僅具有局部學習的特點，即面向任務和較優的性能，而且它具備全局學習的眾多優點，比如能準確提取結構特徵和數據趨勢等。除此之外，我們提出了基於二次錐規劃的優化方法，因而此模型可在多項式時間內優化，從而可被廣泛的應用與實際中。

此外，由 M^4 得到啟發，我們提出了一種新的回歸模型，局部支持向量回歸模型(Local Support Vector Regression 簡稱 LSVR) 。此模型提供了一種系統而且自動的根據數據局部趨勢來調節邊界的方法。而傳統的支持向量回歸模型則缺乏這種自動調節的功能。因而，我們這種新模型能更好的，更靈活的處理數據噪聲。LSVR 在理論上也非常有意義，它的一些特例具有和傳統的支持向量回歸模型相同的物理意義。我們在實際中的考查也顯示，和傳統支持向量回歸模型相比，這種模型具有一定的優勢。

本論文另外一個重要的貢獻在於我們也提出了一種全新的全局學習方法，叫作最小錯誤最大最小概率機(Minimum Error Minimax

Probability Machine 簡稱 MEMPM）。儘管此模型仍然屬於全局學習方法，它不需要事先假設概率分佈模型。因此，比較而言，此模型更具有普適性。此外，作為一個非常重要的貢獻，此模型代表了一個最壞情況下無分佈 Bayes 最優分類器。此模型的另外一個優點是它能夠導出一個顯式的最小錯誤率，因而保證了滿意的歸納性能。

本論文的第四個貢獻在於我們同時也提出了一個偏差最大最小概率機模型 (Biased Minimax Probability Machine 簡稱 BMPM)。雖然此模型是最小錯誤最大最小概率機的一個特例，我們仍然突出此模型，是在於它能夠系統的，嚴格的，定量的，處理一大類分類問題，即不平衡和偏差分類問題。比較而言，傳統的方法大多採用手動的，實際的，定性的方法，因而具有不嚴格的特點。具體的理論和實驗分析表明我們這種模型對於不平衡分類問題有著很好的效果。

Acknowledgment

There are many persons I would like to thank. First and foremost, I want to thank my supervisors, Prof. Irwin King and Prof. Michael R. Lyu. I gain too much from their guidance in both the attitude in doing research and the detailed technique things in conducting my research work. I would like to express my sincere gratitude and appreciation to their supervision, encouragement, and support at all levels. I also extend warm thanks to Prof. Laiwan Chan, from whom, I have received many valuable suggestions in the beginning of my Ph.D study. Moreover, I also thank her for providing some comments for our novel Minimum Error Minimax Probability Machine. I will always be grateful for the outstanding research environment fostered by our department, and also for so many related work done by our clerical staffs.

I extend my thanks to Prof. Shuzhong Zhang for the conservations and discussions on the Conic Programming problems. These communications have profoundly shaped the ideas in this thesis as well as benefited formalizing them into mathematical expressions. I also want to thank Prof. Chenzhen Sun for sharing with me his understanding of many aspects of the research.

I would like to thank my colleagues and my friends. I thank Haiqin Yang for his effort and constructive discussions in conducting the research work in this thesis. I also want to thank my office-mates, Hui Chen, Xinyu Chen, Jifu He, Guangyu Wang, Liang Wan. Their broad knowledge, love of life, sense of humor made me feel relaxed and happy when working in the office. I extend my gratitude to Xia Cai, Gary Chan, Steven Hoi, Xiaoqi Li, Yi Liu, Shi Lu, Gary Ng, Richard Sia, Hackker Wong, Haixuan Yang, Wan Zhang, and Yangfan Zhou for their help and discussion in many aspects of my research work.

Finally, I want to thank my family and my girlfriend, Jun Hu. Without their deep love and constant support, this thesis could not have been completed.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The objective of this thesis is to establish a framework which combines two different paradigms in machine learning: global learning and local learning. The deriving combined model demonstrates that a hybrid learning of these two different schools of approaches can outperform each isolated approach both theoretically and empirically. Global learning focuses on describing a phenomenon or modeling data in a global way. For example, a distribution over the variables is usually estimated for summarizing the data. Its output can usually reconstruct the data. This school of approaches, including Bayesian Networks [46, 54, 148], Gaussian Mixture Models [13, 103], and Hidden Markov Models [8, 126], has a long and distinguished history, which has been extensively applied in artificial intelligence [131], pattern recognition [47], computer vision [45], etc. On the other hand, local learning does not intend to summarize a phenomenon, but builds learning systems by concentrating on some local parts of data. It lacks the flexibility yet surprisingly demonstrates superior performance to global learning according to recent researches [17, 70, 132]. In this thesis, a bridge has been established between these two different paradigms. Moreover, the resulting principled framework subsumes several important models, which respectively locate themselves into the global learning paradigm and the local learning paradigm.

In this chapter, we address the motivations of the two different learning

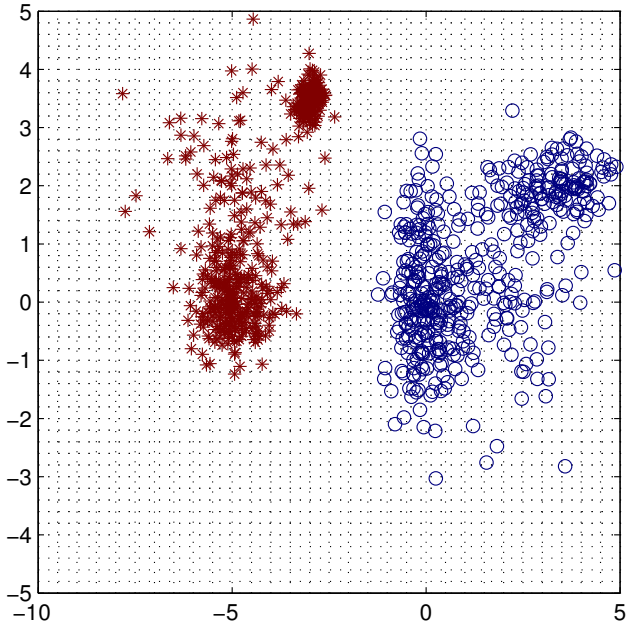


Figure 1.1: Two classes of two-dimensional data.

frameworks. As a summary, we present the objectives of this paper and outline the contributions. Finally, we provide an overview of the rest of this thesis.

1.1 Learning and Global Modeling

When studying real world phenomena, scientists are always wondering whether some underlying laws or nice mathematic formulae exist for governing these complex phenomena. Moreover, in practice, due to incomplete information, the phenomena are usually nondeterministic. This motivates to base probabilistic or statistical models to perform a global investigation on sampled data from the phenomena. A common way for achieving this goal is to fit a density on the observations of data. With the learned density, people can then incorporate prior knowledge, conduct predictions, and perform inferences and marginalizations. One main category in the framework of global learning is the so-called generative learning. By assuming a specific mathematic model on the observations of data, e.g., a Gaussian distribution, the phenomena can

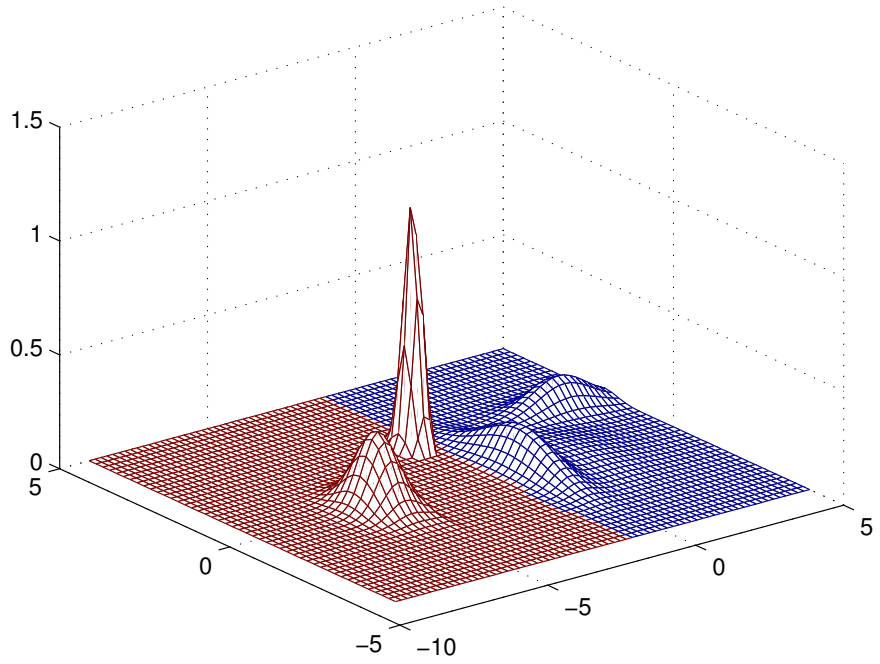


Figure 1.2: An illustration of distribution-based classifications (also known as the Bayes optimal decision theory). Two Gaussian mixtures are engaged to model the distribution of two classes of data respectively. The distribution can then be used to construct the decision plane.

therefore be described or re-generated. Figure 1.1 illustrates such an example. In this figure, two classes of data are plotted as $*$'s for the first class and $\circ$'s for the other class. The data can thus be modeled as two different mixtures of Gaussian distributions as illustrated in Figure 1.2. By knowing only the parameters of these distributions, one can then summarize the phenomena. Furthermore, one can clearly employ this information to distinguish one class of data from the other class or simply know how to separate two classes. This is also well-known as Bayes optimal decision problems [51, 38].

In the development of learning approaches within the community of machine learning, there has been a migration from the early rule-based methods [50, 156] wanting more involvement of domain experts, to widely-used probabilistic global models mainly driven by data itself [24, 47, 62, 72, 117, 163].

However, one question for most probabilistic global models is what kind of global models, or more specifically, which type of densities should be specified beforehand for summarizing the phenomena. For some tasks, this can be prescribed by a slight introduction of domain knowledge from experts. Unfortunately, due to both the increasing sophistication of real world learning tasks and active interactions among different subjects of research, it is more and more difficult to obtain fast and valuable suggestions from experts. A further question is thus proposed, i.e., what is the next stop in the community of machine learning, after experiencing a migration from rule-based models to probabilistic global models? Recent progress in machine learning seems to imply a local learning as a solution.

1.2 Learning and Local Modeling

Global modeling addresses describing phenomena, no matter whether the summarized information from the observations is applicable to specific tasks or not. Moreover, the hidden principle under global learning is that information can be accurately extracted from data. On the other hand, local learning [49, 138, 141], which recently attracts active attentions in the machine learning community, usually regards that a general and accurate global learning is an impossible mission. Therefore, local learning focuses on capturing only local yet useful information from data. Furthermore, recent research progress and empirical study demonstrates that this much different learning paradigm is superior to global learning in many facets.

In further details, instead of globally modeling data, local learning is more task-oriented. It does not aim to estimate a density from data as in global learning, which is usually an intermediate step for many tasks such as pattern recognitions (note that the distribution or density obtained by global learning actually is not directly related to the classification itself); it also does not

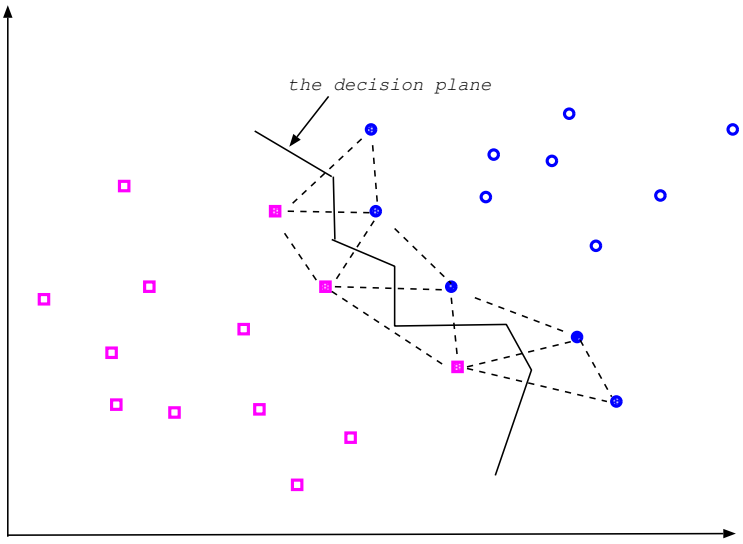


Figure 1.3: An illustration of local learning (also known as the Gabriel Graph classification). The decision boundary is just determined by some local points indicated as filled points.

intend to build an accurate model to fit the observations of data globally. Differently, it only extracts useful information from data and directly optimizes the learning goal. For example, when used in learning classifiers from data, only those observations of data around the separating plane need to be accurate, while inaccurate modeling over other data is certainly acceptable for the classification purpose. Figure 1.3 illustrates such an problem. In this figure, the decision boundary is constructed only based on those filled points, while other points make no contributions to the classification plane (the decision boundary is given based on the Gabriel Graph method [6, 73, 164]).

However, although containing promising performance, local learning appears to locate itself at another extreme end to global learning. Employing only local information may lose the view of data. Consequently, sometimes, it cannot grasp the data trend, which is critical for guaranteeing better performance for future data. This can be seen in the example as illustrated in Figure 1.4. In this figure, the decision boundary (also constructed by the Gabriel Graph classification) is still determined by some local points indicated

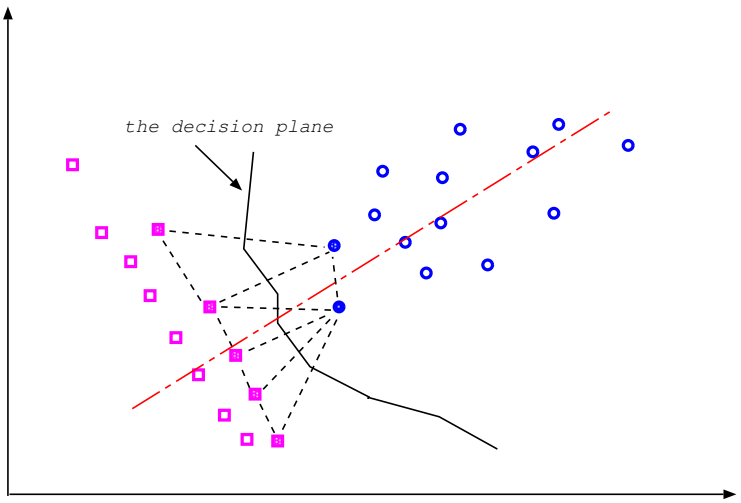


Figure 1.4: An illustration on that local learning cannot grasp data trend. The decision boundary (constructed by the Gabriel Graph classification) is determined by some local points indicated as filled points. It, however, loses the data trend. The decision plane should be obviously closer to the filled squares rather than locating itself in the middle of filled □'s and ○'s.

as filled points. Clearly, this boundary does not grasp the data trend. More specifically, the class associated with ○'s is obviously more scattered than the class associated with □'s in the axis indicated as dashed red line. Therefore, a more promising decision boundary should lie closer to filled □'s than those filled ○'s instead of lying midway between filled points. A similar example can also be seen in Chapter 2 on a more principled local learning model, i.e., the current state-of-the-art classifier, Support Vector Machines (SVM) [153]. Targeting this problem, we then suggest a hybrid learning in this thesis.

1.3 Hybrid Learning

There are complementary advantages for both local learning and global learning. Global learning summarizes data and provides practitioners with knowledge on the structure, independence, trend of data etc, since with the precise modeling of phenomena, the observations can be accurately regenerated and

therefore can be studied or analyzed thoroughly. However, this also presents difficulties in how to choose a valid model to describe all the information. In comparison, local learning directly employs part of information, critical for the specific oriented tasks, and does not assume models to re-synthesize/restore the whole road-map of data. Although demonstrated to be superior to global learning in many facets of machine learning, it may lose some important global information. The question here is thus, can reliable global information, independent of specific model assumptions, be combined into local learning? This question clearly motivates a hybrid learning of two largely different schools of approaches, which is also the objective of this thesis.

1.4 Contributions

In this thesis, we aim to propose a hybrid learning scheme to combine two different paradigms, namely global learning and local learning. Within this scheme, we propose a hybrid model, named the Maxi-Min Margin Machine (M^4), demonstrated to contain both merits of global learning in representing data and the advantages of local learning in handling tasks directly and effectively. Moreover, adopting the viewpoint of local learning, we also develop a global learning model, called the Minimum Error Minimax Probability Machine (MEMPM), which does not assume specific distributions on data and thus distinguishes itself from traditional global learning approaches. The main contributions of this thesis are further described as follows in detail.

- *Proposed the Maxi-Min Margin Machine model, a hybrid learning framework successfully combining global learning and local learning*

  - ◇ *A unified framework of many important models*

    As will be demonstrated, our proposed hybrid model successfully unifies both important models in local learning, e.g., the Support

Vector Machines [17], and significant models in global learning, such as the Minimax Probability Machine (MPM) [85] and the Fisher Discriminant Analysis (FDA) [47].

◇ *With the generalization Guarantee*

Various statements from many views such as the sparsity and Marshall and Olkin Theory [101, 121] will be presented for providing the generalization bound for the combined approach.

◇ *A sequential Conic Programming solving method*

Besides the theoretic advantages of the proposed hybrid learning, we also tailor a sequential Conic Programming method [124, 144] to solve the corresponding optimization problem. The computational cost is shown to be polynomial and thus the proposed M^4 model can be solved practically.

- *Developed a general global learning model, the Minimum Error Minimax Probability Machine*

  ◇ *A worst-case distribution-free Bayes optimal classifier*

  Different from traditional Bayes optimal classifiers, MEMPM does not assume distributions for the data. Starting with the Marshall and Olkin theory, this model attempts to model data under the minimax schemes. It does not intend to extract exact information but the worst-case information from data and thus presents an important progress in global learning.

  ◇ *Derived an explicit error bound for future data*

  Inheriting the advantages of global learning, the proposed general global learning method contains an explicit worst-case error bound for future data under a mild condition. Moreover, the experimental results suggest that this bound is reliable and accurate.

- ◇ *Proposed a sequential Fractional Programming optimization*

  We have proposed a Fractional Programming optimization method for the MEMPM model. In each iteration, the optimization is shown to be a pseudo-concave problem, which thus guarantees that each local solution will be the global solution in this step.

- ◇ *Implemented a Matlab toolbox for this general global learning method*

  We have released a Matlab toolbox for the novel global model. Detailed comments, demos, and examples are provided for its easy usage [160, 161].

- *Developed a global learning method called Biased Minimax Probability Machine (BMPM) for biased or imbalanced learning*

  - ◇ *Presented a rigorous and systematic treatment for biased learning tasks*

    Although being a special case of our proposed general global learning model, MEMPM, this model provides a quantitative and rigorous approach for biased learning tasks, where one class of data are always more important than the other class. Importantly, with explicitly controlling the accuracy of one class, this branch model can precisely impose biases on the important class.

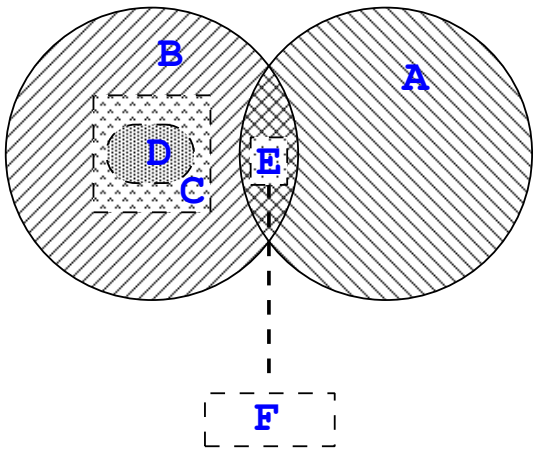
  - ◇ *Containing explicit generalization bounds for both classes of data*

    Inheriting the good feature of the MEMPM model, this model also contains explicit generalization bounds for both classes of data. This therefore guarantees a good prediction accuracy for future data.

- *Developed a novel regression model Local Support Vector Regression (LSVR)*

  - ◇ *Provided a systematic and automatic treatment in adapting margins*

    Motivated from M^4, LSVR focuses on considering the margin setting



A: Local Learning
B: Global Learning
C: Minimum Error Minmax Probability Machine
D: Biased Minimax Probability Machine
E: Maxi-Min Margin Machine
F: Local Support Vector Regression

Figure 1.5: The relation among the developed models in this thesis

locally. When compared to the regression model of SVM, i.e., the Support Vector Regression (SVR), this novel regression model is shown to be more robust with respect to the noise of data in that it contains the volatile margin setting.

◇ *Incorporated special cases very much similar to the standard SVR*
When considering a consistent trend for all data points, the LSVR can derive special cases very much similar to the standard SVR. We further demonstrate that in a meaningful assumption, the standard SVR is actually the special case of our LSVR model.

In a summary, the relation among our developed models is described in Figure 1.5.

1.5 Scope

This thesis states and refers to the learning first as statistical learning, which appears to be the current main trend of learning approaches. We then further restrict the learning in the framework of classification, one of the main problems in machine learning. The corresponding discussion on different models including the conducted analysis of the computational and statistical aspects of machine learning are all subject to the classification tasks. Nevertheless, we will also extend the content of this thesis to regression problems, although it is not the focus of this thesis.

1.6 Thesis Organization

The rest of this thesis is organized as follows:

- Chapter 2

  We will review different learning paradigms in this chapter. We will establish a hierarchy graph attempting to categorize various models in the framework of local learning and global learning. We will then base this graph to describe and discuss these models. Finally, we motivate the Minimum Error Minimax Probability Machine and the Maxi-Min Margin Machine.

- Chapter 3

  We will develop a novel global learning model, called the Mininum Error Minimax Probability Machine. We will demonstrate how this new model represents the worst-case Bayes optimal classifier. We will detail its model definition, provide interpretations, establish a robust version, extend to nonlinear classifications, and present a series of experiments to demonstrate the advantages of this model.

- Chapter 4

  We will present the Maxi-Min Margin Machine, which successfully combines two different but complementary learning paradigms, i.e., local learning and global learning. We will show how this model incorporates the Support Vector Machine, the Minimax Probability Machine, and the Fisher Discriminant Analysis as special cases. We will also demonstrate the advantages of Maxi-Min Margin Machine by providing theoretical, geometrical, and empirical investigations.

- Chapter 5

  An extension of the proposed MEMPM model will be discussed in this chapter. More specifically, the Biased Minimum Minimax Probability Machine will be discussed and applied into the imbalanced learning tasks. We will review different criteria for evaluating imbalanced learning approaches. We will then base these criteria to tailor BMPM into this type of learning. Both illustrations on toy data sets and evaluations on real world imbalanced and medical data sets will be provided in this chapter.

- Chapter 6

  A novel regression model called the Local Support Vector Regression, which can be regarded as an extension from the Maxi-Min Margin Machine, will be introduced in detail in this chapter. We will show that our model can vary the tube (margin) systematically and automatically according to the local data trend. We will show that this novel regression model is more robust with respect to the noise of data. Empirical evaluations on both synthetic data and real financial time series data will be presented to demonstrate the merits of our model with respect to the standard Support Vector Regression.

- Chapter 7

  We will then summarize this thesis and conduct discussions on future

work.

We try to make each of these chapters self-contained. Therefore, in several chapters, some critical contents, e.g., model definitions or illustrative figures, having appeared in previous chapters, may be briefly reiterated.

Chapter 2

Global Learning Vs. Local Learning — A Background Review

In this chapter, we conduct a more detailed and more formal review on two different schools of learning approaches, namely, the global learning and local learning. We first provide a hierarchy graph as illustrated in Figure 2.1 in which we try to classify many statistical models into their proper categories, either global learning or local learning. Our review will also be conducted based on this hierarchy structure. To make it clear, we use filled shapes to highlight our own work in the graph.

Global learning fits a distribution over data. If a specific mathematic model, e.g., a Gaussian model, is assumed on the distribution, this is often called generative learning, whose name implies that the mathematic formulation of the assumed model governs the generation of data in the learning task. To learn the parameters from the observations of data for the specific model, several schemes have been proposed. This includes Maximum Likelihood (ML) learning, which is easy to conduct but is less accurate, Conditional Likelihood (CL) learning, which is usually hard to perform optimization but is more effective, and Bayesian Average (BA) learning, which has a comparatively short

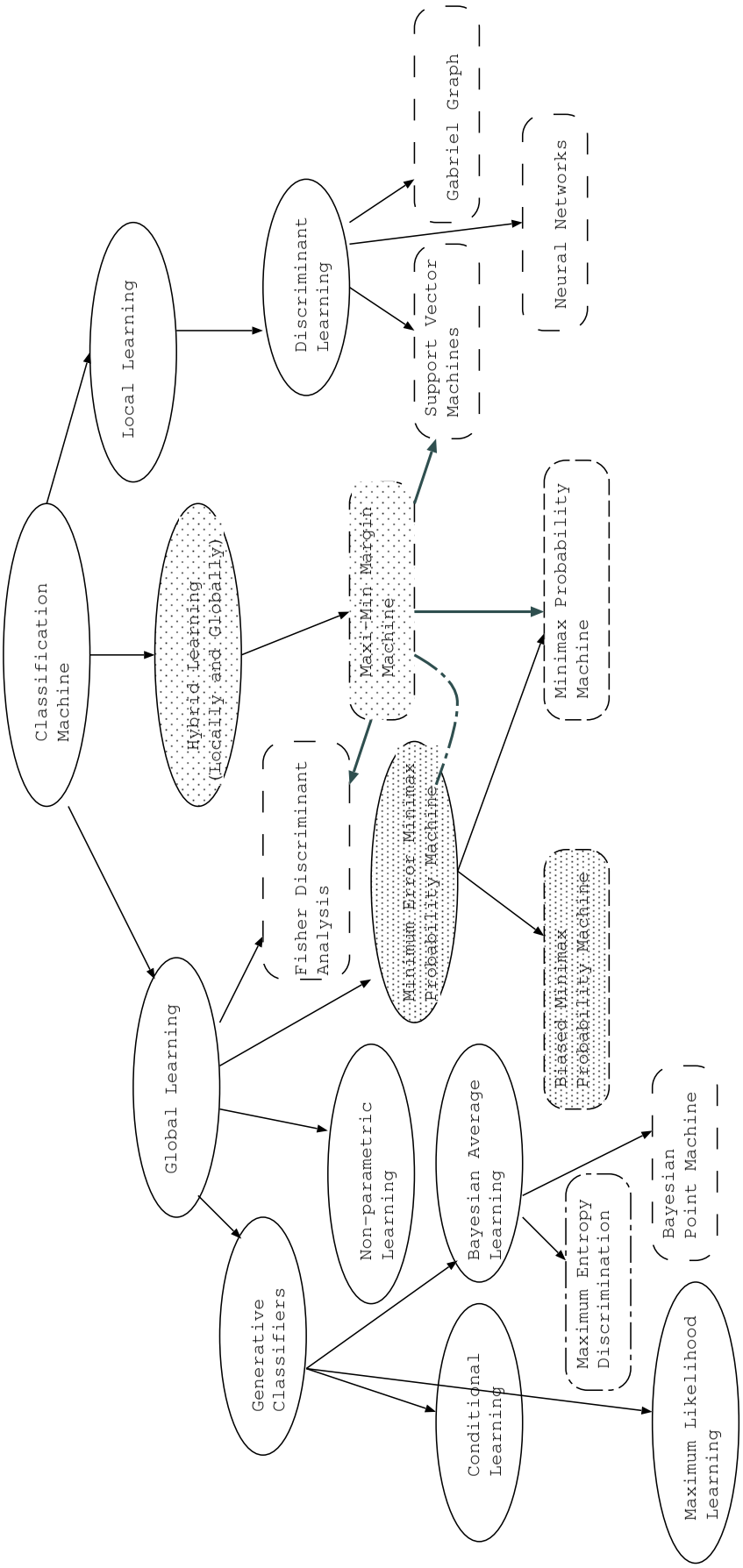


Figure 2.1: A hierarchy graph of statistical learning models

history but is more promising. As generative learning pre-assigns a specific model before learning, it often lacks the generality and thus may be invalid in many cases. This thus motivates the non-parametric learning, which still estimates a distribution on data but assumes no specific mathematic generative models. The common way in this type of learning is to locally fit over each observation a simple density and then sums all the local densities as the final distribution for data. Although in some circumstances, this approach is successful, it is criticized for requiring a huge quantity of training points and containing a large space complexity. Differently, in this thesis, we will demonstrate a novel global learning method, named Minimum Error Minimax Probability Machine (MEMPM). Although still in the framework of global learning, it does not belong to non-parametric learning, therefore requiring no extremely heavy storage spaces. Moreover, it does not assume any specific distribution on data, which hence distinguishes itself from the traditional global generative learning. As a critical contribution, MEMPM represents a distribution-free Bayes optimal classifier in a worst-case scenario. Furthermore, we will show that this model incorporate two important global learning approaches, Biased Minimax Probability Machine (BMPM) and Minimax Probability Machine (MPM) [85, 86]. Since all approaches within the paradigm of global learning requires summarizing the data information completely and globally, it thus may waste computational resources and is widely argued to be less direct. This motivates the local learning, which makes no attempt to model the data globally, but focuses on extracting only those information directly related to the task. This type of learning is often refereed to as discriminative learning in the context of classifications. One famous model among them is Support Vector Machine (SVM). With the task-oriented, robust, computationally tractable properties, SVM has achieved a great success and is considered as the current state-of-the-art classifier. Although local learning demonstrates superior performance to traditional global learning, it appears to situate itself at another

extreme end, which totally discards the useful global information, e.g., the structure information of data.

Our suggestion is that we should combine these two different but complementary paradigms. Towards this end, we then propose a new model called Maxi-Min Margin Machine (M^4), which not only successfully employs the global structure information from data but also holds merits of local learning such as robustness and superior classification accuracies. As a critical contribution, M^4, the hybrid learning model, represents a general model, successfully shown to contain both local learning models and global learning models as special cases. More specifically, it contains two significant and popular global learning models, i.e., Fisher Discriminant Analysis (FDA) [47] and Minimax Probability Machine [84, 85, 86] as special cases. Meanwhile, SVM, the local learning model can also be considered as one of its branches. In addition, M^4 also demonstrates a strong connection with MEMPM, the novel general global learning model.

In the following, we first present the problem definition, which will be used throughout this thesis. We then base Figure 2.1 to provide introductions and comments for each type of learning model sequently. Finally, we summarize the review and conclude with the proposition of the hybrid framework, the objective of this thesis.

2.1 Problem Definition

Given a data set D consisting of N observations, where each observation is of the form $(z_1, z_2, \ldots, z_n, c)$ ($z_i \in \mathbb{R}$, for $1 \leq i \leq n$, $c \in \mathbb{F}$, where $\mathbb{F}$ is a finite set), the basic learning problem is to construct a mapping rule or a function f from $\{z_1, z_2, \ldots, z_n\}$ called features or attributes to the output c, denoted as the class variable, namely $f(z_1, z_2, \ldots, z_n, \Theta, D) \rightarrow c$, where Θ means the function parameters. The function f should be not only as accurate

as possible to fit the observations D, but also can robustly predict the class for the new data. Sometimes, we also use Θ to denote the mapping model f and its associated parameters. For simplicity, we often use $\mathbf{z}$ to denote the n-dimensional variable $\{z_1, z_2, \ldots, z_n\}$. If we use $\mathbf{z}^j$, we refer it to the j-th observation in D. Throughout this thesis, unless we provide statements explicitly, bold typeface will indicate a vector or matrix, while normal typeface will refer to a scale variable or the component of the vectors.

2.2 Global Learning

Global learning often describes the data by attempting to estimate a distribution over variables $(z_1, z_2, \ldots, z_n, c)$, denoted as $p(\mathbf{z}, c, \Theta|D)$. The estimated distribution can then be used to make predictions by calculating the probability that a specific value of c will occur, when given an instance of features $\mathbf{z}$. In more details, the decision rule or the mapping function can be described as:

$$c = \arg\max_{c_k \in \mathbb{F}} p(c_k|D, \mathbf{z}) = \arg\max_{c_k \in \mathbb{F}} \int p(c_k, \Theta|D, \mathbf{z}) d\Theta. \tag{2.1}$$

By employing Bayes theory, one can transform the above joint probability (the item inside the integral) into the following equivalent forms:

$$p(c_k, \Theta|D, \mathbf{z}) = \frac{p(c_k, \mathbf{z}|D, \Theta)p(\Theta|D)}{\sum_{c_k \in \mathbb{F}} \int p(c_k, \mathbf{z}|D, \Theta)p(\Theta|D)d\Theta}. \tag{2.2}$$

Since the denominator in the above does not influence the decision in practice, the decision rule of (2.1) can be written into a relatively easily-calculated form:

$$c = \arg\max_{c_k \in \mathbb{F}} \int p(c_k, \mathbf{z}|D, \Theta)p(\Theta|D)d\Theta. \tag{2.3}$$

Depending on how the model Θ is assumed on D, global learning can be further divided into generative learning and non-parametric learning as

elaborated in the following subsections.

2.2.1 Generative Learning

Generative learning often assumes a specific model on data D. For example, a Gaussian distribution is assumed to be the underlying model to generate D. In this case, the parameters Θ refer to the mean and covariance for the Gaussian distribution. There are many models, which belong to this type of learning. Among them are Naive Bayes model [38, 78, 88], Gaussian Mixture Model [10, 51, 56, 103], Bayesian Network [59, 60, 61, 63, 87, 117], Hidden Markov Model [5, 143], Logistic Regression [71], Bayes Point Machine [58, 106, 130], Maximum Entropy Estimations [70] etc. The key problem for generative learning is how to learn the parameters Θ from data. Generally, in the literatures of machine learning, three schemes, Maximum Likelihood learning, Conditional Likelihood learning, and Bayesian Average learning, are engaged for estimating the parameters. We state these approaches one by one in the following.

Maximum Likelihood Learning & Maximum A Posterior Learning

Considering that it is not always easy to calculate the integral in (2.3), earlier researchers often try to compute some approximations of (2.3) instead. This motivates the Maximum Likelihood learning and Maximum A Posterior (MAP) learning [38, 117].

These learning methods replace (2.3) with the formulation below:

$$c = \arg\max_{c_k \in \mathbb{F}} p(c_k, \mathbf{z}|D, \Theta_*) \ , \tag{2.4}$$

In the above, how Θ_* are estimated thus discriminates MAP from ML. In

MAP, Θ_* are estimated as:

$$\Theta_* = \arg\max p(\Theta|D) \; ; \tag{2.5}$$

while in ML, the parameters are given as:

$$\Theta_* = \arg\max p(D|\Theta) \; . \tag{2.6}$$

Observing (2.3), one can see that MAP actually enforces the approximated conditional distribution over parameters as a delta function situating itself at the most prominent Θ. Namely,

$$\widehat{p}(\Theta|D) = \begin{cases} 1 & \text{if} \quad \Theta = \arg\max p(\Theta|D) \; ; \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

For ML, it is even simpler. This can be observed by looking into the relation between MAP and ML:

$$\arg\max p(\Theta|D) = \arg\max p(D|\Theta)p(\Theta). \tag{2.8}$$

Thus, compared to MAP, ML omits the item $p(\Theta)$, the prior probability over the parameters. In practice, a model with a more complex structure may be more possible to cause over-fitting, which means the model can fit the training data perfectly while having a bad prediction ability on the test or future data. In this sense, discarding the prior probability, ML lacks the flexibility to favor simple models by conditioning the prior probability [15, 150]. On the other hand, MAP permits a regularization on the prior probability and thus contains potentials to resist over-fitting problems.

When applied in practice, under independent, identically distributional data (i.i.d.) conditions, rather than directly optimizing the original form, ML estimations usually take the maximization on the log-likelihood , which can

transform the multiplication form into an easily-solved additional one:

$$\Theta_* = \arg\max p(D|\Theta) = \arg\max \log p(D|\Theta) = \arg\max \sum_{j=1}^{N} \log p(\mathbf{z}^j|\Theta). \quad (2.9)$$

Maximum Conditional Learning

Rather than computing the integral form, both the above ML learning and MAP learning seek to use one specific point Θ_* to calculate (2.3). The difference between them lies in how they estimate the specific parameter Θ_*. Compared with the long history in using ML and MAP estimations, Maximum Conditional learning enjoys a short span of time but has achieved state-of-the-art performance in many domains such as speech recognition [10, 127, 158].

Maximum Conditional learning also focuses on adopting one certain Θ_* to simplify the computation of (2.3). Differently, the selection of Θ_* is based on maximizing a conditional likelihood defined as follows:

$$\Theta_* = \arg\max p(\mathbb{C}|\Theta, \mathbb{Z}) \,, \quad (2.10)$$

where $\mathbb{C} = \{c^1, c^2, \ldots, c^N\}$ is the vector formed by the class label of each observation in D, and $\mathbb{Z} = \{\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^N\}$ corresponds to the data of the attributes (or features) part in D. Similar to the relation between ML and MAP, MC can also plug in a prior probability into the above formulae for resisting over-fitting problems, i.e.,

$$\Theta_* = \arg\max p(\mathbb{C}|\Theta, \mathbb{Z}) p(\Theta) \,, \quad (2.11)$$

By maximizing the conditional likelihood, MC is thus more direct and classification oriented. Note that only the conditional probability, which is maximized above, is directly related to the classification purpose. Maximizing

other quantities as done in ML or MAP, possibly optimizes unnecessary information for classifications, which is wasteful and imprecise. However, although MC appears to be more precise, it is usually hard to conduct the optimization due to the involvement of the conditional item. Such an example can be seen in optimizing a tree-based Bayesian Networks [46]. Moreover, when there is missing information, the optimization of MC may even present a more tough problem in general, while in such circumstances, powerful Expectation Maximization (EM) techniques [82, 105] can easily be applied in ML.

Bayesian Average Learning

It is noted that in ML, MAP and MC, for the easy calculation of (2.3) one certain Θ_* is adopted for approximations. However, although one point estimation enjoys computational advantages in approximating (2.3), in practice it may be very inaccurate and in this sense may impair the prediction ability of global learning. Aiming to solve this problem, recent researches have suggested to use the Bayesian Average learning approaches. This type of approaches facilitates the computation of (2.3) by changing the integral into a summation form based on sampling methods, e.g., Markov Chain Monte Carlo methods [48, 74, 110, 111, 118].

Following this trend, many models are proposed. Among them are Bayesian Point Machine [58, 106, 130] and Maximum Entropy Estimation [70]. Bayes Point Machine restricts the averaging of the parameters in the version space, which denotes the space where the training data can be perfectly classified. This proposed method is reported to contain a better generalization ability within the global learning framework. But it is challenged to lack systematic ways to extend its applications into non-separable data sets, where the version space may include no candidate solutions. Maximum Entropy Estimation, on the other hand, seems to provide a more flexible and more systematic scheme to perform the averaging of models. By trying to maximize an entropy-like

objective, Maximum Entropy Estimation demonstrates some characteristics of both global learning and local learning. However, this approach argues that only two small data sets are used to evaluate its performance. Moreover, the prior, usually unknown, plays an important role in this model, but has to be assumed beforehand.

2.2.2 Non-parametric learning

In contrast with generative learning discussed in the above, non-parametric learning does not assume any specific global models before learning. Therefore, no risk will be taken on possible wrong assumptions on data. Consequently, non-parametric learning appears to set a more valid foundation than generative learning models. Typical non-parametric learning models in the context of classifications consist of Parzen Window estimation [39] and the widely used k-Nearest-Neighbor model [27, 129]. We will discuss these two models in the following.

The Parzen Window estimation also attempts to estimate a density among the training data. However it employs a totally different way. Parzen window first defines an n-dimensional cell hypercube region R_N over each observation. By defining a window function:

$$w(u) = \begin{cases} 1 & |u_j| \leq 1/2 \quad j = 1, 2, \ldots, n \\ 0 & \text{otherwise} \end{cases} \tag{2.12}$$

the density is then estimated as

$$p_N(\mathbf{z}) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{h_N} w(\frac{\mathbf{z} - \mathbf{z}^i}{h_N}) \ , \tag{2.13}$$

where h_N is defined as the length of the edge of R_N.

From the above, one can observe that Parzen Window puts a local density

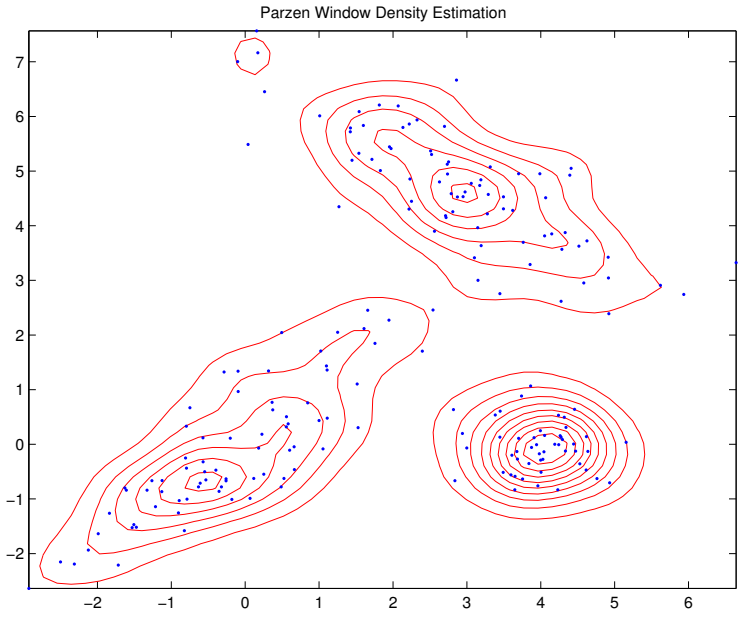


Figure 2.2: An illustration of Parzen window estimation

over each observation, the final density is then the statistical result of averaging all local densities. In practice, the window function can actually be general functions including the most commonly-used Gaussian function. Figure 2.2 illustrates a density estimated by the Parzen window algorithm.

The k-Nearest-Neighbor method can be cast as designing a special cell over each observation and then averages all the cell densities as the overall density for data. More specifically, the cell volume V_N is designed as follows: let the cell volume be a function of the training data, by centering a cell around each point $\mathbf{z}^j$ and increasing the volume until k_N samples are contained, where k_N depends on N. The local density for each observation is then defined as

$$p_N(\mathbf{z}^j) = \frac{k_N/N}{V_N}. \tag{2.14}$$

When used for classifications, the prediction is given by the class with the

maximum posterior probability, i.e.,

$$c = \arg\max_{c_i \in \mathcal{F}} p_N(c_i|\mathbf{z}). \tag{2.15}$$

Further, the posterior probability can be calculated as below:

$$p_N(c_i|\mathbf{z}) = \frac{p_N(c_i, \mathbf{z})}{\sum_{i \in \mathcal{F}} p_N(\mathbf{z}, c_i)} = \frac{\frac{k_i/N}{V}}{\sum_{i \in \mathcal{F}} \frac{k_i/N}{V}} = \frac{k_i}{k}. \tag{2.16}$$

Therefore, the prediction result is just the class with the maximum fraction of the samples in a cell.

These non-parametric methods make no underlying assumptions on data and appear to be more general in real cases. However, using no parameters actually means using many "parameters" so that each parameter would not dominate other parameters (in the discussed models, the data points can be in fact considered as the "parameters"). In such a way, if one parameter fails to work, it will not influence the whole system globally and statistically. However, using many "parameters" also results in serious problems. One of the main problems is that the density is overwhelmingly dependent on the training samples. Therefore, to generate an accurate density, the number of samples needs to be very large (much larger than would be required if we perform the estimation by generative learning approaches). What is even worse is that the number of data will unfortunately increase exponentially with the dimension of data. Another disadvantage caused is its severe requirement for the storage, since all the samples need to be saved beforehand in order to predict new data.

2.2.3 Minimum Error Minimax Probability Machine

Within the context of global learning, a dilemma seems existing: If we assume a specific model as in generative learning, it looses the generality; if we instead use non-parametric learning, it is impractical for high-dimension data. One

question is then proposed, can we have an approach, which does not require a large number of training samples for reducing complexities and also does not assume specific models for maintaining the generality? Towards this end, we propose Minimum Error Minimax Probability Machine in this thesis.

Unlike generative learning or non-parametric learning, Minimum Error Minimax Probability Machine does not try to estimate a distribution over data. Instead, it attempts to extract reliable global information from data and estimates parameters for maximizing the minimal possibility that a future data will fall into the correct class. More precisely, rather than seeking to find an accurate distribution, MEMPM focuses on studying the worst-case probability (which is relatively robust) to predict data. In terms of the style in making decisions, MEMPM is more like a local learning method due to its direct optimization for classification and the task-oriented characteristic. However, because MEMPM only summarizes global information from data (not a distribution) as well, we still locate it in the framework of global learning.

The proposed MEMPM contains many appealing features. Firstly, it represents a distribution-free Bayes optimal classifier in the worst-case scenario. A perfect balance is achieved by MEMPM in this way: No specific model is assumed on data, since it is distribution-free. At the same time, although in the worst-case scenario, it is also the Bayes optimal classifier, which is only originally applicable in the cases with a known distribution. Another critical feature of MEMPM is that under a mild condition, it contains an explicit generalization bound. Furthermore, by exploring the bound, the recently-proposed promising model, Minimax Probability Machine is clearly demonstrated to be its special case. Importantly, based on specifying a bound for one class of data, a Biased Minimax Probability Machine is branched out from MEMPM, which will be shown to provide a rigorous and systematic treatment for biased classifications. We will detail the MEMPM model and BMPM model in the next chapter.

2.3 Local Learning

Local learning adopts a largely different way to construct classifiers. This type of learning is even more task-oriented than Minimum Error Minimax Probability Machine and Maximal Conditional learning. In the context of classifications, only the final mapping function from the features $\mathbf{z}$ to c is crucial. Therefore, describing global information from data or explicitly summarizing a distribution whatever is conditional or joint, is a roundabout or intermediate step and therefore may be deemed wasteful or imprecise especially when the global information cannot be estimated accurately.

Alternatively, recent progress has suggested a local learning method, or well known as the discriminative learning method. The family of approaches directly pin-points the most critical quantities for classifications, while all other information less irrelevant to this purpose is simply omitted. Compared to global learning, no model is assumed and also no explicit global information will be engaged in this scheme. Among this school of methods are Neural Networks [3, 42, 57, 104, 116, 129], Gabriel Graph methods [6, 73, 164], large margin classifiers [30, 137, 139, 141] including Support Vector Machine, a state-of-the-art classifier, which achieves superior performance in various pattern recognition fields. In the following, we will focus on introducing SVM in details.

2.3.1 Support Vector Machines

Support Vector Machine is established based on minimizing the expected classification risk as defined as follows:

$$\mathcal{R}(\Theta) = \int_{\mathbf{z},\mathbf{c}} p(\mathbf{z}, c) l(\mathbf{z}, c, \Theta) \ , \tag{2.17}$$

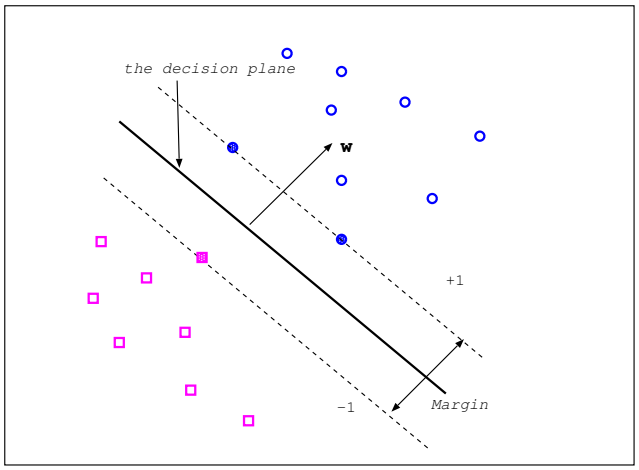


Figure 2.3: An illustration of Support Vector Machine

where, $l(\mathbf{z}, c, \Theta)$ is the loss function. Similar problems occur as in global learning, since generally $p(\mathbf{z}, c)$ is unknown. Therefore, in practice, the above expected risk is often approximated by the so-called empirical risk:

$$\mathcal{R}_{emp}(\Theta) = \frac{1}{N}\sum_{j=1}^{N} l(\mathbf{z}^j, c^j, \Theta) \ . \tag{2.18}$$

The above loss function describes the extent on how close the estimated class disagrees with the real class for the training data. Various metrics can be used for defining this loss function, including the 0-1 loss and the quadratic loss [152].

However, considering only the training data may lead to the over-fitting problem again. In SVM, one big step in dealing with the over-fitting problem has been made, i.e., the margin between two classes should be pulled away in order to reduce the over-fitting risk. Figure 2.3 illustrates the idea of SVM. Two classes of data, depicted as circles and solid dots are presented in this figure. Intuitively observed, there are many decision hyperplanes, which can be adopted for separating these two classes of data. However, the one plotted in this figure is selected as the favorable separating plane, because it contains

the maximum margin between two classes. Therefore, in the objective function of SVM, a regularization term, representing the margin shows up. Moreover, as seen in this figure, only those filled points, called support vectors, mainly determine the separating plane, while other points do not contribute to the margin at all. In another word, only several local points are critical for the classification purpose in the framework of SVM and thus should be extracted.

Actually, a more formal explanation and theoretical foundation can be obtained from the Structure Risk Minimization criterion [17, 154]. Therein, maximizing the margin between different classes of data is minimizing an upper bound of the expected risk, i.e., the VC dimension bound [154]. However, since the focus of this thesis does not lie in the theory of SVM, we will not go further to discuss the details about this. Interested readers can refer to [153, 154].

2.4 Hybrid Learning

Local learning (or simply regarded as SVM) has demonstrated its advantages, such as its state-of-the-art performance (the lower generalization error), the optimal and unique solution, and the mathematical tractability. However, it does discard many useful information from data, e.g., the structure information from data.

An illustrative example has been seen in Figure 1.4. In the current state-of-the-art classifier, i.e., SVM, similar problems also occur. This can be seen in Figure 2.4. In this figure, the purpose is to separate two catergories of data $\mathbf{x}$ and $\mathbf{y}$. As observed, the classification boundary is intuitively observed to be mainly determined by the dotted axis, i.e., the long axis of the $\mathbf{y}$ data (represented by $\square$'s) or the short axis of the $\mathbf{x}$ data (represented by $\circ$'s). Moreover, along this axis, the $\mathbf{y}$ data are more possible to scatter than the $\mathbf{x}$ data, since $\mathbf{y}$ contains a relatively larger variance in this direction. Noting this "global" fact, a good decision hyperplane seems reasonable to lie closer

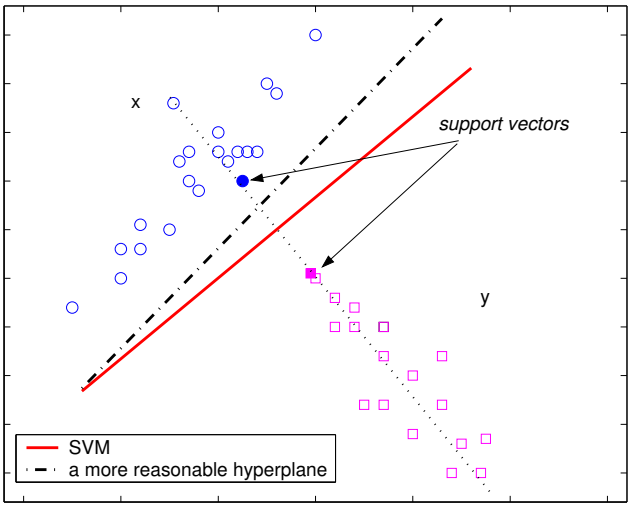


Figure 2.4: A decision hyperplane with considerations of both local and global information.

to the **x** side (see the dash-dot line). However, SVM ignores this kind of "global" information, i.e., the statistical trend of data occurrence: The derived SVM decision hyperplane (the solid line) lies unbiasedly right in the middle of two "local" points (the support vectors). The above considerations directly motivate Maxi-Min Margin Machine.

2.5 Maxi-Min Margin Machine

After examining the road-map of the learning models, especially the global learning and local learning, we have seen a strong motivation for combining two different but complementary schemes. More specifically, borrowing the idea from local learning by assuming no distribution on data would set a valid foundation for the learning models. Meanwhile, fusing robust global information, e.g., structure information, into learning models appears to benefit more on refining decisions in separating data.

Our effort will be made in this direction. As will be detailed in Chapter 4, the hybrid learning model, Maxi-Min Margin Machine successfully plugs

the global information into the learning and enjoys good features from both local learning and global learning. As seen in Figure 2.1, the Maxi-Min Margin Machine model has built up various connections with many models in the literatures; it incorporates Support Vector Machine as a special case, which lies in the framework of local learning; it also includes Minimax Probability Machine and Fisher Discriminant Analysis as direct spin-offs. Moreover, a strong link has been established between this model and Minimum Error Minimax Probability Machine. Moreover, empirical investigations have shown that this combined model outperforms both local learning model such as SVM and global learning models, e.g., MPM.

In the next chapter, we will first present the Minimum Error Minimax Probability Machine, which is a general global learning model. Following that, we then introduce the Maxi-Min Margin Machine and demonstrate its merits both theoretically and empirically.

Chapter 3

A General Global Learning Model: MEMPM

Traditional global learning, especially generative learning, enjoys a long and distinguished history, holding a lot of merits, e.g., a relatively simple optimization, and the flexibility in incorporating global information such as structure information and invariance etc. However, it is widely argued that this model lacks the generality for having to assume a specific model beforehand. Assuming a specific model over data is useful in some cases. However, the assumption may not always coincide with the true data distribution in general and thus may be invalid in many circumstances. In this chapter, we propose a novel global learning model, named Minimum Error Minimax probability Machine (MEMPM), which is directly motivated from Marshall and OlKin Probability Theory [101, 121]. For classifying data correctly, this model focuses on estimating the worse-case probability, which is not only more reliable, but also more importantly provides no need for assuming specific models. Furthermore, this new model consists of several appealing features.

First, MEMPM acutally presents a novel general framework for classifications. As demonstrated later, MEMPM includes a recently-proposed promising model Minimax Probability Machine as its special case, which is reported

to achieve comparable performance to SVM. Interpretations from both viewpoints of the optimal thresholding problem and the geometry will be provided to show the advantages of MEMPM. Moreover, this novel model branches out another promising special case, named Biased Minimax Probability Machine (BMPM) [65] and extends its application into a type of important classifications, i.e., biased classifications.

Second, this model derives a distribution-free Bayes optimal classifier in the worst-case scenario. It thus distinguishes itself from the traditional global learning methods, or more particularly, the traditional Bayes optimal classifiers, which have to assume a distribution on data and thus lack the generality in real cases. Furthermore, we will show that, under some conditions, e.g., when a Gaussian distribution is assumed on data, the worst-case Bayes optimal classifier becomes the true Bayes optimal hyperplane.

Third, the MEMPM model contains an explicit performance indicator, namely an explicit upper bound on the probability of misclassification of future data. Moreover, we will demonstrate theoretically and empirically that MEMPM attains a smaller upper bound of the probability of misclassification than MPM, which thus implies the advantages of MEMPM over MPM.

Fourth, although in general the optimization of MEMPM is shown to be a non-concave problem, empirically, it demonstrates a good concavity in the main "interest" region and thus can be solved practically. Furthermore, we will show that the final optimization problem involves solving a one-dimensional line search problem and thus results in a satisfactory solving method.

This chapter is organized as follows. In the next section, we will first introduce the Marshall and Olkin Theory. We then present the main content of this chapter, the MEMPM model, including its definition, interpretations, the practical solving method, and the sufficient conditions for the convergence into the true Bayes decision hyperplane. Following that, we demonstrate a robust version of MEMPM. In Section 3.4, we seek to kernelize the MEMPM

model to attack nonlinear classification problems. We then, in Section 3.5, present a series of experiments on synthetic data sets and real-world benchmark data sets. In Section 3.6, we analyze the tightness of the worst-case accuracy bound. In Section 3.7, we show that empirically MEMPM is often concave in the main "interest" region. In Section 3.8, we present the limitations of MEMPM and envision the possible future work. Finally, we summarize this chapter in Section 3.9. We also develop a Matlab toolbox to build and evaluate the MEMPM and BMPM classifiers [161, 160].

3.1 Marshall and Olkin Theory

The Marshall and Olkin Theory can be described as follows:

Theorem 1 [**Marshall and Olkin Theory**] The probability that a random vector $\mathbf{y}$ belongs to a convex set $\mathcal{S}$ can be bounded by the following formulation

$$\sup_{\mathbf{y}\sim(\overline{\mathbf{y}},\Sigma_{\mathbf{y}})} \mathbf{Pr}\{\mathbf{y}\in\mathcal{S}\} = \frac{1}{1+d^2}, \text{ with } d^2 = \inf_{\mathbf{y}\in\mathcal{S}} (\mathbf{y}-\overline{\mathbf{y}})^T \Sigma_{\mathbf{y}}^{-1} (\mathbf{y}-\overline{\mathbf{y}}) \ , \qquad (3.1)$$

where the supremum is taken over all distributions for $\mathbf{y}$ containing the mean as $\overline{\mathbf{y}}$ and the covariance matrix as $\Sigma_{\mathbf{y}}$. [1]

The theory provides us with a possibility to assume no model, but bound the probability of misclassifying a point and consequently develop a novel classifier within the framework of global learning. More specifically, one can design a linear separating plane by replacing $\mathcal{S}$ with a half space associated with this linear plane. To take the supremum can then be considered to bound the misclassification rate for one class of data. We in the following, first introduce the model definition and then show how this theory can be applied therein for deriving a distribution-free classifier.

[1] We assume $\Sigma_{\mathbf{y}}$ to be positive definite for simplicity. Otherwise, we can always add a small positive amount to its diagonal elements to force its positive definition.

3.2 Minimum Error Minimax Probability Decision Hyperplane

In this section, we first present the model definition of MEMPM while reviewing the original MPM model. We then in Section 3.2.2 interpret MEMPM with respect to MPM. In Section 3.2.3, we specialize the MEMPM model for dealing with biased classifications. In Section 3.2.4, we analyze the MEMPM optimization problem and propose a practical solving method. In Section 3.2.5, we address the sufficient conditions when the worst-case Bayes optimal classifier derived from MEMPM becomes the true Bayes optimal classifier. In Section 3.2.6, we provide a geometrical interpretation for BMPM and MEMPM.

3.2.1 Problem Definition

The notation in this chapter will largely follow that of [85]. Let $\mathbf{x}$ and $\mathbf{y}$ denote two random vectors representing two classes of data with means and covariance matrices as $\{\overline{\mathbf{x}}, \Sigma_{\mathbf{x}}\}$ and $\{\overline{\mathbf{y}}, \Sigma_{\mathbf{y}}\}$, respectively, in a two-category classification task, where $\mathbf{x}$, $\mathbf{y}$, $\overline{\mathbf{x}}$, $\overline{\mathbf{y}} \in \mathbb{R}^n$, and $\Sigma_{\mathbf{x}}$, $\Sigma_{\mathbf{y}} \in \mathbb{R}^{n\times n}$.

Assuming $\{\overline{\mathbf{x}}, \Sigma_{\mathbf{x}}\}$, $\{\overline{\mathbf{y}}, \Sigma_{\mathbf{y}}\}$ for two classes of data are reliable, MPM attempts to determine the hyperplane $\mathbf{w}^T\mathbf{z} = b$ ($\mathbf{w} \in \mathbb{R}^n\backslash\{\mathbf{0}\}$, $\mathbf{z} \in \mathbb{R}^n$, $b \in \mathbb{R}$, and superscript T denotes the transpose) which can separate two classes of data with the maximal probability. The formulation for the MPM model is written as follows:

$$\max_{\alpha,\beta,\mathbf{w}\neq\mathbf{0},b} \quad \theta\alpha + (1-\theta)\beta \quad \text{s.t.} \tag{3.2}$$

$$\inf_{\mathbf{x}\sim(\overline{\mathbf{x}},\Sigma_{\mathbf{x}})} \mathbf{Pr}\{\mathbf{w}^T\mathbf{x} \geq b\} \geq \alpha, \tag{3.3}$$

$$\inf_{\mathbf{y}\sim(\overline{\mathbf{y}},\Sigma_{\mathbf{y}})} \mathbf{Pr}\{\mathbf{w}^T\mathbf{y} \leq b\} \geq \beta. \tag{3.4}$$

where α and β indicate the worst-case classification accuracies of future

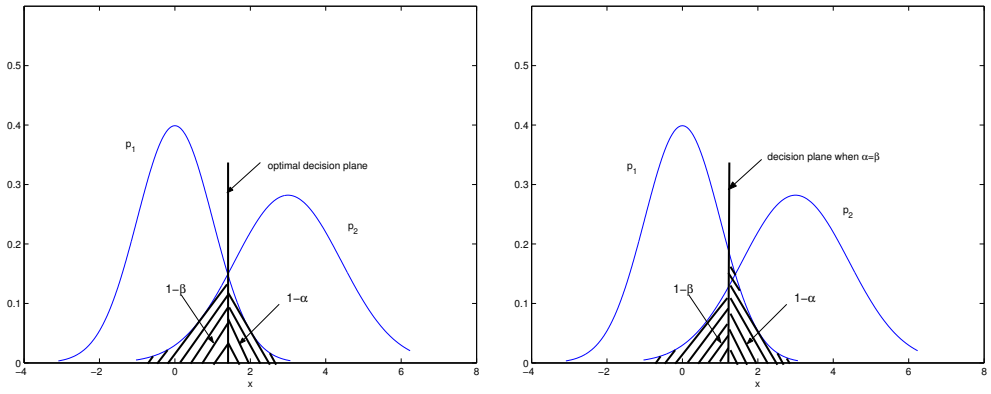
data points for the class $\mathbf{x}$ and $\mathbf{y}$, respectively, namely, the worst-case accuracy for classifying $\mathbf{x}$ data and $\mathbf{y}$ data. Future points $\mathbf{z}$ for which $\mathbf{w}^T\mathbf{z} \geq b$ are then classified as the class $\mathbf{x}$; otherwise they are judged as the class $\mathbf{y}$. $\theta \in [0, 1]$ is the prior probability of the class $\mathbf{x}$ and $1 - \theta$ is thus the prior probability of the class $\mathbf{y}$. Intuitively, maximizing $\theta\alpha + (1 - \theta)\beta$ can be naturally considered as maximizing the expected worst-case accuracy for future data. In other words, this optimization leads to minimizing the expected upper bound of the error rate. More precisely, if we change $\max\{\theta\alpha + (1 - \theta)\beta\}$ to $\min\{\theta(1-\alpha)+(1-\theta)(1-\beta)\}$ and consider $1-\alpha$ as the upper bound probability that an $\mathbf{x}$ data is classified into class $\mathbf{y}$ ($1 - \beta$ is similarly considered), the MEMPM model exactly minimizes the maximum Bayes error and thus derives the Bayes optimal hyperplane in the worst-case scenario. In comparison, MPM assumes the equal worst-case probability for both classes, i.e., it forces $\alpha = \beta$. Obvisouly, this is inappropriate since it is unnecessary that the worst-case accuracies are presumed equal. However, even in such a constrained way, MPM is reported to achieve comparable performacne to SVM, a current state-of-the-art classifier. Therefore, the generalized case of MPM, namely, MEMPM may be expected to be more pomising. This will be empirically demonstrated in the experimental part of this chapter.

3.2.2 Interpretation

We interpret MEMPM with respect to MPM in this section. First, it is evident that if we presume $\alpha = \beta$, the optimization of MEMPM degrades to the MPM optimization. This would mean MPM is actually a special case of MEMPM.

An analogy to illustrate the difference between MEMPM and MPM can be seen in the optimal thresholding problem. Figure 3.1 illustrates this analogy. To separate two classes of one-dimensional data with density functions as p_1 and p_2, respectively, the optimal thresholding is given by the decision plane

in Figure 3.1(a) (assuming the prior probabilities for two classes of data are equal). This optimal thesholding corresponds to the point minimizing the error rate $(1-\alpha)+(1-\beta)$ or maximizing the accuracy $\alpha+\beta$, which is exactly the intersection point of two density functions ($1-\alpha$ represents the area of 135^o-line filled region and $1-\beta$ represents the area of 45^o-line filled region). On the other hand, the thresholding point to force $\alpha=\beta$ is not necessarily the optimal point to separate these two classes.



(a) (b)

Figure 3.1: An analogy to illustrate the difference between MEMPM and MPM with equal prior probabilities for two classes. The optimal decision plane corresponds to the intersection point, where the error $(1-\alpha)+(1-\beta)$ is minimized (or the accuracy $\alpha+\beta$ is maximized) as implied by MEMPM, rather than the one, where α is equal to β as implied by MPM.

It should be clarified that the MEMPM model assumes no distributions. This distinguishes the MEMPM model from the traditional Bayes optimal thresholding method, which has to make specific assumptions on data distribution. On the other hand, although MEMPM minimizes the upper bound of the Bayes error rate of future data points, as shown later in Section 3.2.5, it will represent the true Bayes optimal hyperplane under some conditions, e.g., when a Gaussian distribution is assumed on data.[2]

[2]Another interpretation of the difference between MEMPM and MPM can be stated from

3.2.3 Special Case for Biased Classifications

The above discussion only covers the unbiased classification tasks, which does not favor one class over the other class intentionally. However, another important type of pattern recognition tasks, namely biased classification, arises very often in practice. In this scenario, one class is usually more important than the other class. Thus a bias should be imposed towards the important class. Such typical example can be seen in the diagnosis of epidemical disease. Classifying a patient who is infected with a disease into an opposite class results in serious consequence. Thus in this problem, the classification accuracy should be biased towards the class with disease. In other words, we would prefer to diagnose the person without the disease to be the infected case rather than the other way round.

We in the following demonstrate that MEMPM actually contains a special case we call Biased Minimax Probability Machine for biased classifications. We formulate this special case as:

$$\begin{aligned} \max_{\alpha,\beta,\mathbf{w}\neq\mathbf{0},b} \quad & \alpha \quad \text{s.t.} \\ & \inf_{\mathbf{x}\sim(\overline{\mathbf{x}},\Sigma_{\mathbf{x}})} \mathbf{Pr}\{\mathbf{w}^T\mathbf{x} \geq b\} \geq \alpha, \\ & \inf_{\mathbf{y}\sim(\overline{\mathbf{y}},\Sigma_{\mathbf{y}})} \mathbf{Pr}\{\mathbf{w}^T\mathbf{y} \leq b\} \geq \beta_0, \end{aligned}$$

where β_0 is a pre-specified positive constant, which represents an acceptable accuracy level for the less important class $\mathbf{y}$.

The above optimization utilizes a typical setting in biased classifications,

---

the viewpoint of Game Theory. MPM can be regarded as a non-cooperative competitive game. In this game, each player (class) tries to maximize its individual benefit, i.e., α. The competition leads to each class obtaining the same benefit when all classes fulfill a kind of equilibrium. However, in the game theory, many models, e.g., the prisoners' dilemma, Counot Model and the tragedy of the commons [108], have stated that maximizing individual benefit does not lead to maximizing the global optimum. Our model, on the contrary, can be considered as a kind of cooperative game. It achieves the global optimum through cooperation.

i.e., the accuracy for the important class (associated with $\mathbf{x}$) should be as high as possible, if only the accuracy for the less important class (associated with $\mathbf{y}$) maintains at an acceptable level specified by the lower bound β_0 (which can be set by users).

With quantitatively plugging a specified bias β_0 into classifications and also containing an explicit accuracy bound α for the important class, BMPM provides a more direct and elegant way for biased classifications. Comparatively, to achieve a specified bias, traditional biased classifiers such as the Weighted Support Vector Machine [115] and the Weighted k-Nearest Neighbor method [99] usually adapt different costs for different classes. However, due to the difficulties in building up quantitative connections between the cost and the accuracy,[3] for imposing a specified bias, these methods need resort to the trial and error procedure to attain suitable costs, which are generally indirect and lack rigorous treatments.

3.2.4 Solving the MEMPM Optimization Problem

In this section, we will propose to solve the MEMPM optimization problem. As will be demonstrated shortly, the MEMPM optimization can be transformed into a one-dimensional line search problem. More specifically, the objective function of the line search problem is implicitly determined by dealing with a BMPM problem. Therefore, solving the line search problem corresponds to solving a Sequential Biased Minimax Probability Machine (SBMPM) problem. Before we proceed, we first introduce how to solve the BMPM optimization problem.

[3] Although cross validations could be used to provide empirical connections, they are problem-dependent and are usually slow procedures as well.

Solving the BMPM Optimization Problem

First, we describe Lemma 2, which is developed in [85].

Lemma 2 Given $\mathbf{w} \neq \mathbf{0}$ and b, such that $\mathbf{w}^T\mathbf{y} \leq b$ and $\beta \in [0,1)$, the condition

$$\inf_{\mathbf{y}\sim(\overline{\mathbf{y}},\Sigma_{\mathbf{y}})} \mathbf{Pr}\{\mathbf{w}^T\mathbf{y} \leq b\} \geq \beta,$$

holds if and only if $b - \mathbf{w}^T\overline{\mathbf{y}} \geq \kappa(\beta)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}$ with $\kappa(\beta) = \sqrt{\frac{\beta}{1-\beta}}$.

The lemma can be proved according to the Marshall and Olkin Theory and the Lagrangian Multiplier theory. We provide the detailed proof in Appendix A of this thesis.

By using Lemma 2, we can transform the BMPM optimization problem as follows:

$$\max_{\alpha,\mathbf{w}\neq\mathbf{0},b} \quad \alpha \quad \text{s.t.} \tag{3.5}$$

$$-b + \mathbf{w}^T\overline{\mathbf{x}} \geq \kappa(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}\,, \tag{3.6}$$

$$b - \mathbf{w}^T\overline{\mathbf{y}} \geq \kappa(\beta_0)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}\,, \tag{3.7}$$

where $\kappa(\alpha) = \sqrt{\frac{\alpha}{1-\alpha}}$, $\kappa(\beta_0) = \sqrt{\frac{\beta_0}{1-\beta_0}}$. (3.7) is directly obtained from (3.4) by using Lemma 2. Similarly, by changing $\mathbf{w}^T\mathbf{x} \geq b$ to $\mathbf{w}^T(-\mathbf{x}) \leq -b$, (3.6) can be obtained from (3.3).

From (3.6) and (3.7), we get

$$\mathbf{w}^T\overline{\mathbf{y}} + \kappa(\beta_0)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}} \leq b \leq \mathbf{w}^T\overline{\mathbf{x}} - \kappa(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}\,. \tag{3.8}$$

If we eliminate b from this inequality, we obtain

$$\mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}}) \geq \kappa(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}} + \kappa(\beta_0)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}\,. \tag{3.9}$$

We observe that the magnitude of $\mathbf{w}$ does not influence the solution of (3.9). Moreover, we can assume $\overline{\mathbf{x}} \neq \overline{\mathbf{y}}$; otherwise, if $\overline{\mathbf{x}} = \overline{\mathbf{y}}$, the minimax machine does not have a physical meaning. In this case, (3.9) may even have no solution for every $\beta_0 \neq 0$, since the right hand side would be always positive provided that $\mathbf{w} \neq \mathbf{0}$. Thus in the extreme case, β and α have to be zero, which means the worst-case misclassification are always zero.

Without loss of generality, we can set $\mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}}) = 1$. Thus the problem can be further changed as

$$\max_{\alpha, \mathbf{w} \neq \mathbf{0}} \quad \alpha \quad \text{s.t.} \tag{3.10}$$

$$1 \geq \kappa(\alpha)\sqrt{\mathbf{w}^T \Sigma_{\mathbf{x}} \mathbf{w}} + \kappa(\beta_0)\sqrt{\mathbf{w}^T \Sigma_{\mathbf{y}} \mathbf{w}}\ , \tag{3.11}$$

$$\mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}}) = 1. \tag{3.12}$$

Since $\Sigma_{\mathbf{x}}$ can be assumed as positive definite (otherwise, we can always add a small positive amount to its diagonal elements and make it positive definite), from (3.11) we can obtain:

$$\kappa(\alpha) \leq \frac{1 - \kappa(\beta_0)\sqrt{\mathbf{w}^T \Sigma_{\mathbf{y}} \mathbf{w}}}{\sqrt{\mathbf{w}^T \Sigma_{\mathbf{x}} \mathbf{w}}}\ . \tag{3.13}$$

Because $\kappa(\alpha)$ increases monotonically with α, maximizing α is equivalent to maximizing $\kappa(\alpha)$, which further leads to

$$\max_{\mathbf{w} \neq \mathbf{0}} \quad \frac{1 - \kappa(\beta_0)\sqrt{\mathbf{w}^T \Sigma_{\mathbf{y}} \mathbf{w}}}{\sqrt{\mathbf{w}^T \Sigma_{\mathbf{x}} \mathbf{w}}} \quad \text{s.t.} \quad \mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}}) = 1.$$

This kind of optimization is called Fractional Programming (FP) problem [68, 100, 134]. To elaborate further, this optimization is equivalent to solving the following fractional problem:

$$\max_{\mathbf{w} \neq \mathbf{0}} \quad \frac{f(\mathbf{w})}{g(\mathbf{w})}\ , \tag{3.14}$$

subject to $\mathbf{w} \in A = \{\mathbf{w} | \mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}}) = 1\}$, where $f(\mathbf{w}) = 1 - \kappa(\beta_0)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}$, $g(\mathbf{w}) = \sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}$.

Theorem 3 The Fractional Programming problem (3.14) associated with the BMPM optimization is a pseudo-concave problem, whose every local optimum is the global optimum.

Proof: It is easy to see that the domain A is a convex set on $\mathbb{R}^n$, $f(\mathbf{w})$ and $g(\mathbf{w})$ are differentiable on A. Moreover, since $\Sigma_{\mathbf{x}}$ and $\Sigma_{\mathbf{y}}$ can be both considered as positive definite matrices, $f(\mathbf{w})$ is a concave function on A and $g(\mathbf{w})$ is a convex function on A. Then $\frac{f(\mathbf{w})}{g(\mathbf{w})}$ is a concave-convex FP problem. Hence it is a pseudo-concave problem [134]. Therefore, every local maximum is the global maximum [134]. ■

To handle this specific FP problem, many methods such as the parametric method [134], the dual FP method [29, 133], and the concave FP method [28] can be used. A typical Conjugate Gradient method [11] in solving this problem will have a worst-case $O(n^3)$ time complexity. Adding the time cost to estimate $\overline{\mathbf{x}}$, $\overline{\mathbf{y}}$, $\Sigma_{\mathbf{x}}$, and $\Sigma_{\mathbf{y}}$, the total cost for this method is $O(n^3 + Nn^2)$, where N is the number of data points. This complexity is in the same order as the linear Support Vector Machines [138] and the linear MPM [85].

In this chapter, the Rosen gradient projection method [11] is used to find the solution of this pseudo-concave FP problem, which is proved to converge to a local maximum with a worse-case linear convergence rate. Moreover, the local maximum will exactly be the global maximum in this problem.

Sequential BMPM Optimization Method for MEMPM

We now turn to solving the MEMPM problem. Similar to Section 3.2.4, we can base on Lemma 2 to transform the MEMPM optimization as follows:

$$\max_{\alpha,\beta,\mathbf{w}\neq\mathbf{0},b} \quad \theta\alpha + (1-\theta)\beta \quad \text{s.t.} \tag{3.15}$$

$$-b + \mathbf{w}^T\overline{\mathbf{x}} \geq \kappa(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}\,, \tag{3.16}$$

$$b - \mathbf{w}^T\overline{\mathbf{y}} \geq \kappa(\beta)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}\,. \tag{3.17}$$

Using the similar analysis as in Section 3.2.4, we can further transform the above optimization into

$$\max_{\alpha,\beta,\mathbf{w}\neq\mathbf{0}} \quad \theta\alpha + (1-\theta)\beta \quad \text{s.t.} \tag{3.18}$$

$$1 \geq \kappa(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}} + \kappa(\beta)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}\,, \tag{3.19}$$

$$\mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}}) = 1. \tag{3.20}$$

In the following we provide a lemma to show that the MEMPM solution is actually attained on the boundary of the set formed by the constraints of (3.19) and (3.20).

Lemma 4 The maximum value of $\theta\alpha+(1-\theta)\beta$ under the constraints of (3.19) and (3.20) is achieved when the right hand side of (3.19) is strictly equal to 1.

Proof: Assume the maximum is achieved when $1 > \kappa(\beta)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}} + \kappa(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}$. A new solution constructed by increasing α or $\kappa(\alpha)$ with a small positive amount,[4] and maintaining β, $\mathbf{w}$ unchanged will satisfy the constraints and will be a better solution. ■

By applying Lemma 4, we can transform the optimization problem (3.18)

[4]Since $\kappa(\alpha)$ increases monotonically with α, increasing α a small positive amount corresponds to increasing $\kappa(\alpha)$ a small positive amount.

under the constraints of (3.19) and (3.20) as follows:

$$\max_{\beta,\mathbf{w}\neq\mathbf{0}} \quad \frac{\theta\kappa^2(\alpha)}{\kappa^2(\alpha)+1} + (1-\theta)\beta \quad \text{s.t.} \tag{3.21}$$

$$\mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}}) = 1, \tag{3.22}$$

where $\kappa(\alpha) = \frac{1-\kappa(\beta)\sqrt{\mathbf{w}^T\sum_{\mathbf{y}}\mathbf{w}}}{\sqrt{\mathbf{w}^T\sum_{\mathbf{x}}\mathbf{w}}}$.

In (3.21), if we fix β to a specific value within $[0, 1)$, the optimization is equivalent to maximizing $\frac{\kappa^2(\alpha)}{\kappa^2(\alpha)+1}$ and further equivalent to maximizing $\kappa(\alpha)$, which is exactly the BMPM problem. We can then update β according to some rules and repeat the whole process until an optimal β is found. This is also the so-called line search problem [11, 9]. More precisely, if we denote the value of optimization as a function $f(\beta)$, the above procedure corresponds to finding an optimal β to maximize $f(\beta)$. Instead of using an explicit function as in traditional line search problems, the value of the function here is implicitly given by a BMPM optimization procedure.

Many methods can be used to solve the line search problem. In this chapter, we use the Quadratic Interpolation (QI) method [11]. As illustrated in Figure 3.2, QI finds the maximum point by updating a three-point pattern $(\beta_1,\ \beta_2,\ \beta_3)$ repeatedly. The new β denoted by β_{new} is given by the quadratic interpolation from the three-point pattern. Then a new three-point pattern is constructed by β_{new} and two of $\beta_1, \beta_2, \beta_3$. This method can be shown to converge superlinearly to a local optimum point [11]. Moreover, as shown in Section 3.7, although MEMPM generally cannot guarantee its concavity, empirically it is often a concave problem. Thus the local optimum will be often the global optimum in practice.

Until now, we do not mention how to calculate the intercept b. From Lemma 4, we can see that the inequalities (3.16) and (3.17) will become equalities at the maximum point $(\mathbf{w}_*, b_*)$. The optimal b will thus be obtained

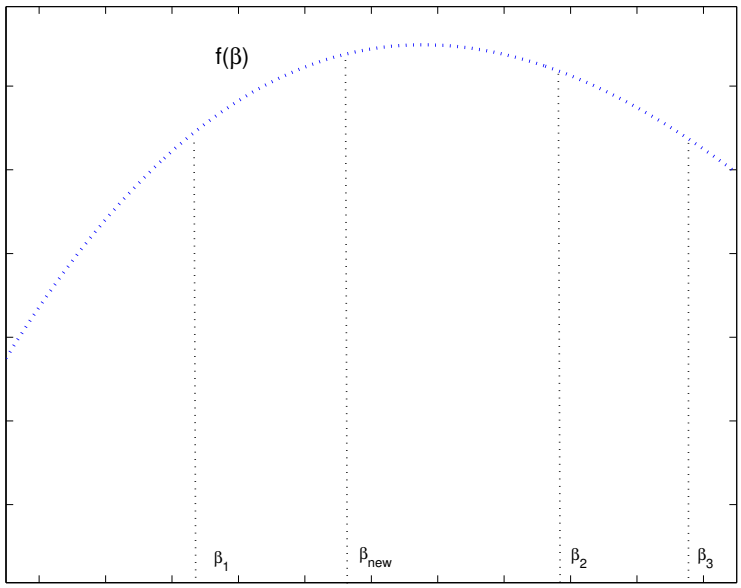


Figure 3.2: A three-point pattern and Quadratic Line search method. A β_{new} is obtained and a new three-point pattern is constructed by β_{new} and two of β_1, β_2 and β_3.

by

$$b_* = \mathbf{w}_*^T\overline{\mathbf{x}} - \kappa(\alpha_*)\sqrt{\mathbf{w}_*^T\Sigma_{\mathbf{x}}\mathbf{a}_*} = \mathbf{w}_*^T\overline{\mathbf{y}} + \kappa(\beta_*)\sqrt{\mathbf{w}_*^T\Sigma_{\mathbf{y}}\mathbf{a}_*}\ . \tag{3.23}$$

3.2.5 When the Worst-case Bayes Optimal Hyperplane Becomes the True One?

As discussed, the MEMPM derives the worst-case Bayes optimal hyperplane, thus it is interesting to dig out on what conditions the worst-case optimal one changes into the true optimal one.

In the following we demonstrate two propositions: the first is that when data are assumed under some distributions, e.g., Gaussian distribution, the MEMPM leads to the Bayes optimal classifier; the second is that when applied into high-dimensional classification tasks, the MEMPM can be adapted to

converge into the true Bayes optimal classifier under the Lyapunov condition.

To introduce the first proposition, we begin with assuming data distribution as a Gaussian distribution.

Assuming $\mathbf{x} \sim \mathcal{N}(\overline{\mathbf{x}}, \Sigma_{\mathbf{x}})$ and $\mathbf{y} \sim \mathcal{N}(\overline{\mathbf{y}}, \Sigma_{\mathbf{y}})$, (3.3) becomes:

$$\begin{aligned} \inf_{\mathbf{x}\sim\mathcal{N}(\overline{\mathbf{x}},\Sigma_{\mathbf{x}})} \mathbf{Pr}\{\mathbf{w}^T\mathbf{x} \geq b\} &= \mathbf{Pr}_{\mathbf{x}\sim\mathcal{N}(\overline{\mathbf{x}},\Sigma_{\mathbf{x}})}\{\mathbf{w}^T\mathbf{x} \geq b\} \\ &= \mathbf{Pr}\{\mathcal{N}(0,1) \geq \frac{b - \mathbf{w}^T\overline{\mathbf{x}}}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}}\} \\ &= 1 - \Phi(\frac{b - \mathbf{w}^T\overline{\mathbf{x}}}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}}) \\ &= \Phi(\frac{-b + \mathbf{w}^T\overline{\mathbf{x}}}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}}) \geq \alpha, \end{aligned} \tag{3.24}$$

where $\Phi(z)$ is the cumulative distribution function for the standard normal Gaussian distribution as defined as:

$$\Phi(z) = \mathbf{Pr}\{\mathcal{N}(0,1) \leq z\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} \exp(-s^2/2) ds.$$

Due to the monotonic property of $\Phi(z)$, we can further write (3.24) as:

$$-b + \mathbf{w}^T\overline{\mathbf{x}} \geq \Phi^{-1}(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}\ .$$

Constraint (3.4) can be reformulated to a similar form. The optimization (3.2) is thus changed as:

$$\begin{aligned} \max_{\alpha,\beta,\mathbf{w}\neq\mathbf{0},b} \quad & \theta\alpha + (1-\theta)\beta, \quad \text{s.t.} \\ & -b + \mathbf{w}^T\overline{\mathbf{x}} \geq \Phi^{-1}(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}\ , \qquad (3.25) \\ & b - \mathbf{w}^T\overline{\mathbf{y}} \geq \Phi^{-1}(\beta)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}\ . \qquad (3.26) \end{aligned}$$

The above optimization is nearly the same as (3.2) subject to the constraints of (3.3) and (3.4) except that, $\kappa(\alpha)$ is equal to $\Phi^{-1}(\alpha)$, instead of $\sqrt{\frac{\alpha}{1-\alpha}}$. Thus,

it can be similarly solved based on the Sequential Biased Minimax Probability Machine method.

On the other hand, the Bayes optimal hyperplane corresponds to the one, $\mathbf{w}^T\mathbf{z} = b$, which minimizes the Bayes error:

$$\min_{\mathbf{w}\neq\mathbf{0},b} \quad \theta\mathbf{Pr}_{\mathbf{x}\sim\mathcal{N}(\overline{\mathbf{x}},\Sigma_{\mathbf{x}})}\{\mathbf{w}^T\mathbf{x} \leq b\} + (1-\theta)\mathbf{Pr}_{\mathbf{y}\sim\mathcal{N}(\overline{\mathbf{y}},\Sigma_{\mathbf{y}})}\{\mathbf{w}^T\mathbf{y} \geq b\}. \quad (3.27)$$

The above is exactly the upper bound of $\theta\alpha + (1-\theta)\beta$. From Lemma 4, we can know (3.25) and (3.26) will eventually become equalities. Traced back to (3.24), the equalities imply that α and β will achieve their upper bounds respectively. Therefore, with the Gaussian distribution assumption on data, the MEMPM derives the optimal Bayes hyperplane.

We propose Proposition 5 to extend the above analysis to general distribution assumptions.

Proposition 5 If the distribution of the normalized random variable $\frac{\mathbf{w}^T\mathbf{x}-\mathbf{w}^T\overline{\mathbf{x}}}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}}$, denoted as $\mathcal{NS}$, is independent of $\mathbf{w}$, as the case in Gaussian distribution, the similar MEMPM version as in Gaussian distribution assumption will be easily derived, except that $\Phi(z)$ is changed as $\mathbf{Pr}\{\mathcal{NS}(0,1) \leq z\}$. In such case, minimizing the Bayes error bound will exactly minimize the true Bayes error.

Before presenting Proposition 7, we first introduce the Central Limit Theorem under the Lyapunov condition [25].

Theorem 6 Let $\mathbf{x}_n$ be a sequence of independent random variables defined on the same probability space. Assume that $\mathbf{x}_n$ has finite expected value μ_n and finite standard deviation σ_n. We define $s_n^2 = \sum_{i=1}^n \sigma_i^2$. Assume that the third central moment $r_n^3 = \sum_{i=1}^n \mathrm{E}(|\mathbf{x}_n - \mu_n|^3)$ are finite for every n, and that $\lim_{n\to\infty} \frac{r_n}{s_n} = 0$ (This is the Lyapunov condition). The sum $S_n = \mathbf{x}_1 + ... + \mathbf{x}_n$ converges towards a Gaussian distribution.

One interesting finding directly elicited from the above Central Limit Theorem is that, if the component variable $\mathbf{x}_i$ of a given n-dimensional random variable $\mathbf{x}$ satisfies the Lyapunov condition, the sum of weighted component variables $\mathbf{x}_i$, $1 \leq i \leq n$, namely, $\mathbf{w}^T\mathbf{x}$ tends to be a Gaussian distribution, as n grows.[5] This shows that, under the Lyapunov condition, when the dimension n grows, the hyperplane derived by MEMPM with Gaussian assumption tends to be the true Bayes optimal hyperplane. In this case, the MEMPM using $\Phi^{-1}(\alpha)$ the inverse function of the normal cumulative distribution, instead of $\sqrt{\frac{\alpha}{1-\alpha}}$, will converge to the true Bayes optimal decision hyperplane in the high-dimensional space. We summarize the analysis into Proposition 7.

Proposition 7 If the component variable $\mathbf{x}_i$ of a given n-dimensional random variable $\mathbf{x}$ satisfies the Lyapunov condition, the MEMPM hyperplane derived by using $\Phi^{-1}(\alpha)$, the inverse function of normal cumulative distribution, will converge to the true Bayes optimal one.

The underlying justifications in the above two propositions root in the fact that the generalized MPM is exclusively determined by the first and second moments. These two propositions actually emphasize the dominance of the first and second moments in representing data. More specifically, Proposition 5 hints that the distribution is only decided by up to the second moments. The Lyapunov condition in Proposition 7 also implies that the second order moment dominates the third order moment in the long run. It also deserves attentions that with the fixed mean and covariance, the distribution of Maximum Entropy Estimation is the Gaussian distribution [75]. This would once again suggest the usage of $\Phi^{-1}(\alpha)$ in the high-dimensional space.

[5] Some techniques such as Independent Component Analysis [32] can be applied to decorrelate the dependence among random variables beforehand.

3.2.6 Geometrical Interpretation

In this section, we first provide a parametric solving method for BMPM. We then demonstrate that this parametric method actually enables a nice geometrical interpretation for both BMPM and MEMPM.

A Parametric Method for BMPM

According to the parametric method, the fractional function can be iteratively optimized in two steps [134]:

Step 1. Find $\mathbf{w}$ by maximizing $f(\mathbf{w}) - \lambda g(\mathbf{w})$ in the domain A, where $\lambda \in \mathbb{R}$ is the newly introduced parameter.

Step 2. Update λ by $\frac{f(\mathbf{w})}{g(\mathbf{w})}$.

The iteration of the above two steps will guarantee to converge to the local maximum, which is also the global maximum in our problem. In the following, we adopt a method to solve the maximization problem in Step 1. Replacing $f(\mathbf{w})$ and $g(\mathbf{w})$, we expand the optimization problem as:

$$\max_{\mathbf{w}\neq\mathbf{0}} \quad 1 - \kappa(\beta_0)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}} - \lambda\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}} \quad \text{s.t.} \quad \mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}}) = 1. \tag{3.28}$$

Maximizing (3.28) is equivalent to $\min_{\mathbf{w}} \kappa(\beta_0)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}} + \lambda\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}$ under the same constraint. By writing $\mathbf{w} = \mathbf{w}_0 + \mathbf{F}\mathbf{u}$, where $\mathbf{w}_0 = (\overline{\mathbf{x}} - \overline{\mathbf{y}})/ \parallel \overline{\mathbf{x}} - \overline{\mathbf{y}} \parallel_2^2$ and $\mathbf{F} \in \mathbb{R}^{n\times(n-1)}$ is an orthogonal matrix whose columns span the subspace of vectors orthogonal to $\overline{\mathbf{x}} - \overline{\mathbf{y}}$, an equivalent form (a factor $\frac{1}{2}$ over each term has been dropped) to remove the constraint can be obtained:

$$\min_{\mathbf{u},\eta>0,\xi>0} \eta + \frac{\lambda^2}{\eta}\|{\Sigma_{\mathbf{x}}}^{1/2}(\mathbf{w}_0 + \mathbf{F}\mathbf{u})\|_2^2 + \xi + \frac{\kappa(\beta_0)^2}{\xi}\|{\Sigma_{\mathbf{y}}}^{1/2}(\mathbf{w}_0 + \mathbf{F}\mathbf{u})\|_2^2 \,, \tag{3.29}$$

where $\eta,\ \xi \in \mathbb{R}$. This optimization form is very similar to the one in Minimax Probability Machine [84] and can also be solved by using an iterative least-squares approach.

A Geometrical Interpretation for BMPM and MEMPM

The parametric method actually enables a nice geometrical interpretation of BMPM and MEMPM in a fashion similar to that of MPM in [85]. Similarly, we assume $\overline{\mathbf{x}} \neq \overline{\mathbf{y}}$ for the meaningful classification and assume that $\Sigma_{\mathbf{x}}$ and $\Sigma_{\mathbf{y}}$ are positive definite for the purpose of simplicity.

By using the 2-norm definition of a vector $\mathbf{z} : \|\mathbf{z}\|_2 = \max\{\mathbf{u}^T\mathbf{z} : \|\mathbf{u}\|_2 \leq 1\}$, we can express (3.28) as its dual form:

$$\begin{aligned} \tau_* := \min_{\mathbf{w}\neq\mathbf{0}} \max_{\mathbf{u},\mathbf{v}} \quad & \lambda\mathbf{u}^T{\Sigma_{\mathbf{x}}}^{1/2}\mathbf{w} + \kappa(\beta_0)\mathbf{v}^T{\Sigma_{\mathbf{y}}}^{1/2}\mathbf{w} + \tau(1 - \mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}})) \\ \text{s.t.} \quad & \|\mathbf{u}\|_2 \leq 1, \|\mathbf{v}\|_2 \leq 1 \ . \end{aligned}$$

We change the order of the min and max operators and consider the min:

$$\begin{aligned} & \min_{\mathbf{w}\neq\mathbf{0}} \quad \lambda\mathbf{u}^T{\Sigma_{\mathbf{x}}}^{1/2}\mathbf{w} + \kappa(\beta_0)\mathbf{v}^T{\Sigma_{\mathbf{y}}}^{1/2}\mathbf{w} + \tau(1 - \mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}})) \\ & = \begin{cases} \tau & \text{if } \tau\overline{\mathbf{x}} - \lambda{\Sigma_{\mathbf{x}}}^{1/2}\mathbf{u} = \tau\overline{\mathbf{y}} + \kappa(\beta_0){\Sigma_{\mathbf{y}}}^{1/2}\mathbf{v} \\ -\infty & \text{otherwise} \end{cases} \end{aligned}$$

Thus, the dual problem can be further changed as:

$$\max_{\tau,\mathbf{u},\mathbf{v}} \quad \tau : \|\mathbf{u}\|_2 \leq 1, \|\mathbf{v}\|_2 \leq 1, \tau\overline{\mathbf{x}} - \lambda{\Sigma_{\mathbf{x}}}^{1/2}\mathbf{u} = \tau\overline{\mathbf{y}} + \kappa(\beta_0){\Sigma_{\mathbf{y}}}^{1/2}\mathbf{v} \ . \quad (3.30)$$

By defining $\ell := 1/\tau$, we rewrite the dual problem as:

$$\min_{\ell,\mathbf{u},\mathbf{v}} \quad \ell : \overline{\mathbf{x}} - \lambda{\Sigma_{\mathbf{x}}}^{1/2}\mathbf{u} = \overline{\mathbf{y}} + \kappa(\beta_0){\Sigma_{\mathbf{y}}}^{1/2}\mathbf{v}, \|\mathbf{u}\|_2 \leq \ell, \|\mathbf{v}\|_2 \leq \ell \ . \quad (3.31)$$

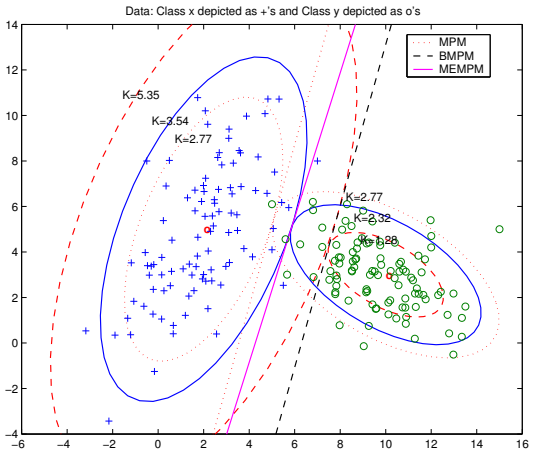


Figure 3.3: The Geometrical interpretation of MEMPM and BMPM. Finding the optimal BMPM hyperplane corresponds to finding the decision plane (the black dashed line) tangent to an ellipsoid (the inner red dashed ellipsoid on the $\mathbf{y}$ side) , which is centered at $\overline{\mathbf{y}}$, shaped by the covariance $\Sigma_{\mathbf{y}}$ and whose Mahalanobis distance to $\overline{\mathbf{y}}$ is exactly equal to $\kappa(\beta_0)$ ($\kappa(\beta_0) = 1.28$ in this example). The worst-case accuracy α for $\mathbf{x}$ is determined by the Mahalanobis distance κ ($\kappa = 5.35$ in this example), at which, an ellipsoid (centered at $\overline{\mathbf{x}}$ and shaped by $\Sigma_{\mathbf{x}}$) is tangent to that $\kappa(\beta_0)$ ellipsoid, i.e., the outer red dahsed ellipsoid on the $\mathbf{x}$ side. In comparison, MPM tries to find out the minimum equality-constrained κ, at which two ellipsoids for $\mathbf{x}$ and $\mathbf{y}$ intersect (both dotted red ellipsoids with $\kappa = 2.77$). For MEMPM, it achieves a tangent hyperplane in a non-balanced fashion, i.e., two ellipsoids may not attain the same κ but is globally optimal in the worst-case setting (see the solid blue ellipsoids).

When the optimum is attained, we have

$$\tau_* = \lambda \|{\Sigma_{\mathbf{x}}}^{1/2}\mathbf{w}_*\|_2 + \kappa(\beta_0)\|{\Sigma_{\mathbf{y}}}^{1/2}\mathbf{w}_*\|_2 = 1/\ell_* \,. \tag{3.32}$$

We consider each side of (3.31) as an ellipsoid centered at the mean $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$ and shaped by the weighted covariance matrices $\lambda\Sigma_{\mathbf{x}}$ and $\kappa(\beta_0)\Sigma_{\mathbf{y}}$ respectively:

$$\mathcal{H}_{\mathbf{x}}(\ell) = \{\mathbf{x} = \overline{\mathbf{x}} + \lambda{\Sigma_{\mathbf{x}}}^{1/2}\mathbf{u} : \|\mathbf{u}\|_2 \le \ell\}, \tag{3.33}$$

$$\mathcal{H}_{\mathbf{y}}(\ell) = \{\mathbf{y} = \overline{\mathbf{y}} + \kappa(\beta_0){\Sigma_{\mathbf{y}}}^{1/2}\mathbf{v} : \|\mathbf{v}\|_2 \le \ell\} \tag{3.34}$$

The above optimization involves finding a minimum ℓ for which two ellipsoids intersect. For the optimum ℓ, these two ellipsoids would be tangent to each other. We further note that, according to Lemma 4, at the optimum, λ_*, which is maximized via a series of the above procedures, would satisfy

$$1 = \lambda_* \|{\Sigma_{\mathbf{x}}}^{1/2}\mathbf{w}_*\|_2 + \kappa(\beta_0)\|{\Sigma_{\mathbf{y}}}^{1/2}\mathbf{w}_*\|_2 = \tau_* = 1/\ell_* \tag{3.35}$$

$$\Rightarrow \ell_* = 1 \; . \tag{3.36}$$

This means that the ellipsoid for the class $\mathbf{y}$ finally changes to the one centered at $\overline{\mathbf{y}}$, whose Mahalanobis distance to $\overline{\mathbf{y}}$ is exactly equal to $\kappa(\beta_0)$. Moreover, the ellipsoid for the class $\mathbf{x}$ would be the one centered at $\overline{\mathbf{x}}$ and tangent to the ellipsoid for the class $\mathbf{y}$. In comparison, for MPM, two ellipsoids grow with the same speed (with the same $\kappa(\alpha)$ and $\kappa(\beta)$). On the other hand, since MEMPM corresponds to solving a sequence of BMPMs, it similarly leads to a hyperplane tangent to two ellipsoids, which achieves to minimize the maximum of the worst-case Bayes error. Moreover, it is not necessarily attained in a balanced way as in MPM, i.e., two ellipsoids do not necessarily grow with the same speed and hence probably contain the unequal Mahalanobis distance from their corresponding centers. This is illustrated in Figure 3.3.

3.3 Robust Version

In the above, the estimates of means and covariance matrices are assumed reliable. We now consider how the probabilistic framework in (3.2) changes against the variation of the means and covariance matrices:

$$\max_{\alpha,\beta,\mathbf{w}\neq\mathbf{0},b} \quad \theta\alpha + (1-\theta)\beta \quad \text{s.t.} \tag{3.37}$$

$$\inf_{\mathbf{x}\sim(\bar{\mathbf{x}},\Sigma_{\mathbf{x}})} \mathbf{Pr}\{\mathbf{w}^T\mathbf{x} \geq b\} \geq \alpha, \forall(\bar{\mathbf{x}},\Sigma_{\mathbf{x}}) \in \mathcal{X} \; , \tag{3.38}$$

$$\inf_{\mathbf{y}\sim(\bar{\mathbf{y}},\Sigma_{\mathbf{y}})} \mathbf{Pr}\{\mathbf{w}^T\mathbf{y} \leq b\} \geq \beta, \forall(\bar{\mathbf{y}},\Sigma_{\mathbf{y}}) \in \mathcal{Y} \; , \tag{3.39}$$

where $\mathcal{X}$ and $\mathcal{Y}$ are the sets of means and covariance matrices and are the subsets of $\mathbb{R}\times\mathcal{P}_n^+$, where $\mathcal{P}_n^+$ is the set of $n\times n$ symmetric positive semidefinite matrices.

Motivated by the tractability of the problem and from the statistical view, a specific setting of $\mathcal{X}$ and $\mathcal{Y}$ is proposed in [85]. However, they consider the same variations of the means for two classes, which is easy to handle but less general. Now, considering the unequal treatment of each class, we propose the following setting, which is in a more general and complete form:

$$\mathcal{X} = \left\{(\bar{\mathbf{x}}, \Sigma_{\mathbf{x}}) \,|\, (\bar{\mathbf{x}} - \bar{\mathbf{x}}^0){\Sigma_{\mathbf{x}}}^{-1}(\bar{\mathbf{x}} - \bar{\mathbf{x}}^0) \le \nu_{\mathbf{x}}^2,\, \Sigma_{\mathbf{x}} \in \|\Sigma_{\mathbf{x}} - {\Sigma_{\mathbf{x}}}^0\|_F \le \rho_{\mathbf{x}}\right\},$$
$$\mathcal{Y} = \left\{(\bar{\mathbf{y}}, \Sigma_{\mathbf{y}}) \,|\, (\bar{\mathbf{y}} - \bar{\mathbf{y}}^0){\Sigma_{\mathbf{y}}}^{-1}(\bar{\mathbf{y}} - \bar{\mathbf{y}}^0) \le \nu_{\mathbf{y}}^2,\, \Sigma_{\mathbf{y}} \in \|\Sigma_{\mathbf{y}} - {\Sigma_{\mathbf{y}}}^0\|_F \le \rho_{\mathbf{y}}\right\},$$

where $\bar{\mathbf{x}}^0$, $\Sigma_{\mathbf{x}}^0$ are the "nominal" means and covariance matrices obtained through estimating. Parameters $\nu_{\mathbf{x}}$, $\nu_{\mathbf{y}}$, $\rho_{\mathbf{x}}$, and $\rho_{\mathbf{y}}$ are positive constants. The matrix norm is defined as the Frobenius norm: $\|M\|_F^2 = \mathrm{Tr}(M^T M)$.

With the assumption that variations of the means for two classes are the same, the parameters $\nu_{\mathbf{x}}$ and $\nu_{\mathbf{y}}$ are required equal in [85]. This may enable the direct usage of the MPM optimization into its robust version. However, the assumption may not be true in real cases. Moreover, in MEMPM, this requirement is also not necessary and inappropriate. This will be later demonstrated in the experiment.

By applying the results from [85], we obtain the robust MEMPM as:

$$\begin{aligned}
\max_{\alpha,\beta,\mathbf{w}\neq\mathbf{0},b} \quad & \theta\alpha + (1-\theta)\beta \quad \text{s.t.} \\
-b + \mathbf{w}^T\bar{\mathbf{x}}^0 &\ge (\kappa(\alpha) + \nu_{\mathbf{x}})\sqrt{\mathbf{w}^T({\Sigma_{\mathbf{x}}}^0 + \rho_{\mathbf{x}} I_n)\mathbf{w}}, \\
b - \mathbf{w}^T\bar{\mathbf{y}}^0 &\ge (\kappa(\beta) + \nu_{\mathbf{y}})\sqrt{\mathbf{w}^T({\Sigma_{\mathbf{y}}}^0 + \rho_{\mathbf{y}} I_n)\mathbf{w}}.
\end{aligned}$$

Analogously, we transform the above optimization problem as:

$$\max_{\alpha,\beta,\mathbf{w}\neq\mathbf{0}} \theta\frac{\kappa_r^2(\alpha)}{1+\kappa_r^2(\alpha)} + (1-\theta)\beta \quad \text{s.t. } \mathbf{w}^T(\bar{\mathbf{x}}^0 - \bar{\mathbf{y}}^0) = 1, \tag{3.40}$$

where $\kappa_r(\alpha) = \max\left(\frac{1-(\kappa(\beta)+\nu_{\mathbf{y}})\sqrt{\mathbf{w}^T(\Sigma_{\mathbf{y}}{}^0+\rho_{\mathbf{y}}I_n)\mathbf{w}}}{\sqrt{\mathbf{w}^T(\Sigma_{\mathbf{x}}{}^0+\rho_{\mathbf{x}}I_n)\mathbf{w}}} - \nu_{\mathbf{x}}, 0\right)$ and thus can be solved by the SBMPM method. The optimal b is therefore calculated by

$$\begin{aligned} b_* &= \mathbf{a}_*{}^T\bar{\mathbf{x}}^0 - (\kappa(\alpha_*)+\nu_{\mathbf{x}})\sqrt{\mathbf{a}_*{}^T(\Sigma_{\mathbf{x}}{}^0+\rho_{\mathbf{x}}I_n)\mathbf{a}_*} \\ &= \mathbf{a}_*{}^T\bar{\mathbf{y}}^0 + (\kappa(\beta_*)+\nu_{\mathbf{y}})\sqrt{\mathbf{a}_*{}^T(\Sigma_{\mathbf{y}}{}^0+\rho_{\mathbf{y}}I_n)\mathbf{a}_*}. \end{aligned}$$

Remarks. Interestingly, if MPM is treated with unequal robust parameters $\nu_{\mathbf{x}}$ and $\nu_{\mathbf{y}}$, it leads to solving an optimization similar to MEMPM, since $\kappa(\alpha)+\nu_{\mathbf{x}}$ will not be equal to $\kappa(\alpha)+\nu_{\mathbf{y}}$. In addition, similar to the robust MPM, when applied in practice, the specific values of $\nu_{\mathbf{x}}$, $\nu_{\mathbf{y}}$, $\rho_{\mathbf{x}}$, and $\rho_{\mathbf{y}}$ can be provided based on the Central Limit Theorem.

3.4 Kernelization

We note that, in the above, the classifier derived from MEMPM is given in a linear configuration. In order to handle nonlinear classification problems, in this section, we seek to use the kernelization trick [109] to map the n-dimensional data points into a high-dimensional feature space $\mathbb{R}^f$, where a linear classifier corresponds to a nonlinear hyperplane in the original space.

Since the optimization of MEMPM corresponds to a sequence of BMPM optimization problems, this model naturally inherits the kernelization ability of BMPM. We thus in the following mainly address the kernelization of BMPM.

Assuming training data points are represented by $\{\mathbf{x}_i\}_{i=1}^{N_{\mathbf{x}}}$ and $\{\mathbf{y}_j\}_{j=1}^{N_{\mathbf{y}}}$ for

the class $\mathbf{x}$ and $\mathbf{y}$, respectively, the kernel mapping can be formulated as:

$$\mathbf{x} \to \varphi(\mathbf{x}) \sim (\overline{\varphi(\mathbf{x})}, \Sigma_{\varphi(\mathbf{x})}),$$
$$\mathbf{y} \to \varphi(\mathbf{y}) \sim (\overline{\varphi(\mathbf{y})}, \Sigma_{\varphi(\mathbf{y})}),$$

where $\varphi : \mathbb{R}^n \to \mathbb{R}^f$ is a mapping function. The corresponding linear classifier in $\mathbb{R}^f$ is $\mathbf{w}^T\varphi(\mathbf{z}) = b$, where $\mathbf{w}$, $\varphi(\mathbf{z}) \in \mathbb{R}^f$, and $b \in \mathbb{R}$. Similarly, the transformed FP optimization in BMPM can be written as:

$$\max_{\mathbf{w}} \frac{1 - \kappa(\beta_0)\sqrt{\mathbf{w}^T\Sigma_{\varphi(\mathbf{y})}\mathbf{w}}}{\sqrt{\mathbf{w}^T\Sigma_{\varphi(\mathbf{x})}\mathbf{w}}} \quad \text{s.t.} \quad \mathbf{w}^T(\overline{\varphi(\mathbf{x})} - \overline{\varphi(\mathbf{y})}) = 1. \tag{3.41}$$

However, to make the kernel work, we need to represent the final decision hyperplane and the optimization into a kernel form, $K(\mathbf{z}_1, \mathbf{z}_2) = \varphi(\mathbf{z}_1)^T\varphi(\mathbf{z}_2)$, namely an inner product form of the mapping data points.

3.4.1 Kernelization Theory for BMPM

In the following, we demonstrate that, although BMPM possesses a significantly different optimization form from MPM, the kernelization theory proposed in [85] is still viable, provided that suitable estimates for means and covariance matrices are applied therein.

We first state a theory similar to Corollary 5 of [85] and prove its validity in BMPM.

Corollary 8 If the estimates of means and covariance matrices are given in

BMPM as

$$\overline{\varphi(\mathbf{x})} = \sum_{i=1}^{N_{\mathbf{x}}} \lambda_i \varphi(\mathbf{x}_i), \quad \overline{\varphi(\mathbf{y})} = \sum_{j=1}^{N_{\mathbf{y}}} \omega_j \varphi(\mathbf{y}_j) \ ,$$

$$\Sigma_{\varphi(\mathbf{x})} = \rho_{\mathbf{x}} \mathbf{I}_n + \sum_{i=1}^{N_{\mathbf{x}}} \Lambda_i (\varphi(\mathbf{x}_i) - \overline{\varphi(\mathbf{x})})(\varphi(\mathbf{x}_i) - \overline{\varphi(\mathbf{x})})^T \ ,$$

$$\Sigma_{\varphi(\mathbf{y})} = \rho_{\mathbf{y}} \mathbf{I}_n + \sum_{j=1}^{N_{\mathbf{y}}} \Omega_j (\varphi(\mathbf{y}_j) - \overline{\varphi(\mathbf{y})})(\varphi(\mathbf{y}_j) - \overline{\varphi(\mathbf{y})})^T \ ,$$

where $\mathbf{I}_n$ is the identity matrix of dimension n, then the optimal $\mathbf{w}$ in problem (3.41) lies in the space spanned by the training points.

Proof: Similar to [85], we write $\mathbf{w} = \mathbf{w}_p + \mathbf{w}_v$, where $\mathbf{w}_p$ is the projection of $\mathbf{w}$ in the vector space spanned by all the training data points and $\mathbf{w}_v$ is the orthogonal component to this span space. It can be easily verified that (3.41) changes to maximize the following:

$$\frac{1 - \kappa(\beta_0)\sqrt{\mathbf{w}_p^T \sum_{i=1}^{N_{\mathbf{x}}} \Lambda_i (\varphi(\mathbf{x}_i) - \overline{\varphi(\mathbf{x})})(\varphi(\mathbf{x}_i) - \overline{\varphi(\mathbf{x})})^T \mathbf{w}_p + \rho_{\mathbf{x}}(\mathbf{w}_p^T \mathbf{w}_p + \mathbf{w}_d^T \mathbf{w}_d)}}{\sqrt{\mathbf{w}_p^T \sum_{j=1}^{N_{\mathbf{y}}} \Omega_j (\varphi(\mathbf{y}_j) - \overline{\varphi(\mathbf{y})})(\varphi(\mathbf{y}_j) - \overline{\varphi(\mathbf{y})})^T \mathbf{w}_p + \rho_{\mathbf{y}}(\mathbf{w}_p^T \mathbf{w}_p + \mathbf{w}_d^T \mathbf{w}_d)}}$$

subject to the constraints of $\mathbf{w}_p^T(\overline{\varphi(\mathbf{x})} - \overline{\varphi(\mathbf{y})}) = 1$. Since we intend to maximize the fractional form and both the denominator and the numerator are positive, the denominator needs to be as small as possible and the numerator needs to be as large as possible. This would finally lead to $\mathbf{w}_d = \mathbf{0}$. In other words, the optimal $\mathbf{w}$ lies in the vector space spanned by all the training data points. Note that the introduction of $\rho_{\mathbf{x}}$ and $\rho_{\mathbf{y}}$ actually enables a direct application of the robust estimates into the kernelization. ■

According to Corollary 8, if appropriate estimates of means and covariance matrices are applied, the optimal $\mathbf{w}$ can be written as the linear combination of training points. In particular, if we obtain the means and covariance matrices

as the plug-in estimates, i.e.,

$$\overline{\varphi(\mathbf{x})} = \frac{1}{N_{\mathbf{x}}} \sum_{i=1}^{N_{\mathbf{x}}} \varphi(\mathbf{x}_i) \ ,$$

$$\overline{\varphi(\mathbf{y})} = \frac{1}{N_{\mathbf{y}}} \sum_{j=1}^{N_{\mathbf{y}}} \varphi(\mathbf{y}_j) \ ,$$

$$\Sigma_{\varphi(\mathbf{x})} = \frac{1}{N_{\mathbf{x}}} \sum_{i=1}^{N_{\mathbf{x}}} (\varphi(\mathbf{x}_i) - \overline{\varphi(\mathbf{x})})(\varphi(\mathbf{x}_i) - \overline{\varphi(\mathbf{x})})^T \ ,$$

$$\Sigma_{\varphi(\mathbf{y})} = \frac{1}{N_{\mathbf{y}}} \sum_{j=1}^{N_{\mathbf{y}}} (\varphi(\mathbf{y}_j) - \overline{\varphi(\mathbf{y})})(\varphi(\mathbf{y}_j) - \overline{\varphi(\mathbf{y})})^T \ ,$$

we can write $\mathbf{w}$ as

$$\mathbf{w} = \sum_{i=1}^{N_{\mathbf{x}}} \mu_i \varphi(\mathbf{x}_i) + \sum_{j=1}^{N_{\mathbf{y}}} \upsilon_j \varphi(\mathbf{y}_j), \tag{3.42}$$

where the coefficients μ_i, $\upsilon_j \in \mathbb{R}$ for $i = 1, \ldots, N_{\mathbf{x}}$ and $j = 1, \ldots, N_{\mathbf{y}}$.

By simply substituting (3.42) and four plug-in estimates into (3.41), we can obtain the Kernelization Theorem of BMPM.

3.4.2 Notations in Kernelization Theorem of BMPM

Before we present the main kernelization result, we first introduce the notations. Let $\{\mathbf{z}\}_{i=1}^{N}$ denote all $N = N_{\mathbf{x}} + N_{\mathbf{y}}$ data points in the training set where

$$\mathbf{z}_i = \mathbf{x}_i \qquad i = 1, 2, \ldots, N_{\mathbf{x}} \ ,$$

$$\mathbf{z}_i = \mathbf{y}_{i-N_{\mathbf{x}}} \qquad i = N_{\mathbf{x}} + 1, N_{\mathbf{x}} + 2, \ldots, N.$$

The element of the Gram matrix $\mathbf{K}$ in the position of (i, j) is defined as $\mathbf{K}_{i,j} = \varphi(\mathbf{z}_i)^T \varphi(\mathbf{z}_j)$ for $i, j = 1, 2, \ldots, N$. We further define $\mathbf{K}_{\mathbf{x}}$ and $\mathbf{K}_{\mathbf{y}}$ as the

matrices formed by the first $N_{\mathbf{x}}$ rows and the last $N_{\mathbf{y}}$ rows of $\mathbf{K}$, respectively, namely,

$$\mathbf{K} := \begin{pmatrix} \mathbf{K_x} \\ \mathbf{K_y} \end{pmatrix}.$$

By setting the row average of the $\mathbf{K_x}$ block and the $\mathbf{K_x}$ block to zero, the block-row-averaged Gram matrix $\tilde{\mathbf{K}}$ is thus obtained:

$$\mathbf{K} := \begin{pmatrix} \tilde{\mathbf{K}}_{\mathbf{x}} \\ \tilde{\mathbf{K}}_{\mathbf{y}} \end{pmatrix} = \begin{pmatrix} \mathbf{K_x} - \mathbf{1}_{N_{\mathbf{x}}} \tilde{\mathbf{k}}_{\mathbf{x}}^T \\ \mathbf{K_y} - \mathbf{1}_{N_{\mathbf{y}}} \tilde{\mathbf{k}}_{\mathbf{y}}^T \end{pmatrix},$$

where $\tilde{\mathbf{k}}_{\mathbf{x}}, \tilde{\mathbf{k}}_{\mathbf{y}} \in \mathbb{R}^{N_{\mathbf{x}}+N_{\mathbf{y}}}$ are defined as:

$$[\tilde{\mathbf{k}}_{\mathbf{x}}]_i := \frac{1}{N_{\mathbf{x}}} \sum_{j=1}^{N_{\mathbf{x}}} \mathbf{K}(\mathbf{x}_j, \mathbf{z}_i),$$
$$[\tilde{\mathbf{k}}_{\mathbf{y}}]_i := \frac{1}{N_{\mathbf{y}}} \sum_{j=1}^{N_{\mathbf{y}}} \mathbf{K}(\mathbf{y}_j, \mathbf{z}_i).$$

In the above, $\mathbf{1}_{N_{\mathbf{x}}} \in \mathbb{R}^{N_{\mathbf{x}}}$ and $\mathbf{1}_{N_{\mathbf{y}}} \in \mathbb{R}^{N_{\mathbf{y}}}$, which are defined as:

$$\mathbf{1}_i = 1, \quad i = 1, 2, \ldots N_{\mathbf{x}},$$
$$\mathbf{1}_j = 1, \quad j = 1, 2, \ldots N_{\mathbf{y}}.$$

Finally, we define vector formed by the coefficients of $\boldsymbol{\gamma}$ as

$$\boldsymbol{w} = [\mu_1, \mu_2, \ldots, \mu_{N_{\mathbf{x}}}, \upsilon_1, \upsilon_2, \ldots, \upsilon_{N_{\mathbf{y}}}]^T. \tag{3.43}$$

3.4.3 Kernelization Results

Theorem 9 [Kernelization Theorem of BMPM] The optimal decision hyperplane of the problem (3.41) involves solving the Fractional Programming

problem

$$\kappa(\alpha_*) = \max_{\mathbf{w}\neq\mathbf{0}} \frac{1-\kappa(\beta_0)\sqrt{\frac{1}{N_\mathbf{y}}\mathbf{w}^T\tilde{\mathbf{K}}_\mathbf{y}^T\tilde{\mathbf{K}}_\mathbf{y}\mathbf{w}}}{\sqrt{\frac{1}{N_\mathbf{x}}\mathbf{w}^T\tilde{\mathbf{K}}_\mathbf{x}^T\tilde{\mathbf{K}}_\mathbf{x}\mathbf{w}}} \quad \text{s.t.} \quad \mathbf{w}^T(\tilde{\mathbf{k}}_\mathbf{x}-\tilde{\mathbf{k}}_\mathbf{y}) = 1 \ .$$

The intercept b is calculated as

$$b_* = \mathbf{w}_*^T\tilde{\mathbf{k}}_\mathbf{x} - \kappa(\alpha_*)\sqrt{\frac{1}{N_\mathbf{x}}\mathbf{w}_*^T\tilde{\mathbf{K}}_\mathbf{x}^T\tilde{\mathbf{K}}_\mathbf{x}\mathbf{w}_*} = \mathbf{w}_*^T\tilde{\mathbf{k}}_\mathbf{y} + \kappa(\beta_0)\sqrt{\frac{1}{N_\mathbf{y}}\mathbf{w}_*^T\tilde{\mathbf{K}}_\mathbf{y}^T\tilde{\mathbf{K}}_\mathbf{y}\mathbf{w}_*} \ ,$$

where $\kappa(\alpha_*)$ is obtained when (3.44) attains its optimum $(\mathbf{w}_*, b_*)$. For the robust version of BMPM, we can incorporate the variations of the means and covariances by conducting the following replacements:

$$\begin{aligned}
&\frac{1}{N_\mathbf{x}}\mathbf{w}_*^T\tilde{\mathbf{K}}_\mathbf{x}^T\tilde{\mathbf{K}}_\mathbf{x}\mathbf{w}_* \to \mathbf{w}_*^T(\frac{1}{N_\mathbf{x}}\tilde{\mathbf{K}}_\mathbf{x}^T\tilde{\mathbf{K}}_\mathbf{x} + \rho_\mathbf{x}\mathbf{K})\mathbf{w}_* \ , \\
&\frac{1}{N_\mathbf{y}}\mathbf{w}_*^T\tilde{\mathbf{K}}_\mathbf{y}^T\tilde{\mathbf{K}}_\mathbf{y}\mathbf{w}_* \to \mathbf{w}_*^T(\frac{1}{N_\mathbf{y}}\tilde{\mathbf{K}}_\mathbf{y}^T\tilde{\mathbf{K}}_\mathbf{y} + \rho_\mathbf{y}\mathbf{K})\mathbf{w}_* \ , \\
&\kappa(\beta_0) \to \kappa(\beta_0) + \mu_\mathbf{y} \ , \\
&\kappa(\alpha_*) \to \kappa(\alpha_*) + \mu_\mathbf{x} \ .
\end{aligned}$$

The optimal decision hyperplane can be represented as a linear form in the kernel space

$$f(\mathbf{z}) = \sum_{i=1}^{N_\mathbf{x}} \mathbf{w}_{*i}\mathbf{K}(\mathbf{z},\mathbf{x}_i) + \sum_{i=1}^{N_\mathbf{y}} \mathbf{w}_{*N_\mathbf{x}+i}\mathbf{K}(\mathbf{z},\mathbf{y}_i) - b_* .$$

3.5 Experiments

In this section, we first evaluate our model on a synthetic dataset. Then we compare the performance of MEMPM with that of MPM, on six real-world benchmark data sets (since MPM is reported comparable to SVM, we do not perform comparisons with SVM. To demonstrate BMPM is ideal for imposing

a specified bias in classification, we also implement it on the Heart-disease dataset. The means and covariance matrices for two classes are obtained directly from the training data sets by plug-in estimations. The prior probability θ is given by the proportion of $\mathbf{x}$ data in the training dataset.

3.5.1 Model Illustration on a Synthetic Dataset

To verify that the MEMPM model achieves the minimum Bayes error rate in the Gaussian distribution, we synthetically generate two classes of two-dimensional Gaussian data. As plotted in Figure 3.4(a), data associated with the class $\mathbf{x}$ are generated with the mean $\overline{\mathbf{x}}$ as $[3, 0]^T$ and the covariance matrix $\Sigma_{\mathbf{x}}$ as $[4,\ 0; 0,\ 1]$, while data associated with the class $\mathbf{y}$ are generated with the mean $\overline{\mathbf{y}}$ as $[-1, 0]^T$ and the covariance matrix $\Sigma_{\mathbf{y}}$ as $[1,\ 0; 0,\ 5]$. The solved decision hyperplane $Z_1 = 0.333$ given by MPM is plotted as the solid blue line and the solved decision hyperplane $Z_1 = 0.660$ given by MEMPM is plotted as the dashed red line. From the geometrical interpretation, both hyperplanes should be perpendicular to the Z_1 axis.

As shown in Figure 3.4(b), the MEMPM hyperplane exactly represents the optimal thresholding under the distributions of the first dimension for two classes of data, i.e., the intersection point of two density functions. On the other hand, we find that, the MPM hyperplane exactly corresponds to the thresholding point with the same error rate for two classes of data, since the cumulative distribution $P_{\mathbf{x}}(Z_1 < 0.333)$ and $P_{\mathbf{y}}(Z_1 > 0.333)$ are exactly the same.

3.5.2 Evaluations on Benchmark Data Sets

We next evaluate our algorithm on six benchmark data sets. Data for the Twonorm problem were generated according to [16]. The rest five data sets

including the Breast, Ionosphere, Pima, Heart-disease, and Vote data were obtained from UCI machine learning repository [12]. Since handling the missing attribute values is out of the scope of this chapter, we simply remove instances with missing attribute values in these data sets.

We randomly partition data into 90% training and 10% test sets. The final results are averaged over 50 random partitions of data. We compare the performance of MEMPM and MPM in both the linear setting and Gaussian kernel setting. The width parameter (σ) for the Gaussian kernel is obtained via cross validations over 50 random partitions of the training set. The experimental results are summarized in Table 3.1 and Table 3.2 for the linear kernel and Guassian kernel respectively.

From the results, we can see that, our MEMPM demonstrates better performance than MPM in both the linear and Gaussian kernel setting. Moreover, as observed in these benchmark datasets, the MEMPM hyperplanes are obtained with significantly unequal α and β except in the Twonorm set. This further confirms the validity of our proposition, i.e., the optimal minimax machine is not certain to achieve the same worst-case accuracies for two classes. For the Twonorm, it is also not an exception. The two classes of data in this set are generated under the multivariate normal distributions with the same covariance matrices. In this special case, the intersection point of two density functions will exactly represent the optimal thresholding point and the one with the same error rate for each class as well. Another important finding is that, the accuracy bounds, namely $\theta\alpha + (1-\theta)\beta$ in MEMPM and α in MPM are all increased in the Gaussian kernel setting when compared with those in the linear setting. This shows the advantage of the kernelized probability machine over the linear probability machine.

In addition, to clearly see the relationship between the bounds and the test set accuracies (TSA), we plot them in Figure 3.5. As observed, the test set accuracies including $\text{TSA}_{\mathbf{x}}$ (for the class $\mathbf{x}$), $\text{TSA}_{\mathbf{y}}$ (for the class $\mathbf{y}$), and

the overall accuracies TSA are all greater than their corresponding accuracy bounds both in MPM and MEMPM. This demonstrates how the accuracy bound can serve as the performance indicator on future data. It is also observed that the overall worst-case accuracies $\theta\alpha+(1-\theta)\beta$ in MEMPM are greater than α in MPM both in the linear and Gaussian setting. This again demonstrates the advantages of MEMPM over MPM.

Dataset	MEMPM				MPM	
	α	β	$\theta\alpha+(1-\theta)\beta$	Accuracy	α	Accuracy
Twonorm(%)	$80.3 \pm 0.2\%$	$79.9 \pm 0.1\%$	$80.1 \pm 0.1\%$	$\mathbf{97.9 \pm 0.1}\%$	$80.1 \pm 0.1\%$	$\mathbf{97.9 \pm 0.1}\%$
Breast(%)	$77.8 \pm 0.8\%$	$91.4 \pm 0.5\%$	$86.7 \pm 0.5\%$	$96.9 \pm 0.3\%$	$84.4 \pm 0.5\%$	$\mathbf{97.0 \pm 0.2}\%$
Ionosphere(%)	$95.9 \pm 1.2\%$	$36.5 \pm 2.6\%$	$74.5 \pm 0.8\%$	$\mathbf{88.5 \pm 1.0}\%$	$63.4 \pm 1.1\%$	$84.8 \pm 0.8\%$
Pima(%)	$0.9 \pm 0.0\%$	$62.9 \pm 1.1\%$	$41.3 \pm 0.8\%$	$\mathbf{76.8 \pm 0.6}\%$	$32.0 \pm 0.8\%$	$76.1 \pm 0.6\%$
Heart-disease(%)	$43.6 \pm 2.5\%$	$66.5 \pm 1.5\%$	$56.3 \pm 1.4\%$	$\mathbf{84.2 \pm 0.7}\%$	$54.9 \pm 1.4\%$	$83.2 \pm 0.8\%$
Vote(%)	$82.6 \pm 1.3\%$	$84.6 \pm 0.7\%$	$83.9 \pm 0.9\%$	$\mathbf{94.9 \pm 0.4}\%$	$83.8 \pm 0.9\%$	$94.8 \pm 0.4\%$

Table 3.1: Lower bound α, β, and test accuracy compared to MPM in the linear setting.

Dataset	MEMPM				MPM	
	α	β	$\theta\alpha+(1-\theta)\beta$	Accuracy	α	Accuracy
Twonorm(%)	$91.7 \pm 0.2\%$	$91.7 \pm 0.2\%$	$91.7 \pm 0.2\%$	$\mathbf{97.9 \pm 0.1}\%$	$91.7 \pm 0.2\%$	$\mathbf{97.9 \pm 0.1}\%$
Breast(%)	$88.4 \pm 0.6\%$	$90.7 \pm 0.4\%$	$89.9 \pm 0.4\%$	$\mathbf{96.9 \pm 0.2}\%$	$89.9 \pm 0.4\%$	$\mathbf{96.9 \pm 0.3}\%$
Ionosphere(%)	$94.2 \pm 0.8\%$	$80.9 \pm 3.0\%$	$89.4 \pm 0.8\%$	$\mathbf{93.8 \pm 0.4}\%$	$89.0 \pm 0.8\%$	$92.2 \pm 0.4\%$
Pima(%)	$2.6 \pm 0.1\%$	$62.3 \pm 1.6\%$	$41.4 \pm 1.1\%$	$\mathbf{77.0 \pm 0.7}\%$	$32.1 \pm 1.0\%$	$76.2 \pm 0.6\%$
Heart-disease(%)	$47.1 \pm 2.2\%$	$66.6 \pm 1.4\%$	$58.0 \pm 1.5\%$	$\mathbf{83.9 \pm 0.9}\%$	$57.4 \pm 1.6\%$	$83.1 \pm 1.0\%$
Vote(%)	$85.1 \pm 1.3\%$	$84.3 \pm 0.7\%$	$84.7 \pm 0.8\%$	$\mathbf{94.7 \pm 0.5}\%$	$84.4 \pm 0.8\%$	$94.6 \pm 0.4\%$

Table 3.2: Lower bound α, β, and test accuracy compared to MPM with the Gaussian kernel.

Since the lower bounds keep well with the test accuracies in the above experimental results, we do not perform the robust version of both models for the real-world data sets. To see how the robust version works, we generate two classes of Gaussian data. As illustrated in Figure 3.6, the $\mathbf{x}$ data are sampled from the Gaussian distribution with the mean as $[3, 0]^T$ and the covariance as [1 0; 0 3], while the $\mathbf{y}$ data are sampled from another Gaussian distribution with the mean as $[-3, 0]^T$ and the covariance as [3 0; 0 1]. We randomly select 10 points of each class for training and leave the rest points for test from the above synthetic dataset. We present the result in the following.

First, we calculate the corresponding means, $\bar{\mathbf{x}}^0$ and $\bar{\mathbf{y}}^0$, covariance matrices, $\Sigma_{\mathbf{x}}{}^0$ and $\Sigma_{\mathbf{y}}{}^0$ and plug them into the linear MPM and the linear MEMPM. We obtain the MPM decision line (magenta dotted line) with a lower bound (assuming the Gaussian distribution) being 99.1% and the MEMPM decision line (black dash-dot line) with a lower bound as 99.7% respectively. However, for the test set, we only obtain the accuracies 93.0% for MPM and 97.0% for MEMPM, (see Figure 3.6(a)). This obviously violates the lower bound.

Based on our knowledge of the real means and covariance matrices in this example, we set the parameters as

$$\begin{aligned}
\nu_{\mathbf{x}} &= \sqrt{(\bar{\mathbf{x}} - \bar{\mathbf{x}}^0)^T \Sigma_{\mathbf{x}}{}^{-1} (\bar{\mathbf{x}} - \bar{\mathbf{x}}^0)} = 0.046, \\
\nu_{\mathbf{y}} &= \sqrt{(\bar{\mathbf{y}} - \bar{\mathbf{y}}^0)^T \Sigma_{\mathbf{y}}{}^{-1} (\bar{\mathbf{y}} - \bar{\mathbf{y}}^0)} = 0.496, \\
\rho_{\mathbf{x}} &= \|\Sigma_{\mathbf{x}} - \Sigma_{\mathbf{x}}{}^0\|_F = 1.561, \\
\rho_{\mathbf{y}} &= \|\Sigma_{\mathbf{y}} - \Sigma_{\mathbf{y}}{}^0\|_F = 0.972, \\
\nu &= \max(\nu_{\mathbf{x}}, \nu_{\mathbf{y}}),
\end{aligned}$$

We then train the robust linear MPM and the robust linear MEMPM by these parameters and obtain the robust MPM decision line (red dashed line), the robust MEMPM decision line (blue solid line), as seen in Figure 3.6(a). The lower bounds decrease to 87.3% for MPM and 93.2% for MEMPM respectively, but the test accuracies increase to 98.0% for MPM and 100.0% for MEMPM. Obviously, the lower bounds accord with the test accuracies.

Note that in the above, the robust MEMPM also achieves a better performance than the robust MPM. Moreover, $\nu_{\mathbf{x}}$ and $\nu_{\mathbf{y}}$ are not necessarily the same. To see the result of MEMPM when $\nu_{\mathbf{x}}$ and $\nu_{\mathbf{y}}$ are forced to be the same, we train the robust MEMPM again by setting the parameters as $\nu_{\mathbf{x}} = \nu_{\mathbf{y}} = \nu$ as used in MPM. We obtain the corresponding decision line (black dash-dot line) as seen in Figure 3.6(b). The lower bound decreases to 91.0% and the test accuracy decreases to 98.0%. The above example indicates how the robust

MEMPM clearly improves over the standard MEMPM when a bias is incorporated by the inaccurate plug-in estimates and also validates that $\nu_{\mathbf{x}}$ need not be equal to $\nu_{\mathbf{y}}$.

3.5.3 Evaluations of BMPM on Heart-disease Dataset

To demonstrate the advantages of the BMPM model in dealing with biased classifications, we implement BMPM on the Heart-disease dataset, where different treatments for different classes are necessary. The $\mathbf{x}$ class is associated with data with heart diseases, whereas the $\mathbf{y}$ class corresponds to data without heart diseases. Obviously, a bias should be considered for $\mathbf{x}$, since misclassification of an $\mathbf{x}$ case into the opposite class would delay the therapy and is more risky than the other way round. Similarly, we randomly partition data into 90% training and 10% test sets. Also, the width parameter (σ) for the Gaussian kernel is obtained via cross validations over 50 random partitions of the training set. We repeat the above procedures 50 times and report the average results.

By intentionally varying β_0 from 0 to 1, we obtain a series of test accuracies, including the $\mathbf{x}$ accuracy, $\text{TSA}_{\mathbf{x}}$, the $\mathbf{y}$ accuracy $\text{TSA}_{\mathbf{y}}$ for both the linear and Gaussian kernel. For simplicity, we denote the $\mathbf{x}$ accuracy in the linear setting as $\text{TSA}_{\mathbf{x}}(\text{L})$, while others are similarly defined.

The results are summarized in Figure 3.5. Four observations are worth highlighting. First, in both linear and Gaussian kernel settings, the smaller β_0, the higher the test accuracy for $\mathbf{x}$. This indicates a bias can be indeed embedded in the classification boundary for the important class $\mathbf{x}$ by specifying a relatively smaller β_0. In comparison, MPM forces an equal treatment on each class and thus is not suitable for biased classification. Second, the test accuracies for $\mathbf{y}$ and $\mathbf{x}$ are strictly lower bounded by β_0 and α. This shows how a bias can be quantitatively, directly, and rigorously imposed towards the

important class $\mathbf{x}$. Note that again, for other weight-adapting based biased classifiers, the weights themselves lack accurate interpretations and thus cannot rigorously impose a specified bias, i.e., they would try for different weights for a specified bias. Third, when given a prescribed β_0, the test accuracy for $\mathbf{x}$ and its worst-case accuracy α in the Gaussian kernel setting are both increased compared to the corresponding accuracies in the linear setting. Once again, this demonstrates the power of the kernelization. Fourth, we note that β_0 actually contains an upper bound, which is around 90% for the linear BMPM in this dataset. This is reasonable. Observed from (3.11), the maximum β_0, denoted as β_{0_m}, is decided by setting $\alpha = 0$, i.e.,

$$\kappa(\beta_{0_m}) = \max_{\mathbf{w}\neq\mathbf{0}} \quad \frac{1}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}} \quad \text{s.t.} \quad \mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}}) = 1. \tag{3.44}$$

It is interesting noting that when β_0 is set to zero, the test accuracies for $\mathbf{y}$ in the linear and Gaussian settings are both around 50% (see Figure 3.7(b)). This seeming "irrationality" is actually reasonable. We will discuss this in the next section.

3.6 How Tight Is the Bound?

A natural question for MEMPM is, how tight is the worst-case bound? In this section, we present a theoretical analysis in addressing this problem.

In Marshall and Olkin Theory, if we define $\mathcal{S} = \{\mathbf{w}^T\mathbf{y} \geq b\}$, the theorem is changed to:

$$\sup_{\mathbf{y}\sim\{\overline{\mathbf{y}},\Sigma_{\mathbf{y}}\}} \mathbf{Pr}\{\mathbf{w}^T\mathbf{y} \geq b\} = \frac{1}{1+d^2}, \quad \text{with} \quad d^2 = \inf_{\mathbf{w}^T\mathbf{y}\geq\mathbf{b}} (\mathbf{y} - \overline{\mathbf{y}})^T\Sigma_{\mathbf{y}}^{-1}(\mathbf{y} - \overline{\mathbf{y}}) \,.$$

Looking into the above equation and (3.4), for a given hyperplane $\{\mathbf{w}, b\}$,

we can easily obtain

$$\beta = \frac{d^2}{1+d^2} \ . \tag{3.45}$$

Moreover, in [85], a simple closed-form expression for the minimum distance d is derived:

$$d^2 = \inf_{\mathbf{w}^T\mathbf{y} \geq b} (\mathbf{y}-\overline{\mathbf{y}})^T {\Sigma_{\mathbf{y}}}^{-1} (\mathbf{y}-\overline{\mathbf{y}}) = \frac{\max((b-\mathbf{w}^T\overline{\mathbf{y}}),0)}{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}} \ . \tag{3.46}$$

It is easy to see that when the decision hyperplane $(\mathbf{w}, b)$ passes the center $\overline{\mathbf{y}}$, d would be equal to 0 and the worst-case accuracy β would be 0 according to (3.45).

However, if we consider the Gaussian data (which we assume as $\mathbf{y}$ data) in Figure 3.9(a), a vertical line approximating $\overline{\mathbf{y}}$ would achieve about 50% test accuracy. The large gap between the worst-case accuracy and the real test accuracy seems strange. In the following, we construct an example of one-dimensional data to show the inner rationality of this observation. We attempt to provide the worst-case distribution containing the given mean and covariance, while a hyperplane passing its mean achieves a real test accuracy of zero.

Consider one-dimensional data y consist of $N-1$ observations with values as m and one single observation with the value as $\sigma\sqrt{N}+m$. If we calculate the mean and the covariance, we obtain:

$$\overline{y} = m + \frac{\sigma}{\sqrt{N}} \ ,$$
$$\Sigma_y = \frac{N-1}{N}\sigma^2 \ .$$

When N goes to infinity, the above one-dimensional data have the mean as m and the covariance as σ. In this extreme case, a hyperplane passing the mean

will achieve a zero test accuracy, which is exactly the worst-case accuracy given the fixed mean and covariance as m and σ respectively. This example demonstrates the inner rationality of the minimax probability machines.

To further examine the tightness of the worst-case bound in Figure 3.9(a), we vary β from 0 to 1 and plot against β the real test accuracy that a vertical line classifies the $\mathbf{y}$ data by using (3.45). Note that the real accuracy can be calculated as $\Phi(z \leq d)$. This curve is plotted in Figure 3.8. Observed from Figure 3.8, the smaller the worst-case accuracy, the looser it is. On the other hand, if we skew the $\mathbf{y}$ data towards the left side, while simultaneously maintaining the mean and covariance unchanged (see Figure 3.9(b)), even a bigger gap will be generated when β is small; analogically, if we skew the data towards the right side (see Figure 3.9(c)), a tighter accuracy bound will be expected. This finding would mean that only adopting up to the second order moments may not achieve a satisfactory bound. In other words, for a tighter bound, higher order moments such as skewness may need to be considered. This problem of estimating a probability bound based on moments is presented as the (n, k, Ω)-bound problem, which means "finding the tightest bound for n-dimensional variable in the set Ω based on up to the k-th moments." Unfortunately, as proved in [121], it is NP-hard for $(n, k, \mathbb{R}^n)$-bound problems with $k \geq 3$. Thus tightening the bound by simply scaling up the moment order may be intractable in this sense. We may have to exploit other statistical techniques to achieve this goal. Certainly, this deserves a closer examination in the future.

3.7 On the Concavity of MEMPM

We address the issue of the concavity on the MEMPM model in this section. We will demonstrate that, although MEMPM cannot generally guarantee its concavity, there is strong empirical evidence showing that many real-world

problems demonstrates reasonable concavity in MEMPM. Hence, the MEMPM model can be solved successfully by standard optimization methods, e.g., the linear search method proposed in this chapter.

We first present a lemma on BMPM.

Lemma 10 The optimal solution for BMPM is a strictly and monotonically decreasing function with respect to β_0.

Proof: Let the corresponding optimal worst-case accuracies on $\mathbf{x}$ be α_1 and α_2 respectively, when β_{01} and β_{02} are set as the acceptable accuracy levels for $\mathbf{y}$ in BMPM. We will prove that if $\beta_{01} > \beta_{02}$, then $\alpha_1 < \alpha_2$.

This can be proved by considering the contrary case, i.e., we assume $\alpha_1 \geq \alpha_2$. From the problem definition of BMPM, we have:

$$\begin{aligned} &\alpha_1 \geq \alpha_2 \Longrightarrow \kappa(\alpha_1) \geq \kappa(\alpha_2) \\ &\Longrightarrow \frac{1-\kappa(\beta_{01})\sqrt{\mathbf{a_1}^T\Sigma_\mathbf{y}\mathbf{a_1}}}{\sqrt{\mathbf{a_1}^T\Sigma_\mathbf{x}\mathbf{a_1}}} \geq \frac{1-\kappa(\beta_{02})\sqrt{\mathbf{a_2}^T\Sigma_\mathbf{y}\mathbf{a_2}}}{\sqrt{\mathbf{a_2}^T\Sigma_\mathbf{x}\mathbf{a_2}}}\,, \end{aligned} \tag{3.47}$$

where, $\mathbf{w}_1$ and $\mathbf{w}_2$ are the corresponding optimal solutions which maximize $\kappa(\alpha_1)$ and $\kappa(\alpha_2)$ respectively, when β_{01} and β_{02} are specified.

From $\beta_{01} > \beta_{02}$ and (3.47), we have

$$\frac{1-\kappa(\beta_{02})\sqrt{\mathbf{a_1}^T\Sigma_\mathbf{y}\mathbf{a_1}}}{\sqrt{\mathbf{a_1}^T\Sigma_\mathbf{x}\mathbf{a_1}}} > \frac{1-\kappa(\beta_{01})\sqrt{\mathbf{a_1}^T\Sigma_\mathbf{y}\mathbf{a_1}}}{\sqrt{\mathbf{a_1}^T\Sigma_\mathbf{x}\mathbf{a_1}}} \tag{3.48}$$

$$\geq \frac{1-\kappa(\beta_{02})\sqrt{\mathbf{a_2}^T\Sigma_\mathbf{y}\mathbf{a_2}}}{\sqrt{\mathbf{a_2}^T\Sigma_\mathbf{x}\mathbf{a_2}}}\,. \tag{3.49}$$

On the other hand, since $\mathbf{w}_2$ is the optimal solution of $\max_\mathbf{w} \frac{1-\kappa(\beta_{02})\sqrt{\mathbf{w}^T\Sigma_\mathbf{y}\mathbf{w}}}{\sqrt{\mathbf{w}^T\Sigma_\mathbf{x}\mathbf{w}}}$, we have:

$$\frac{1-\kappa(\beta_{02})\sqrt{\mathbf{a_2}^T\Sigma_\mathbf{y}\mathbf{a_2}}}{\sqrt{\mathbf{a_2}^T\Sigma_\mathbf{x}\mathbf{a_2}}} \geq \frac{1-\kappa(\beta_{02})\sqrt{\mathbf{a_1}^T\Sigma_\mathbf{y}\mathbf{a_1}}}{\sqrt{\mathbf{a_1}^T\Sigma_\mathbf{x}\mathbf{a_1}}}\,.$$

This is obviously contradictory to (3.49). ■

From the sequential solving method of MEMPM, we know that MEMPM actually corresponds to a one-dimensional line search problem. More specifically, it further corresponds to maximizing the sum of two functions, namely, $f_1(\beta) + f_2(\beta)$,[6] where $f_1(\beta)$ is determined by the BMPM optimization and $f_2(\beta) = \beta$. According to Lemma 10, $f_1(\beta)$ strictly decreases as β increases. Thus it is strictly pseudo-concave. However, generally speaking, the sum of a pseudo-concave function and a linear function is not necessarily a pseudo-concave function and thus cannot assure that every local optimum is the global optimum. This can be clearly observed in Figure 3.10. In this figure, f_1 is pseudo-concave in three sub-figures; however, the sum $f_1 + f_2$ does not necessarily lead to a pseudo-concave function.

Nevertheless, there is strong empirical evidence showing that for many "well-behaved" real world classification problems, f_1 is overall concave, which results in the concavity of $f_1 + f_2$. This is first verified by the data sets used in this chapter. We shift β from 0 to the corresponding upper bound and plot out α against β in Figure 3.11. It is clearly observed that in all six data sets including both kernel and linear cases, the curves of α against β are overall concave. This motivates us to look further into the concavity of MEMPM. As shown in the following, when two classes of data are "well-separated," f_1 would be concave in the main "interest" region.

We analyze the concavity of $f_1(\beta)$ by imagining that β changes from 0 to 1. In this process, the decision hyperplane moves slowly from $\overline{\mathbf{y}}$ to $\overline{\mathbf{x}}$ according to (3.45) and (3.46). At the same time, $\alpha = f_1(\beta)$ should decrease accordingly. More precisely, if we denote $d_{\mathbf{x}}$ and $d_{\mathbf{y}}$ respectively as the Mahalanobis distances that $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$ are from the associated decision hyperplane with a

[6]For simplicity, we assume θ as 0.5. Since a constant does not influence the concavity analysis, the factor of 0.5 is simply dropped.

specified β, we can formulate the changing of α and β as:

$$\alpha \to \alpha - k_1(d_{\mathbf{x}})\Delta d_{\mathbf{x}},$$
$$\beta \to \beta + k_2(d_{\mathbf{y}})\Delta d_{\mathbf{y}},$$

where $k_1(d_{\mathbf{x}})$ and $k_2(d_{\mathbf{y}})$ can be considered as the changing rate of α and β when the hyperplane lies $d_{\mathbf{x}}$ distance far away from $\overline{\mathbf{x}}$ and $d_{\mathbf{y}}$ distance far away from $\overline{\mathbf{y}}$ respectively. We consider the changing of α against β, namely, f_1':

$$f_1' = \frac{-k_1(d_{\mathbf{x}})\Delta d_{\mathbf{x}}}{k_2(d_{\mathbf{y}})\Delta d_{\mathbf{y}}}.$$

If we consider $d_{\mathbf{x}}$ and $\Delta d_{\mathbf{y}}$ insensitively change against each other or change with a proportional rate, i.e., $\Delta d_{\mathbf{x}} \approx c\Delta d_{\mathbf{y}}$ (c is a positive constant) as the decision hyperplane moves, the above equation can be further written as

$$f_1' = c\frac{-k_1(d_{\mathbf{x}})}{k_2(d_{\mathbf{y}})}.$$

Lemma 11 (1) If $d_{\mathbf{y}} \geq 1/\sqrt{3}$ or the corresponding $\beta \geq 0.25$, $k_2(d_{\mathbf{y}})$ decreases as $d_{\mathbf{y}}$ increases.
(2) If $d_{\mathbf{x}} \geq 1/\sqrt{3}$ or the corresponding $\alpha \geq 0.25$, $k_1(d_{\mathbf{x}})$ decreases as $d_{\mathbf{x}}$ increases.

Proof: Since (1) and (2) are actually very similar statements, we only prove (1). $k_2(d)$ is actually the first order derivative of $\frac{d^2}{1+d^2}$ according to (3.45). We consider the first order derivative of $k_2(d)$ or the second order derivative of $\frac{d^2}{1+d^2}$. It is easily verified that $(\frac{d^2}{1+d^2})'' \leq 0$ when $d \geq 1/\sqrt{3}$. This is also illustrated in Figure 3.13. According to the definition of the second derivative, we immediately obtain the lemma. Note that $d \geq 1/\sqrt{3}$ corresponds to $\beta \geq 0.25$. Thus the condition can be also replaced by $\beta \geq 0.25$.

■

In the above procedure, $d_{\mathbf{y}}$, β increase and $d_{\mathbf{x}}$, α decrease, as the hyperplane moves towards $\overline{\mathbf{x}}$. Therefore, according to Lemma 11, $k_1(d_{\mathbf{x}})$ increases while $k_2(d_{\mathbf{y}})$ decreases when $\alpha, \beta \in [0.25,\ 1)$. This shows that f_1' is getting smaller as the hyperplane moves towards $\overline{\mathbf{x}}$. In other words, f_1'' would be less than 0 and thus is concave when $\alpha, \beta \in [0.25,\ 1)$. It should be noted that in many well-separated real world data sets, the optimal α and β will be greater than 0.25 with a high possibility, since to achieve good performance, the worst-case accuracies are naturally required to be greater than a smaller amount, e.g., 0.25. This is observed in the data sets used in the chapter. All the data sets except Pima attain their optimums satisfying this constraint. For Pima, the overall accuracy is relatively lower, which implies two classes of data in this dataset appear to largely overlap with each other.[7]

An illustration can be also seen in Figure 3.12. We generate two classes of Gaussian data with $\overline{\mathbf{x}} = [0,\ 0]^T$, $\overline{\mathbf{y}} = [L,\ 0]^T$, and $\Sigma_{\mathbf{x}} = \Sigma_{\mathbf{y}} = [1,\ 0; 0,\ 1]$. The prior probability for each data is set as an equal value 0.5. We plot the curves of $f_1(\beta)$ and $f_1(\beta) + \beta$ when L is set as different values. It is observed that when two classes of data largely overlap with each other, for example in Figure 3.12(a) with $L = 1$, the optimal solution of MEMPM lies in the small-value range of α and β, which is usually not concave. On the other hand, (b), (c), and (d) show that when two classes of data are well-separated, the optimal solutions lie in the region with $\alpha, \beta \in [0.25,\ 1)$, which is often concave.

Note that, in the above, we make an assumption that as the decision hyperplane moves, $d_{\mathbf{x}}$ and $d_{\mathbf{y}}$ change at an approximately fixed proportional rate. From the definition of $d_{\mathbf{x}}$ and $d_{\mathbf{y}}$, this assumption implies that $\mathbf{w}$, the direction of the optimal decision hyperplane, is insensitive to β. This assumption does not hold in all cases; however, observed from the geometrical interpretation of

[7]It is observed, even for Pima, the proposed solving algorithm is still successful, since α is approximately linear as shown in Figure 3.11. Moreover, due to the fact that the slope of α is slightly greater than -1, the final optimum naturally leads β achieves its maximum.

MEMPM, for those data with isotropic or not significantly anisotropic $\Sigma_{\mathbf{x}}$ and $\Sigma_{\mathbf{y}}$, $\mathbf{w}$ would be indeed insensitive to β.

We summarize the above analysis into the following proposition.

Proposition 12 Assuming (1) two classes of data are well-separated and (2) $d_{\mathbf{x}}$ and $d_{\mathbf{y}}$ change at an approximately fixed proportional rate as the optimal decision hyperplane (associated with a specified β) moves, the one-dimensional line search problem of MEMPM is often concave in the range of $\alpha, \beta \in [0.25, 1)$ and will often attain its optimum in this range. Therefore the proposed solving method leads to a satisfactory solution.

Remarks. As demonstrated in the above, although the MEMPM is often overall concave in real world tasks, there exist cases that the MEMPM optimization problem is not concave. This may lead to the case that the solved local optimum, based on the SBMPM method, is not the global optimum. In these instances, we may need carefully choose the initial starting point. In addition, the physical interpretation of β as the worst-case accuracy, may make it relatively easy to choose a suitable initial value. For example, we can set the initial value by using the information obtained from prior domain knowledge.

3.8 Limitations and Future Work

In this section, we present the limitations and future work.

First, although MEMPM achieves better performance to MPM, its sequential optimization of Biased Minimax Probability Machine may cost more training time than MPM. In our experiments, the MEMPM needs to solve 5-15 BMPM optimizations on the average. Supposing that BMPM is solved based on Conjugate Gradient Methods (with a worst-case time complexity in the same order as MPM), the MEMPM would be 5-15 times as expensive as MPM. Although in pattern recognition tasks, especially in off-line classifications, effectiveness is often more important than efficiency, expensive time-cost

presents one of the main limitations of the MEMPM model, in particular for large scale data sets with millions of samples. To solve this problem, one possible direction is to reduce those redundant points, which actually make less contributions to the classification. In this way, the problem dimension (in the kernelization) would be greatly decreased and therefore may help in reducing the computational time required. Another possible direction is to exploit some techniques to decompose the Gram matrix (as is done in SVM) and to develop some specialized optimization procedures for MEMPM. Undoubtedly, speeding up the algorithm will be a highly worthy topic in the future.

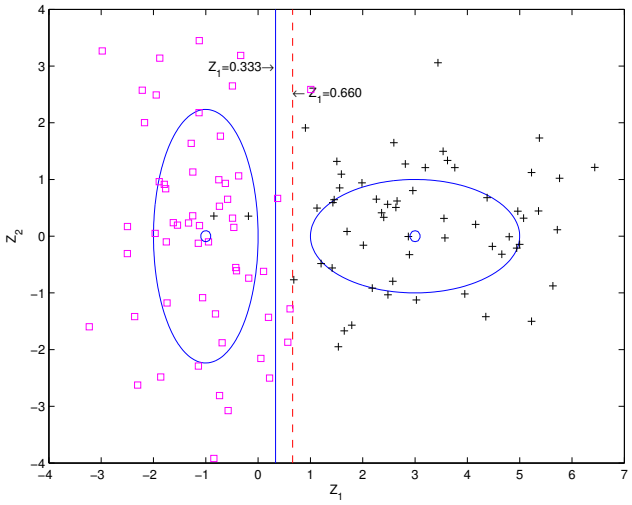
Second, as a generalized model, MEMPM actually incorporates some other variations. For example, when the prior probability (θ) cannot be estimated reliably (e.g., in sparse data), maximizing $\alpha + \beta$, namely the sum of the accuracies or the difference between true positive and false positive, would be considered . This type of approaches is widely used in pattern recognition field, e.g., in medical diagnosis [52] and in graph detection, especially line detection and arc detection, where it is called Vector Recovery Index [35, 93] . Moreover, when there are domain experts at hand, a variation of MEMPM, namely, the maximization of $C_{\mathbf{x}}\alpha + C_{\mathbf{y}}\beta$ may be used, where $C_{\mathbf{x}}$ ($C_{\mathbf{y}}$) is the cost of a misclassification of $\mathbf{x}$ ($\mathbf{y}$) obtained from experts. Exploring these variations in some specific domains is thus a valuable direction in the future (we actually will discuss these variations as criteria for biased or imbalanced learning in Chapter 5).

Third, [85] has built up a connection between MPM and SVM from the perspective of the margin definition, i.e., MPM corresponds to finding the hyperplane with the maximal margin from the class center. Nevertheless, some deeper connections need to be investigated, e.g., how is the bound of the MEMPM related to the generation bound of SVM? More recently, [64] and also the next chapter have disclosed the relationship between them from either a local or global viewpoint of data. It is particularly useful to look into these

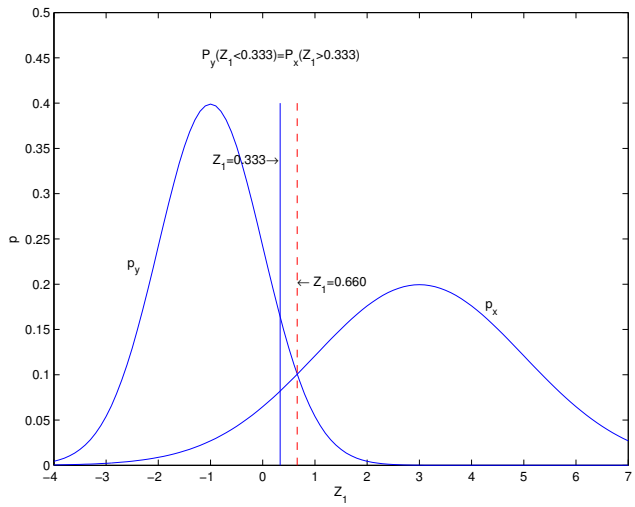
links and explore their further connections in the future.

3.9 Summary

In this chapter, we have proposed a novel global learning model, named Minimum Error Minimax Probability Machine. By minimizing the upper bound of the Bayes error of future data points, our model derives the distribution-free Bayes optimal hyperplane in the worst-case setting. This thus distinguish itself from the traditional global learning approaches or more particular from traditional Bayes optimal classsifers. More importantly, we have shown that, the worst-case Bayes optimal hyperplane derived by MEMPM becomes the true Bayes optimal hyperplane, when some conditions are satisfied, e.g., when a Gaussian distribution is assumed on data. We have shown how to exploit Mercer kernels in this setting to derived a nonlinear classification boundary. We also have demonstrated that how a robust framework can be introduced to make solid the foundation of the proposed model. Moreover, we have demonstrated that this novel model permits an explicit accuracy bound on future data theoretically and validate this proposition empirically as well. We have evaluated our algorithms on both synthetic data sets and real-world benchmark data sets. The performance of MEMPM is demonstrated to outperform MPM, a comparable model with SVM.

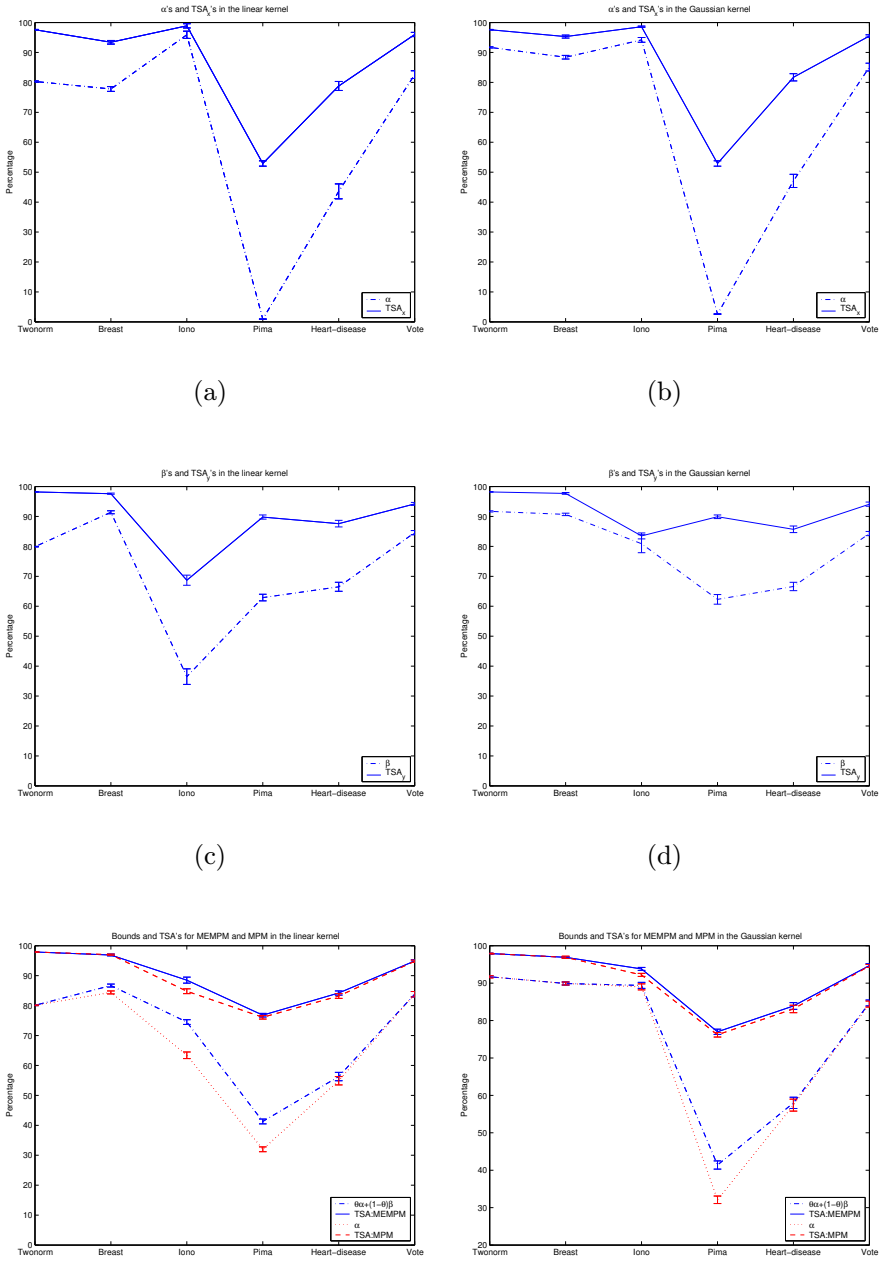


(a)



(b)

Figure 3.4: An evaluation of MEMPM and MPM on a synthetic dataset. The decision hyperplane derived from MEMPM (the dashed red line) exactly corresponds to the optimal threshholding point, i.e., the intersection point, while the decision hyperplane given by MPM (the solid blue line) corresponds to the point in which, two error rates for two classes of data are equal.

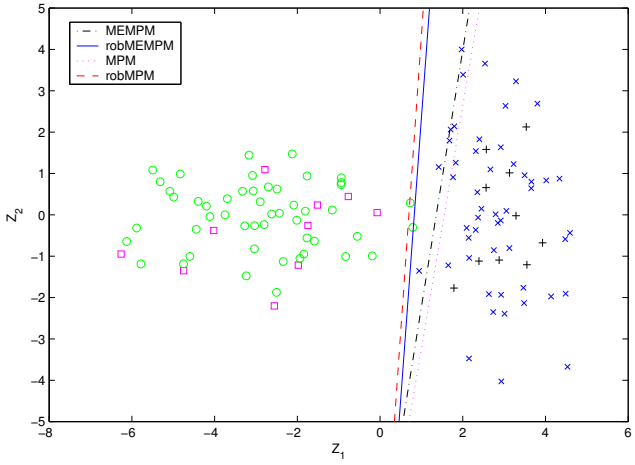


(a) (b)

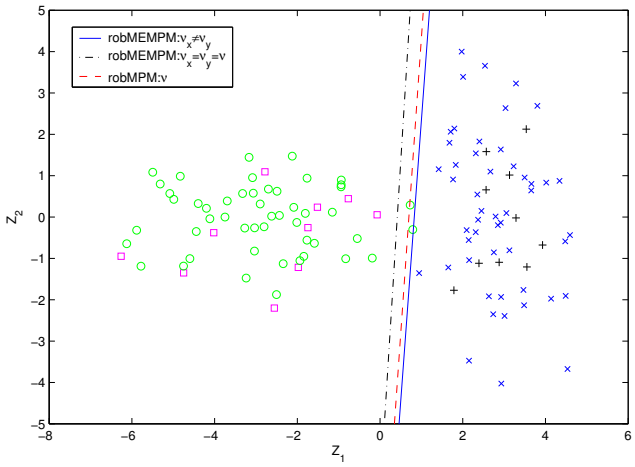
(c) (d)

(e) (f)

Figure 3.5: Empirical evaluations on bounds and test set accuracies of MEMPM. The test accuracies including $\text{TSA}_{\mathbf{x}}$ (for the class $\mathbf{x}$), $\text{TSA}_{\mathbf{y}}$ (for the class $\mathbf{y}$), and the overall accuracies TSA are all greater than their corresponding accuracy bounds both in MPM and MEMPM. This demonstrates how the accuracy bound can serve as the performance indicator on future data.

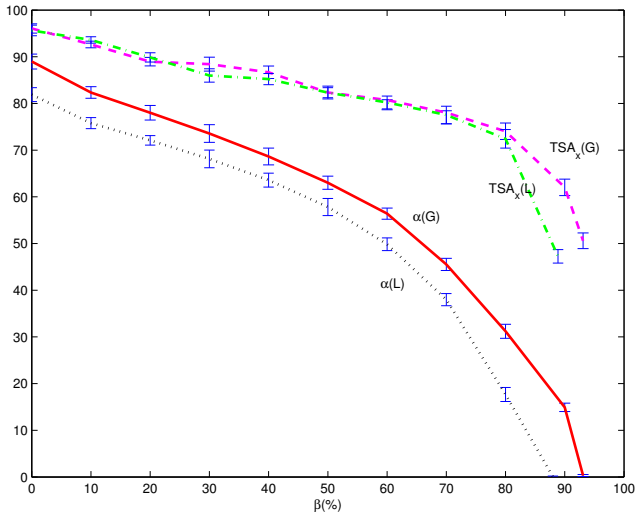


(a)

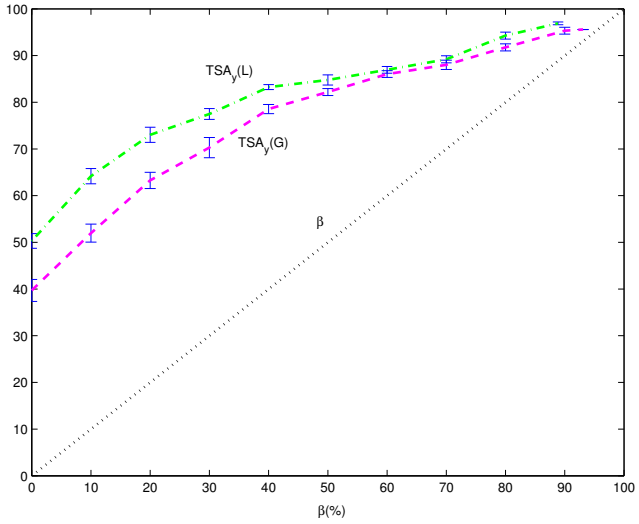


(b)

Figure 3.6: An example in $\mathbb{R}^2$ demonstrates the results of robust versions of MEMPM and MPM. Training points are indicated with black +'s for the class $\mathbf{x}$ and magenta □'s for class $\mathbf{y}$. Test points are represented by blue ×'s for class $\mathbf{x}$ and by green o's for the class $\mathbf{y}$. In (a), the robust MEMPM outperforms both MEMPM and the robust MPM. In (b), the robust MEMPM with $\nu_{\mathbf{x}} \neq \nu_{\mathbf{y}}$ outperforms the robust MEMPM with $\nu_{\mathbf{x}} = \nu_{\mathbf{y}}$.



(a)



(b)

Figure 3.7: Bounds and real accuracies for BMPM in Heart-disease data set. With β_0 varying from 0 to 1, the real accuracies are lower bounded by the worst-case accuracies. In addition, α(G) is above α(L), which shows the power of the kernelization.

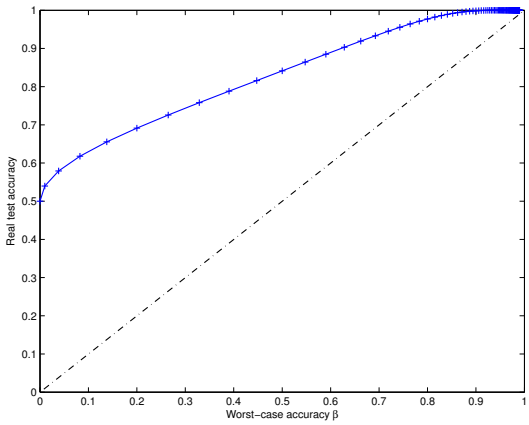
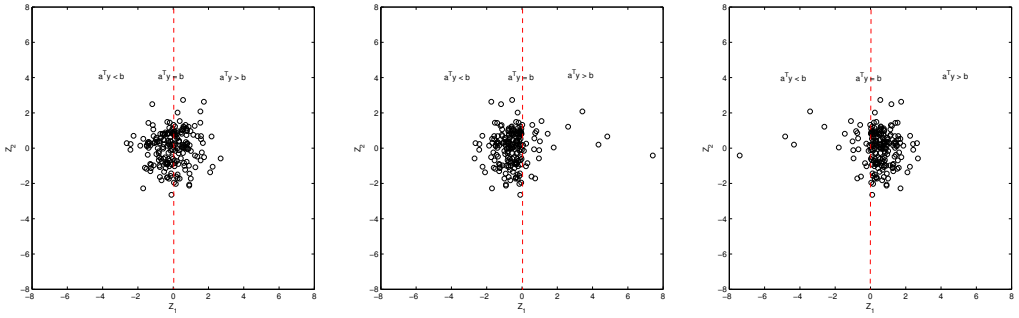
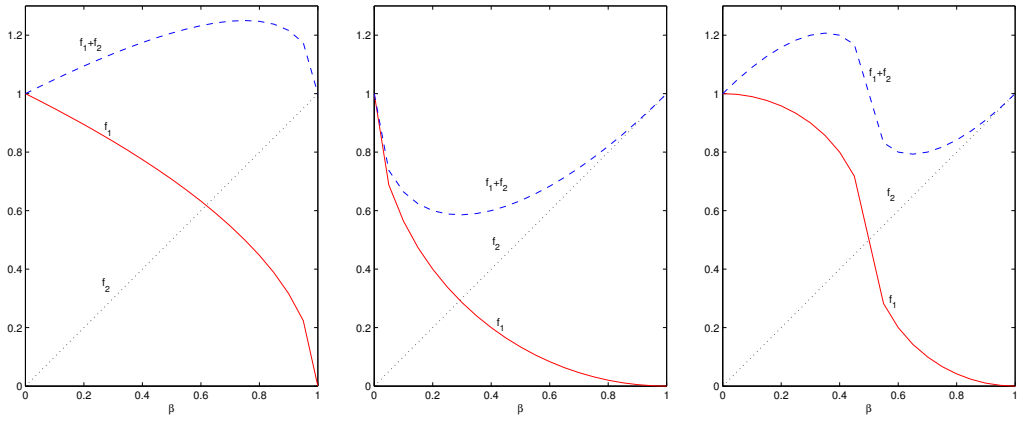


Figure 3.8: Theoretical comparison between the worst-case accuracy and the real test accuracy for the Gaussian data in Figure 3.9(a).



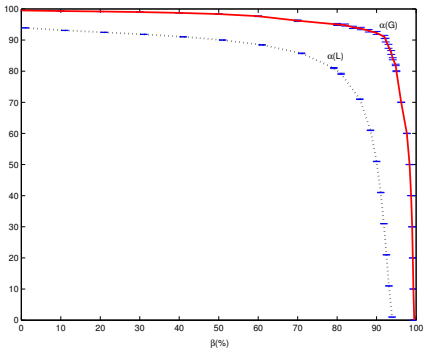
(a) (b) (c)

Figure 3.9: Three two-dimensional data with the same means and covariances but with different skewness. The worst-case accuracy bound of (a) is tighter than that of (b) and looser than that of (c).

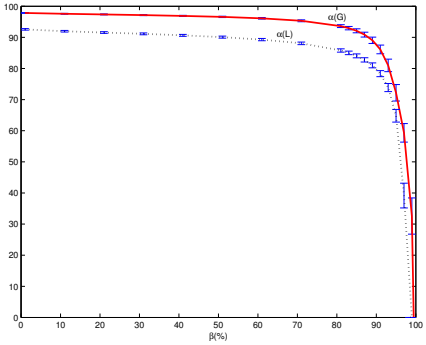


(a) (b) (c)

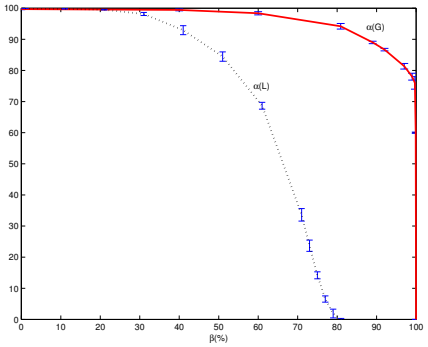
Figure 3.10: The sum of a pseudo-concave function and a linear function is not necessarily a concave function. In (a), $f_1 + f_2$ is a concave function, however in (b) and (c) it is not.



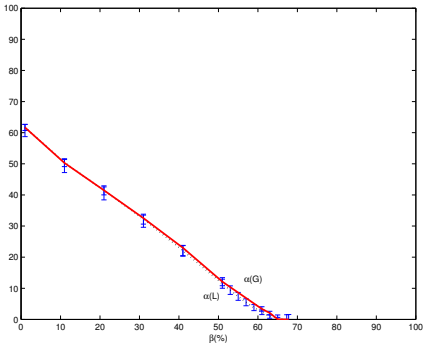
(a) Twonorm



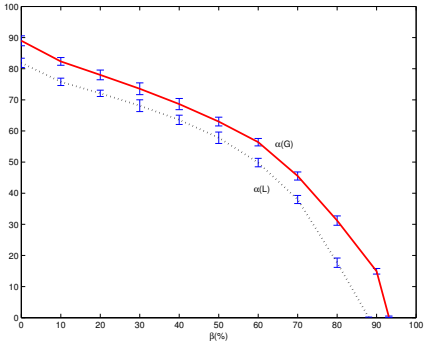
(b) Breast



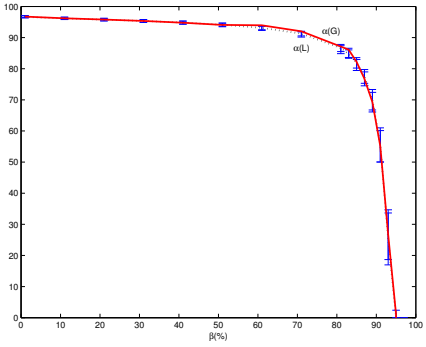
(c) Ionosphere



(d) Pima



(e) Heart-disease



(f) Vote

Figure 3.11: The curves of α against β (f_1) are all concave-like in the data sets used in this chapter.

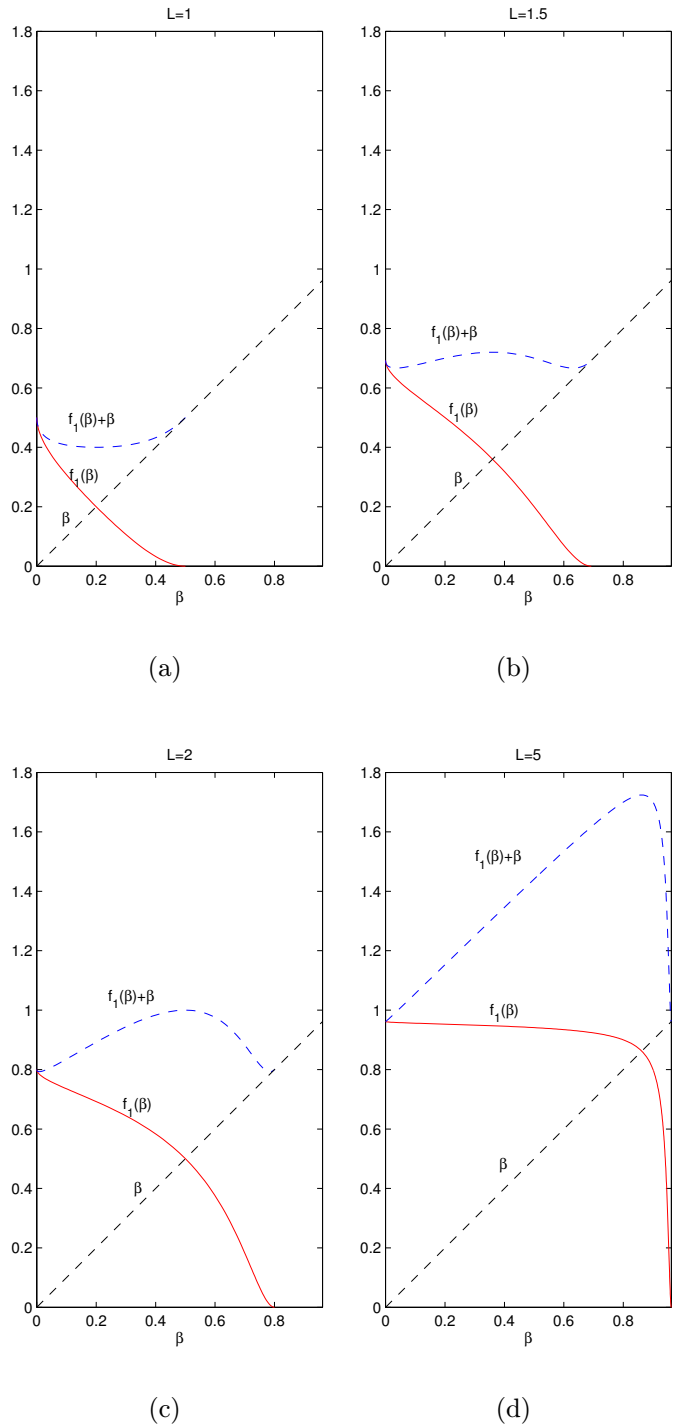


Figure 3.12: An illustration of the concavity of the MEMPM. Subfigure (a) shows that when two classes of data overlap largely with each other, the optimal solution of MEMPM lies in the small-value range of α and β, which is usually not concave. (b), (c), and (d) show that when two classes of data are well-separated, the optimal solutions lie in the region with $\alpha, \beta \in [0.25,\ 1)$, which is often concave.

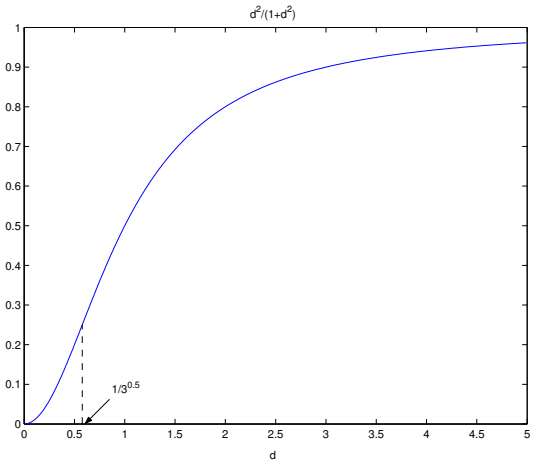


Figure 3.13: The curve of $d^2/(1+d^2)$. This function is concave when $d \geq 1/\sqrt{3}$.

Chapter 4

Learning locally and Globally: Maxi-Min Margin Machine

The proposed MEMPM model obtains the decision hyperplane by using only global information, e.g., the mean and covariance matrices. However, although these moments can be more reliably obtained than estimating the distribution, they may still be inaccurate in many cases, e.g., when the data are very sparse.

Recently, local learning methods, especially large margin classifiers [141] have attracted much interest in the community of machine learning and pattern recognition. Support Vector Machine (SVM) [154], the most famous one of them, represents a state-of-the-art classifier. The essential point of SVM is to find a linear separating hyperplane, which achieves the maximal margin among different classes of data. Furthermore, one can extend SVM to build nonlinear separating decision hyperplanes by exploiting kernelization techniques.

These methods do not try to summarize any global information beforehand, but to focus on obtaining the decision hyperplane in a "local" way. For example, in SVM, the decision boundary is exclusively determined by some critical points, which are called support vectors, whereas all other points are totally irrelevant to this hyperplane. Although this scheme is both theoretically and empirically demonstrated to be powerful, it actually discards the global information of data.

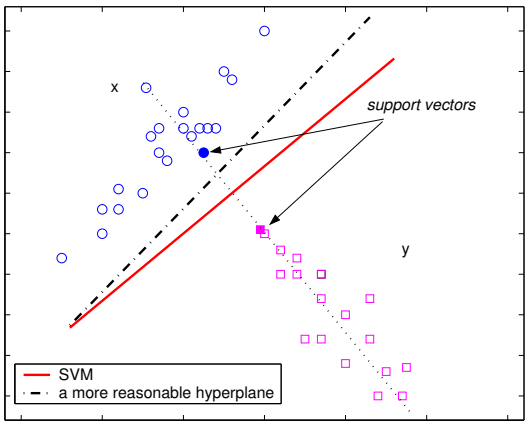


Figure 4.1: A decision hyperplane with considerations of both local and global information.

An illustration example can be seen in Figure 4.1. In this figure, the classification boundary is intuitively observed to be mainly determined by the dotted axis, i.e., the long axis of the **y** data (represented by □'s) or the short axis of the **x** data (represented by ○'s). Moreover, along this axis, the **y** data are more possible to scatter than the **x** data, since **y** contains a relatively larger variance in this direction. Noting this "global" fact, a good decision hyperplane seems reasonable to lie closer to the **x** side (see the dash-dot line). However, SVM ignores this kind of "global" information, i.e., the statistical trend of data occurrence: the derived SVM decision hyperplane (the solid line) lies unbiasedly right in the middle of two "local" points (the support vectors).[1]

Aiming to construct classifiers both locally and globally, we propose the Maxi-Min Margin Machine (M^4) in this chapter. We will attempt to combine into the local learning the global information, i.e., the covariance information, which can represent the data trend. Moreover, as this model also contains the properties of local learning, it will naturally neutralize the impact when the global information is inaccurate.

As we show later, one critical contribution of this novel model is that

[1]This figure has appeared earlierin Chapter 2. However, for the purpose of self-containing for each chapter, we still present it here.

M^4 actually presents a unified model of SVM and another recently-proposed promising model Minimax Probability Machine (MPM) [85]. Moreover, based on our proposed local and global view of data, another popular model, Fisher Discriminant Analysis (FDA) [47] can also be interpreted as its special case.

Another good feature of the M^4 model is that it can be cast as a sequential Conic Programming problem [124], or more specifically, a sequential Second Order Cone Programming (SOCP) problem [94, 112, 79], which thus can be practically solved in polynomial time. In addition, with incorporating the global information, a reduction method is proposed for decreasing the computation time of this new model.

The third important feature of our proposed model is that, the kernelization methodology is also applicable for this formulation. This thus generalizes the linear M^4 into a more powerful classification approach, which can derive nonlinear decision boundaries.

The rest of this chapter is organized as follows. In the next section, we introduce the M^4 model in detail, including its model definition, the geometrical interpretation, connections with other models, and the associated solving methods. In Section 4.2, we derive a generation bound for the M^4 model. In Section 4.3, we develop a reduction method to remove redundant points for decreasing the computation time. In Section 4.4, we exploit the kernelization trick to extend M^4 into nonlinear classification tasks. In Section 4.5, we evaluate this novel model on both synthetic data sets and real world benchmark data sets. In Section 4.6, we make discussions on the M^4 model and also present future work. Finally, we conclude this chapter in Section 4.7. This work can be also seen in [64] [66] for a short version .

4.1 Maxi-Min Margin Machine

In the following, we first, for the purpose of clarity, divide M^4 into separable and nonseparable categories, and then introduce the corresponding hard-margin M^4 and soft-margin M^4 sequently. In this section, we will also establish the connections of the M^4 model with other large margin classifiers including SVM, MPM, FDA, and Mininum Error Minimax Probability Machine (MEMPM) [67].

4.1.1 Separable Case

Assuming the classification samples are separable, we first introduce the model definition and the geometrical interpretation. We then transform the model optimization problem into a sequential SOCP problem and discuss the detailed optimization method.

Problem Definition

Only two-category classification tasks are considered in this chapter. Let a training data set contain two classes of samples, represented by $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_j \in \mathbb{R}^n$ respectively, where $i = 1, 2, \ldots, N_{\mathbf{x}}$, $j = 1, 2, \ldots, N_{\mathbf{y}}$. The basic task here can be informally described to find a suitable hyperplane $f(\mathbf{z}) = \mathbf{w}^T\mathbf{z} + b$ separating two classes of data as robustly as possible ($\mathbf{w} \in \mathbb{R}^n \backslash \{\mathbf{0}\}$, $b \in \mathbb{R}$, and $\mathbf{w}^T$ is the transpose of $\mathbf{w}$). Future data points $\mathbf{z}$ for which $f(\mathbf{z}) \geq 0$ are then classified as the class $\mathbf{x}$; otherwise, they are classified as the class $\mathbf{y}$.

The formulation for M^4 can be written as:

$$\max_{\rho, \mathbf{w} \neq \mathbf{0}, b} \quad \rho \quad \text{s.t.} \tag{4.1}$$

$$\frac{(\mathbf{w}^T\mathbf{x}_i + b)}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}} \geq \rho, \quad i = 1, 2, \ldots, N_{\mathbf{x}}\ , \tag{4.2}$$

$$\frac{-(\mathbf{w}^T\mathbf{y}_j + b)}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}} \geq \rho, \quad j = 1, 2, \ldots, N_{\mathbf{y}}\ , \tag{4.3}$$

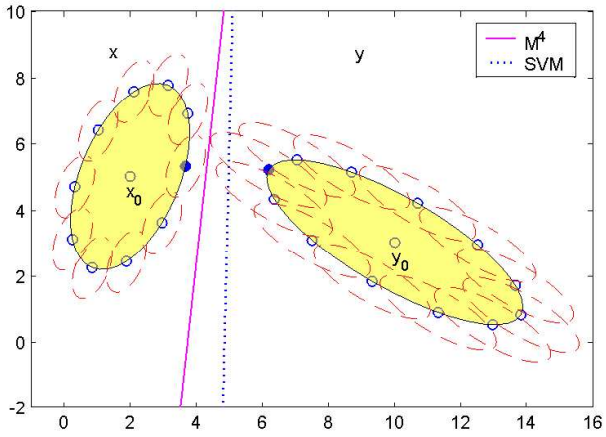


Figure 4.2: A geometric interpretation of M^4. The M^4 hyperplane corresponds to the tangent line (the solid magenta line) of two small dashed ellipsoids centered at the support vectors (the local information) and shaped by the corresponding covariances (the global information). It is thus more reasonable than SVM (the dotted line).

where $\Sigma_{\mathbf{x}}$ and $\Sigma_{\mathbf{y}}$ refer to the covariance matrices of the $\mathbf{x}$ and the $\mathbf{y}$ data, respectively.

This model tries to *maximize* the margin defined as the *minimum* Mahalanobis distance for all training samples,while simultaneously classifying all the data correctly. Compared to SVM, M^4 incorporates the data information in a global way; namely, the covariance information of data or the statistical trend of data occurrence is considered, while SVMs, including l_1-SVM [165] and l_2-SVM [153] (l_p-SVM means the "p-norm" distance-based SVM) [141], simply discard this information or consider the same covariance for each class.

Geometrical Interpretation

A geometrical interpretation of M^4 can be seen in Figure 4.2. In this figure, the $\mathbf{x}$ data are represented by the inner ellipsoid on the left side with its center as $\mathbf{x}_0$, while the $\mathbf{y}$ data are represented by the inner ellipsoid on the right side with its center as $\mathbf{y}_0$. It is observed that these two ellipsoids contain

unequal covariances or risks of data occurrence. However, SVM does not consider this global information: its decision hyperplane (the dotted blue line) locates unbiasedly in the middle of two support vectors (filled points). In comparison, M^4 defines the margin as a Maxi-Min Mahalanobis distance, which thus constructs a decision plane (the solid magenta line) with considerations of both the local and global information: the M^4 hyperplane corresponds to the tangent line of two dashed ellipsoids centered at the support vectors (the local information) and shaped by the corresponding covariances (the global information).

Optimization Method

In the following, we propose the optimization method for the M^4 model. We will demonstrate that the above problem can be cast as a sequential Conic Programming problem, or more specifically, a sequential SOCP problem.

Our strategy is based on the "Divide and Conquer" technique. One may note that in the optimization problem of M^4, if ρ is fixed to a constant ρ_n, the problem is exactly changed to "conquer" the problem of checking whether the constraints of (4.2) and (4.3) can be satisfied. Moreover, as will be demonstrated shortly, this "checking" procedure can be stated as an SOCP problem. Thus the problem now becomes how ρ is set, which we can use "divide" to handle: if the constraints are satisfied, we can increase ρ_n accordingly; otherwise, we decrease ρ_n.

We detail this solving technique in the following two steps:

1. **Divide:** Set $\rho_n = (\rho_0 + \rho_m)/2$, where ρ_0 is a feasible ρ, ρ_m is an infeasible ρ, and $\rho_0 \leq \rho_m$.

2. **Conquer:** Call the Modified Second Order Cone Programming (MSOCP) procedure elaborated in the following to check whether ρ_n is a feasible ρ. If yes, set $\rho_0 = \rho_n$; otherwise, set $\rho_m = \rho_n$;

In the above, if a ρ satisfies the constraints of (4.2) and (4.3), we call it a feasible ρ; otherwise, we call it an infeasible ρ. These two steps are iterated until $|\rho_0 - \rho_m|$ is less than a small positive value.

We propose the following Theorem showing that the MSOCP procedure, namely, the checking problem with ρ fixed to a constant ρ_n, is solvable by casting it as an SOCP problem.

Theorem 13 The problem of checking whether there exist a $\mathbf{w}$ and a b satisfying the following two sets of constraints (4.4) and (4.5) can be transformed as an SOCP problem, which can be solved in polynomial time,

$$(\mathbf{w}^T\mathbf{x}_i + b) \geq \rho_n\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}},\ i = 1, \ldots, N_{\mathbf{x}}\ , \tag{4.4}$$

$$-(\mathbf{w}^T\mathbf{y}_j + b) \geq \rho_n\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}},\ j = 1, \ldots, N_{\mathbf{y}}\ . \tag{4.5}$$

Proof: Introducing dummy variables $\boldsymbol{\tau}$, we rewrite the above checking problem into an equivalent optimization problem:

$$\begin{aligned}
&\max_{\mathbf{w}\neq\mathbf{0},b,\boldsymbol{\tau}} \quad \{\min_{k=1}^{N_{\mathbf{x}}+N_{\mathbf{y}}} \quad \boldsymbol{\tau}_k\} \quad \text{s.t.} \\
&(\mathbf{w}^T\mathbf{x}_i + b) \geq \rho_n\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}} - \boldsymbol{\tau}_i, \\
&-(\mathbf{w}^T\mathbf{y}_j + b) \geq \rho_n\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}} - \boldsymbol{\tau}_{j+N_{\mathbf{x}}}\ ,
\end{aligned}$$

where $i = 1, \ldots, N_{\mathbf{x}}$ and $j = 1, \ldots, N_{\mathbf{y}}$.

By checking whether the minimum $\boldsymbol{\tau}_k$ at the optimum point is positive, we can know whether the constraints of (4.2) and (4.3) can be satisfied. If we go further, we can introduce another dummy variable and transform the above

problem into an SOCP problem:

$$\begin{aligned}
&\max_{\mathbf{w}\neq\mathbf{0},b,\boldsymbol{\tau},\eta} \quad \eta \quad \text{s.t.} \\
&(\mathbf{w}^T\mathbf{x}_i + b) \geq \rho_n\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}} - \boldsymbol{\tau}_i \ , \\
&-(\mathbf{w}^T\mathbf{y}_j + b) \geq \rho_n\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}} - \boldsymbol{\tau}_{j+N_{\mathbf{x}}} \ , \\
&\eta \leq \boldsymbol{\tau}_k \ ,
\end{aligned}$$

where $i = 1, \ldots, N_{\mathbf{x}}$, $j = 1, \ldots, N_{\mathbf{y}}$, and $k = 1, \ldots, N_{\mathbf{x}} + N_{\mathbf{y}}$. By checking whether the optimal η is greater than 0, we can immediately know whether there exist a $\mathbf{w}$ and a b satisfying the constraints of (4.2) and (4.3). Moreover, the above optimization is easily verified to be the standard SOCP form, since the optimization function is a linear form and the constraints are either linear or the typical second order conic constraints. ■

Remarks. In practice, many SOCP programs, e.g., Sedumi [144], provide schemes to directly handle the above checking procedure. It thus need not introduce dummy variables as what we have done in the proof.

We now analyze the time complexity of M^4. As indicated in [94], if the SOCP is solved based on interior-point methods, it contains a worst-case complexity of $O(n^3)$. If we denote the range of feasible ρ's as $L = \rho_{max} - \rho_{min}$ and the required precision as ε, then the number of iterations for M^4 is $\log(L/\varepsilon)$ in the worst case. Adding the cost of forming the system matrix (constraint matrix), which is $O(Nn^3)$ (N represents the number of training points), the total complexity would be $O(\log(L/\varepsilon)n^3 + Nn^3) \approx O(Nn^3)$, which is relatively large but can still be solved in polynomial time.[2]

[2]Note that the system matrix needs to be formed only once.

4.1.2 Connections with Other Models

In this section, we establish connections between M^4 and other models. We show that SVM and MPM are actually special cases of our model. Moreover, FDA can be interpreted and extended according to our local and global views of data.

Connection with Minimax Probability Machine

If one expands the constraints of (4.2) and add all of them together, one can immediately obtain the following:

$$\begin{aligned} & \mathbf{w}^T \sum_{i=1}^{N_{\mathbf{x}}} \mathbf{x}_i + N_{\mathbf{x}} b \geq N_{\mathbf{x}} \rho \sqrt{\mathbf{w}^T \Sigma_{\mathbf{x}} \mathbf{w}} \,, \\ \Rightarrow \quad & \mathbf{w}^T \overline{\mathbf{x}} + b \geq \rho \sqrt{\mathbf{w}^T \Sigma_{\mathbf{x}} \mathbf{w}} \,, \end{aligned} \tag{4.6}$$

where $\overline{\mathbf{x}}$ denotes the mean of the $\mathbf{x}$ training data.

Similarly, from (4.3) one can obtain:

$$\begin{aligned} & -(\mathbf{w}^T \sum_{j=1}^{N_{\mathbf{y}}} \mathbf{y}_j + N_{\mathbf{y}} b) \geq N_{\mathbf{y}} \rho \sqrt{\mathbf{w}^T \Sigma_{\mathbf{y}} \mathbf{w}} \,, \\ \Rightarrow \quad & -(\mathbf{w}^T \overline{\mathbf{y}} + b) \geq \rho \sqrt{\mathbf{w}^T \Sigma_{\mathbf{y}} \mathbf{w}} \,, \end{aligned} \tag{4.7}$$

where $\overline{\mathbf{y}}$ denotes the mean of the $\mathbf{y}$ training data.

Adding (4.6) and (4.7), one can obtain:

$$\begin{aligned} & \max_{\rho, \mathbf{w} \neq \mathbf{0}} \quad \rho \quad \text{s.t.} \\ & \mathbf{w}^T (\overline{\mathbf{x}} - \overline{\mathbf{y}}) \geq \rho (\sqrt{\mathbf{w}^T \Sigma_{\mathbf{x}} \mathbf{w}} + \sqrt{\mathbf{w}^T \Sigma_{\mathbf{y}} \mathbf{w}}) \,. \end{aligned} \tag{4.8}$$

The above optimization is exactly the MPM optimization [85]. Note, however, that the above procedure cannot be reversed. This means the MPM is a

special case of M^4.

Remarks. In MPM, since the decision is completely determined by the global information, namely, the mean and covariance matrices [85],[3] to assure an accurate performance, the estimates of mean and covariance matrices need to be reliable. However, it cannot always be the case in real world tasks. On the other hand, M^4 seems to solve this problem in a natural way, because the impact caused by inaccurately estimated mean and covariance matrices can be neutralized by utilizing the local information, namely by satisfying those constraints of (4.2) and (4.3) for each local data point. This is also demonstrated in the later experiment.

Connection with Support Vector Machine

If one assumes $\Sigma_{\mathbf{x}} = \Sigma_{\mathbf{y}} = \Sigma$, the optimization of M^4 can be changed as:

$$\begin{aligned} \max_{\rho,\mathbf{w}\neq\mathbf{0},b} \quad & \rho \quad \text{s.t.} \\ & (\mathbf{w}^T\mathbf{x}_i + b) \geq \rho\sqrt{\mathbf{w}^T\Sigma\mathbf{w}}\ , \\ & -(\mathbf{w}^T\mathbf{y}_j + b) \geq \rho\sqrt{\mathbf{w}^T\Sigma\mathbf{w}}\ , \end{aligned}$$

where $i = 1, \ldots, N_{\mathbf{x}}$ and $j = 1, \ldots, N_{\mathbf{y}}$.

Observing that the magnitude of $\mathbf{w}$ will not influence the optimization, without loss of generality, one can further assume $\rho\sqrt{\mathbf{w}^T\Sigma\mathbf{w}} = 1$. Therefore the optimization can be changed as:

$$\begin{aligned} \min_{\mathbf{w}\neq\mathbf{0},b} \quad & \mathbf{w}^T\Sigma\mathbf{w} \quad \text{s.t.} && (4.9) \\ & (\mathbf{w}^T\mathbf{x}_i + b) \geq 1\ , && (4.10) \\ & -(\mathbf{w}^T\mathbf{y}_j + b) \geq 1\ , && (4.11) \end{aligned}$$

[3] This can be directly observed from (4.8).

where $i = 1, \ldots, N_{\mathbf{x}}$ and $j = 1, \ldots, N_{\mathbf{y}}$.

A special case of the above with $\Sigma = \mathbf{I}$ is precisely the optimization of SVM, where $\mathbf{I}$ is the identity matrix.

Remarks. In the above, two assumptions are implicitly made by SVM: One is the assumption on data "orientation" or data shape, i.e., $\Sigma_{\mathbf{x}} = \Sigma_{\mathbf{y}} = \Sigma$, and the other is the assumption on data "scattering magnitude" or data compactness, i.e., $\Sigma = \mathbf{I}$. However, these two assumptions are inappropriate. We demonstrate this in Figure 4.3 and Figure 4.4. We assume the orientation and the magnitude of each ellipsoid represent the data shape and compactness, respectively, in these figures.

Figure 4.3 plots two types of data with the same data orientations but different data scattering magnitudes. It is obvious that, by ignoring data scattering, SVM is improper to locate itself unbiasedly in the middle of the support vectors (filled points), since $\mathbf{x}$ is more possible to scatter in the horizontal axis. Instead, M^4 is more reasonable (see the solid line in this figure). Furthermore, Figure 4.4 plots the case with the same data scattering magnitudes but different data orientations. Similarly, SVM does not capture the orientation information. In comparison, M^4 grasps this information and demonstrates a more suitable decision plane: M^4 represents the tangent line between two small dashed ellipsoids centered at the support vectors (filled points). Note that SVM and M^4 do not need to achieve the same support vectors. In Figure 4.4, M^4 contains the above two filled points as support vectors, whereas SVM has all the three filled points as support vectors.

Link with Fisher Discriminant Analysis

FDA, an important and popular method, is used widely in constructing decision hyperplanes and reducing the feature dimensionality. In the following discussion, we mainly consider its application as a classifier. FDA involves

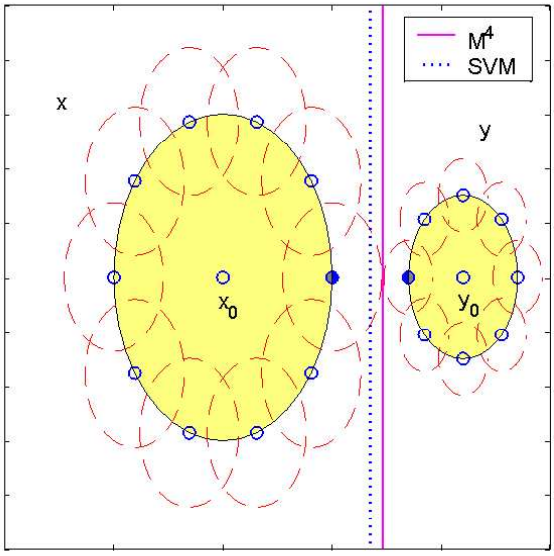


Figure 4.3: An illustration on that SVM omits the data compactness information.

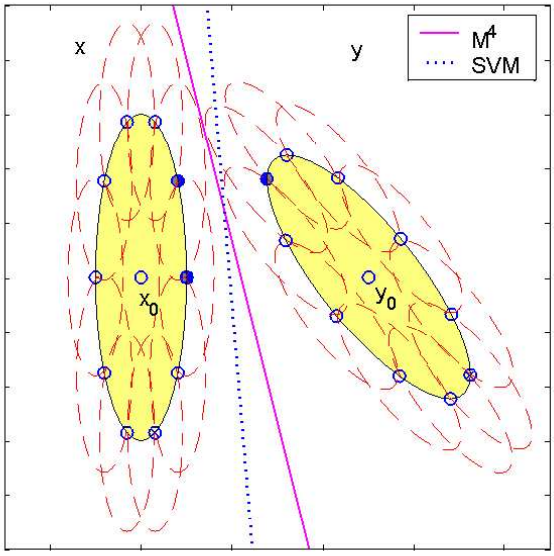


Figure 4.4: An illustration on that SVM discards the data orientation information.

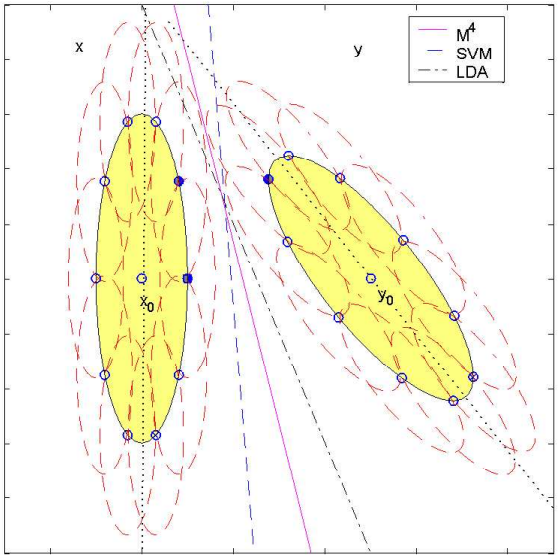


Figure 4.5: An illustration that FDA partly yet incompletely considers the data orientation.

solving the following optimization problem:

$$\max_{\mathbf{w}\neq\mathbf{0}} \frac{|\mathbf{w}^T(\overline{\mathbf{x}}-\overline{\mathbf{y}})|}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}+\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}}.$$

Similar to MPM, FDA also focuses on using the global information rather than considering data both locally and globally. We now show that FDA can be modified to consider data both locally and globally.

If one changes the denominators in (4.2) and (4.3) as $\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}+\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}$, the optimization can be changed as:

$$\max_{\rho,\mathbf{w}\neq\mathbf{0},b} \quad \rho \quad \text{s.t.} \tag{4.12}$$

$$\frac{(\mathbf{w}^T\mathbf{x}_i+b)}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}+\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}} \geq \rho\ , \tag{4.13}$$

$$\frac{-(\mathbf{w}^T\mathbf{y}_j+b)}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}+\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}} \geq \rho\ , \tag{4.14}$$

where $i = 1, \ldots, N_{\mathbf{x}}$ and $j = 1, \ldots, N_{\mathbf{y}}$. The above optimization is actually a generalized case of FDA, which considers data locally and globally. This is verified as follows.

If one performs the procedure similar to that of Section 4.1.2, the above optimization problem is easily verified to be the following optimization:

$$\begin{aligned} &\max_{\rho, \mathbf{w} \neq \mathbf{0}, b} \quad \rho \quad \text{s.t.} \\ &\mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}}) \geq \rho \sqrt{\mathbf{w}^T \Sigma_{\mathbf{x}} \mathbf{w} + \mathbf{w}^T \Sigma_{\mathbf{y}} \mathbf{w}}. \end{aligned} \tag{4.15}$$

One can change (4.15) as: $\rho \leq \frac{|\mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}})|}{\sqrt{\mathbf{w}^T \Sigma_{\mathbf{x}} \mathbf{w} + \mathbf{w}^T \Sigma_{\mathbf{y}} \mathbf{w}}}$, which is exactly the optimization of the FDA ($\mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}})$ is implicitly implied as a positive value from (4.13) and (4.14)).

Remarks. The extended FDA optimization actually focuses on considering the data orientation, while omitting the data scattering magnitude information. Using the analysis similar to that of Section 4.1.2, we can know that the extended FDA lacks the consideration on the data scattering magnitude. Its decision hyperplane in the example of Figure 4.3 coincides with that of SVM. With respect to the data orientation, it actually uses the average of covariances for two types of data. As illustrated in Figure 4.5, the extended FDA corresponds to the line lying exactly in the middle of the long axes of the $\mathbf{x}$ and $\mathbf{y}$ data. This shows that the extended FDA considers the data orientation partially yet incompletely.

4.1.3 Nonseparable Case

In this section, we modify the M^4 model to handle the nonseparable case. We need to introduce slack variables in this case. The optimization of M^4 is

changed as:

$$\max_{\rho,\mathbf{w}\neq\mathbf{0},b,\boldsymbol{\xi}} \quad \rho - C\sum_{k=1}^{N_\mathbf{x}+N_\mathbf{y}} \boldsymbol{\xi}_k \quad \text{s.t.} \tag{4.16}$$

$$(\mathbf{w}^T\mathbf{x}_i + b) \geq \rho\sqrt{\mathbf{w}^T\Sigma_\mathbf{x}\mathbf{w}} - \boldsymbol{\xi}_i \ , \tag{4.17}$$

$$-(\mathbf{w}^T\mathbf{y}_j + b) \geq \rho\sqrt{\mathbf{w}^T\Sigma_\mathbf{y}\mathbf{w}} - \boldsymbol{\xi}_{j+N_\mathbf{x}} \ , \tag{4.18}$$

$$\xi_k \geq 0 \ ,$$

where $i = 1, \ldots, N_\mathbf{x}$, $j = 1, \ldots, N_\mathbf{y}$, and $k = 1, \ldots, N_\mathbf{x} + N_\mathbf{y}$. C is the positive penalty parameter and $\boldsymbol{\xi_k}$ is the slack variable, which can be considered as the extent how the training point $\mathbf{z}_k$ disobeys the ρ margin ($\mathbf{z}_k = \mathbf{x}_k$ when $1 \leq k \leq N_\mathbf{x}$; $\mathbf{z}_k = \mathbf{y}_{k-N_\mathbf{y}}$ when $N_\mathbf{x} + 1 \leq k \leq N_\mathbf{x} + N_\mathbf{y}$). Thus $\sum_{k=1}^{N_\mathbf{x}+N_\mathbf{y}} \boldsymbol{\xi}_k$ can be conceptually regarded as the training error or the empirical error. In other words, the above optimization achieves maximizing the minimum margin while minimizing the total training error.

Solving Method

As clearly observed, when ρ is fixed, the optimization is equivalent to minimizing $\sum_{k=1}^{N_\mathbf{x}+N_\mathbf{y}} \boldsymbol{\xi}_k$ under the same constraints. This is once again an SOCP problem and thus can be solved in polynomial time. We can then update ρ according to some rules and repeat the whole process until an optimal ρ is found. This is once again the so-called line search problem. We still adopt Quadratic Interpolation method to solve this problem, which converges superlinearly to the global optimum if suitable starting points are assigned [11]. Since we have introduced this linear search method in Chapter 3, we simply omit it here.

In summary, we iterate the following two steps to solve the modified optimization.

Step 1. Generate a new ρ_n from three previous ρ_1, ρ_2, ρ_3 by using the Quadratic Interpolation method.

Step 2. Fix $\rho = \rho_n$, perform the optimization based on SOCP algorithms. Update ρ_1, ρ_2, ρ_3.

4.1.4 Further Connection with Minimum Error Minimax Probability Machine

In this section, we show how the M^4 can be connected with Minimum Error Minimax Probability Machine [67], which is a worst-case Bayes optimal classifier and a superset of MPM as well.

If one looks into carefully the optimization of nonseparable M^4, a more precise form is the one replacing $\boldsymbol{\xi_k}$ with $\boldsymbol{\xi_k}\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}$ in (4.17) and $\boldsymbol{\xi_k}\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}$ in (4.18). However, this optimization may prove to be a difficult problem. Nevertheless, we can start from this precise form and derive the connection of M^4 with MEMPM.

We reformulate the optimization of (4.17)-(4.18) as their precise forms as follows:

$$\max_{\rho,\mathbf{w}\neq\mathbf{0},b,\boldsymbol{\xi}} \quad \rho - C\sum_{k=1}^{N_{\mathbf{x}}+N_{\mathbf{y}}}\boldsymbol{\xi}_k \quad \text{s.t.} \tag{4.19}$$

$$\frac{\mathbf{w}^T\mathbf{x}_i + b}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}} \geq \rho - \boldsymbol{\xi}_i \ , \tag{4.20}$$

$$-\frac{\mathbf{w}^T\mathbf{y}_j + b}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}} \geq \rho - \boldsymbol{\xi}_{j+N_{\mathbf{x}}} \ , \tag{4.21}$$

$$\boldsymbol{\xi}_k \geq 0 \ , \tag{4.22}$$

where $i = 1, \ldots, N_{\mathbf{x}}$, $j = 1, \ldots, N_{\mathbf{y}}$, and $k = 1, \ldots, N_{\mathbf{x}} + N_{\mathbf{y}}$.

Maximizing (4.19) contains a similar meaning as minimizing $\frac{1}{\rho^2} + B\sum_{k=1}^{N_{\mathbf{x}}+N_{\mathbf{y}}}\boldsymbol{\xi}_k$ (B is a positive parameter) in a sense that they both attempt to maximize the margin ρ and minimize the error rate. If we consider $\sum_{k=1}^{N_{\mathbf{x}}+N_{\mathbf{y}}}\boldsymbol{\xi}_k$ as the residue and regard $\frac{1}{\rho^2}$ as the regularization term, the optimization can be cast into the

framework of solving ill-posed problems. [4]

According to [153, 155], the above optimization, pointed as the Tikhonov's Variation Method [149], is equivalent to the optimization below refereed to Ivannov's Quasi-Solution Method [69], in the sense that if one of the methods for a given value of the parameter (say C) produces a solution $\{\mathbf{w}, b\}$, then the other method can derive the same solution by adapting its corresponding parameter (say A).

$$\min_{\rho,\mathbf{w}\neq\mathbf{0},b,\boldsymbol{\xi}} \quad \sum_{k=1}^{N_\mathbf{x}+N_\mathbf{y}} \boldsymbol{\xi}_k \quad \text{s.t.} \tag{4.23}$$

$$\frac{\mathbf{w}^T\mathbf{x}_i + b}{\sqrt{\mathbf{w}^T\Sigma_\mathbf{x}\mathbf{w}}} \geq \rho - \boldsymbol{\xi}_i \ , \tag{4.24}$$

$$-\frac{\mathbf{w}^T\mathbf{y}_j + b}{\sqrt{\mathbf{w}^T\Sigma_\mathbf{y}\mathbf{w}}} \geq \rho - \boldsymbol{\xi}_{j+N_\mathbf{x}} \ , \tag{4.25}$$

$$\rho \geq A \ , \boldsymbol{\xi}_k \geq 0 \ , \tag{4.26}$$

where A is a positive constant parameter.

Now if we expand (4.24) for each i and add them all together, we can obtain:

$$N_\mathbf{x}\frac{\mathbf{w}^T\overline{\mathbf{x}} + b}{\sqrt{\mathbf{w}^T\Sigma_\mathbf{x}\mathbf{w}}} \geq N_\mathbf{x}\rho - \sum_{i=1}^{N_\mathbf{x}} \boldsymbol{\xi}_i \ . \tag{4.27}$$

This equation can easily be changed as:

$$\sum_{i=1}^{N_\mathbf{x}} \boldsymbol{\xi}_i \geq N_\mathbf{x}\rho - N_\mathbf{x}\frac{\mathbf{w}^T\overline{\mathbf{x}} + b}{\sqrt{\mathbf{w}^T\Sigma_\mathbf{x}\mathbf{w}}} \ . \tag{4.28}$$

[4] A trick can be made by assuming $\frac{1}{\rho^2}$ as a new variable and thus the condition that the regularization is convex can be satisfied.

Similarly, if we expand (4.25) for each j and add them all together, we obtain:

$$\sum_{j=1}^{N_{\mathbf{y}}} \boldsymbol{\xi}_{j+N_{\mathbf{x}}} \geq N_{\mathbf{y}}\rho + N_{\mathbf{y}} \frac{\mathbf{w}^T\overline{\mathbf{y}} + b}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}} \ . \tag{4.29}$$

By adding (4.28) and (4.29), we obtain:

$$\sum_{k=1}^{N} \boldsymbol{\xi}_k \geq N\rho - (N_{\mathbf{x}} \frac{\mathbf{w}^T\overline{\mathbf{x}} + b}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}} - N_{\mathbf{y}} \frac{\mathbf{w}^T\overline{\mathbf{y}} + b}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}}) \ . \tag{4.30}$$

To achieve minimum training error, namely, $\min_{\rho,\mathbf{w}\neq\mathbf{0},b,\boldsymbol{\xi}} \quad \sum_{k=1}^{N_{\mathbf{x}}+N_{\mathbf{y}}} \boldsymbol{\xi}_k$, we may consider to minimize its lower bound as specified by the right hand side of (4.30). Hence in this case ρ should attain its lower bound A, while the second part should be as large as possible, i.e.,

$$\max_{\mathbf{w}\neq\mathbf{0},b} \theta \frac{\mathbf{w}^T\overline{\mathbf{x}} + b}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}} - (1-\theta) \frac{\mathbf{w}^T\overline{\mathbf{y}} + b}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}} \ , \tag{4.31}$$

where θ is defined as $\frac{N_{\mathbf{x}}}{N}$ and thus $1-\theta$ denotes $\frac{N_{\mathbf{y}}}{N}$. If one further transforms the above to:

$$\max_{\mathbf{w}\neq\mathbf{0},b} \quad \theta t + (1-\theta)s \quad \text{s.t.} \tag{4.32}$$

$$\frac{\mathbf{w}^T\overline{\mathbf{x}} + b}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}} \geq t \ , \tag{4.33}$$

$$-\frac{\mathbf{w}^T\overline{\mathbf{y}} + b}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}} \geq s \ , \tag{4.34}$$

one can see that the above optimizes a very similar form as the MEMPM model except that (4.32) changes to $\min_{\mathbf{w}\neq\mathbf{0},b} \theta\frac{t^2}{1+t^2} + (1-\theta)\frac{s^2}{1+s^2}$ [67]. In MEMPM, $\frac{t^2}{1+t^2}$ ($\frac{s^2}{1+s^2}$) (denoted as α (β)) represents the worst-case accuracy for the classification of future $\mathbf{x}$ ($\mathbf{y}$) data. Thus MEMPM maximizes the weighted accuracy on the future data. In M^4, s and t represent the corresponding margin, which is defined as the distance from the hyperplane to the class center. Therefore,

it represents the weighted maximum margin machine in this sense. Moreover, since the function of $g(u) = \frac{u^2}{1+u^2}$ increases monotonically with u, maximizing the above formulae contains a physical meaning similar to the optimization of MEMPM in some sense.

Remarks. Implicit constraints are contained for the optimization of the above derived special case of M^4. Empirically, (4.27) cannot achieve the equality in the normal case, since (4.24) and (4.25) can only achieve equalities for support vectors. Moreover, the slack variables are usually far smaller than ρ. This implies we can consider $\frac{\mathbf{w}^T\overline{\mathbf{x}}+b}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}} > \rho = A$. Analogously, for $\mathbf{y}$, a similar statement can be obtained. The presence of these two constraints is essential, since with the constraints, the parameter ρ involves in the optimization. Moreover, these two constraints also prevent the circumstance that the decision hyperplane is extremely far away from one class center, while being very close to the other class center.

4.2 Bound on the Error Rate

In this section, we provide theoretical results on the bound of the error rate of $\mathbf{M}^4$. We first borrow the leave-one-out theorem from [95] and [154]:

Lemma 14 The leave-one-out estimator is almost unbiased.

We then present the generation bound of $\mathbf{M}^4$ as the following theorem:

Theorem 15 If (1) the training set containing N samples are separated by the decision hyperplane derived by $\mathbf{M}^4$ and (2) the mean and covariance matrices are reliably estimated, then the expectation of the probability of the test error is bounded by the expectation of the minimum of two values: The ratio m/N and $\theta\frac{1}{1+d_{\mathbf{x}}^2} + (1-\theta)\frac{1}{1+d_{\mathbf{y}}^2}$, where m is the number of support vectors, $d_{\mathbf{x}}$ and $d_{\mathbf{y}}$ are the corresponding Mahalanobis distances from the class center $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$ to

the decision hyperplane, and θ is prior probability of the $\mathbf{x}$ data. Namely,

$$E[P_{error}] \leq E[\min(\frac{m}{N}, \theta\frac{1}{1+d_{\mathbf{x}}^2} + (1-\theta)\frac{1}{1+d_{\mathbf{y}}^2})] \tag{4.35}$$

Proof: According to Lemma 14, to prove $E[P_{error}] \leq E[\frac{m}{N}]$, we only need to show that the number of errors by the leave-one-out method does not exceed the number of support vectors. Actually, this is the case. If we leave a non-support vector out and then we perform training on the remaining data, the decision hyperplane will not change, since the decision hyperplane is just decided by support vectors and the covariance matrices (statistically, one point will not influence the covariance of data). Therefore, this non-support vector will be recognized correctly. Thus the leave-one-out method classifies correctly all the samples that are not support vectors, i.e., the number of the leave-one-out errors does not exceed the number of the support vectors.

We next prove $E[P_{error}] \leq E[\theta\frac{1}{1+d_{\mathbf{x}}^2} + (1-\theta)\frac{1}{1+d_{\mathbf{y}}^2}]$. According to [85, 67, 101], if the means and covariances are reliably estimated, $\frac{d_{\mathbf{x}}^2}{(1+d\mathbf{x}^2)}$ and $\frac{d_{\mathbf{y}}^2}{(1+d\mathbf{y}^2)}$ represent the worst-case rates in recognizing correctly the $\mathbf{x}$ data and $\mathbf{y}$ data respectively. Therefore, $\theta\frac{1}{1+d_{\mathbf{x}}^2} + (1-\theta)\frac{1}{1+d_{\mathbf{y}}^2}$ represents the expected maximum error rate, i.e., $E[P_{error}] \leq E[\theta\frac{1}{1+d_{\mathbf{x}}^2} + (1-\theta)\frac{1}{1+d_{\mathbf{y}}^2}]$. ■

Remarks. Note that the above two items actually represent two meanings of the $\mathbf{M}^4$ model, i.e., minimizing the leave-one-out error presents the contribution by considering the local information from data; on the other hand, the second item describes the effect by considering the global information from data. Moreover, if we further examine the second item, $d_{\mathbf{x}}(d_{\mathbf{y}})$ is actually determined by two parts: the Mahalanobis distance from the support vectors to the corresponding class center $\mathbf{x}$ ($\mathbf{y}$) and the margin ρ. This can be observed in Figure. 4.2. Intuitively, the larger the margin ρ is, the larger $d_{\mathbf{x}}$ and $d_{\mathbf{y}}$ are, which leads to a smaller expected test error in the future. This motivates the margin maximization in the large margin machines.

4.3 Reduction

The variable in previous sections is $[\mathbf{w}, b, \xi_1, \ldots, \xi_{N_\mathbf{x}}, \ldots, \xi_{N_\mathbf{x}+N_\mathbf{y}}]$, whose dimension is $n+1+N_\mathbf{x}+N_\mathbf{y}$. The number of the second order conic constraints is easily verified to be $N_\mathbf{x}+N_\mathbf{y}$. This size of the generated constraint matrix will be a big number and may thus encounter problems in solving large scale classification tasks. Therefore, we should reduce both the number of constraints and the number of variables.

Since this problem is caused by the number of the data points, we consider removing some redundant points to reduce both the space and time complexity. The reduction rule is introduced as follows:

Reduction Rule: Set a threshold $\nu \in [0,\ 1)$. In each class, calculate the Manhalanobis distance, d_i, of each point to its corresponding class center. if $d_i^2/(1+d_i^2)$, denoted as ν_i, is greater than ν, namely, $\nu_i \geq \nu$, keep this point; otherwise, remove this point.

The intuition under this rule is that, in general the more discriminant information the point contains, the further it is from its center (unless it is a noise point). The inner justification under this rule is from [85]: $d^2/(1+d^2)$ is the worst-case classification accuracy for future data, where d is the minimax Manhalanobis distance from the class center to the decision hyperplane. Thus removing those points with small ν's, namely, $d_i^2/(1+d_i^2)$ will not affect the worst-case classification accuracy and will not greatly reduce the overall performance.

Nevertheless, to cancel the negative impact caused by removing those points, we add the following global constraint:

$$\mathbf{w}^T(\overline{\mathbf{x}}-\overline{\mathbf{y}}) \geq \rho(\sqrt{\mathbf{w}^T\Sigma_\mathbf{x}\mathbf{w}}+\sqrt{\mathbf{w}^T\Sigma_\mathbf{y}\mathbf{w}}). \tag{4.36}$$

Integrating the above, we formulate the modified model as follows:

$$\begin{aligned}
&\max_{\rho,\mathbf{w}\neq\mathbf{0},b,\xi}\{\rho - C(\sum_{k=1}^{r_{\mathbf{x}}+r_{\mathbf{y}}}\xi_k + (N_{\mathbf{x}}+N_{\mathbf{y}}-r_{\mathbf{x}}-r_{\mathbf{y}})\xi_m)\} \quad \text{s.t.}\\
&(\mathbf{w}^T\mathbf{x}_i + b) \geq \rho(\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}) - \xi_i,\ i = 1,\ldots,r_{\mathbf{x}}\ ,\\
&-(\mathbf{w}^T\mathbf{y}_j + b) \geq \rho(\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}) - \xi_{j+r_{\mathbf{x}}},\ j = 1,\ldots,r_{\mathbf{y}}\ ,\\
&\mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}}) \geq \rho(\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}} + \sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}) - \xi_m\ ,\\
&\xi_m \geq 0, \quad \xi_k \geq 0, k = 1,\ldots,r_{\mathbf{x}}+r_{\mathbf{y}}\ ,
\end{aligned}$$

where, ξ_m is the slack variable for the global constraint (4.36), ξ_k are modified slack variables for the remaining data points, $r_{\mathbf{x}}$ is the number of the remaining points for $\mathbf{x}$, and $r_{\mathbf{y}}$ is the number of the remaining points for $\mathbf{y}$.

Remarks. An interesting observation from the above is that, when we set the reduction threshold ν to a larger value, or simply to the maximum value 1, the M^4 optimization degrades to the standard MPM optimization. This would imply that the above modified M^4 model contains a worst-case performance of MPM, if the incorporated local information is useful.

4.4 Kernelization

One may note that in the above, the classifier derived from M^4 is provided in a linear configuration. In order to handle nonlinear classification problems, in this section, we seek to use the kernelization trick [138] to map the n-dimensional data points into a high-dimensional feature space $\mathbb{R}^f$, where a linear classifier corresponds to a nonlinear hyperplane in the original space.

The kernel mapping can be formulated as: $\mathbf{x}_i \to \varphi(\mathbf{x}_i)$, $\mathbf{y}_j \to \varphi(\mathbf{y}_j)$, where $i = 1,\ldots,N_{\mathbf{x}}$, $j = 1,\ldots,N_{\mathbf{y}}$, and $\varphi : \mathbb{R}^n \to \mathbb{R}^f$ is a mapping function. The corresponding linear classifier in $\mathbb{R}^f$ is $\boldsymbol{\gamma}^T\varphi(\mathbf{z}) = b$, where $\boldsymbol{\gamma}$, $\varphi(\mathbf{z}) \in \mathbb{R}^f$, and $b \in \mathbb{R}$.

The optimization of M^4 in the feature space can be written as:

$$\max_{\rho,\boldsymbol{\gamma}\neq\mathbf{0},b} \quad \rho \quad \text{s.t.} \tag{4.37}$$

$$\frac{(\boldsymbol{\gamma}^T\varphi(\mathbf{x}_i)+b)}{\sqrt{\boldsymbol{\gamma}^T\Sigma_{\varphi(\mathbf{x})}\boldsymbol{\gamma}}} \geq \rho, \quad i=1,2,\ldots,N_{\mathbf{x}}\ , \tag{4.38}$$

$$\frac{-(\boldsymbol{\gamma}^T\varphi(\mathbf{y}_j)+b)}{\sqrt{\boldsymbol{\gamma}^T\Sigma_{\varphi(\mathbf{y})}\boldsymbol{\gamma}}} \geq \rho, \quad j=1,2,\ldots,N_{\mathbf{y}}\ . \tag{4.39}$$

However, to make the kernel work, we need to represent the optimization and the final decision hyperplane into a kernel form, $\mathbf{K}(\mathbf{z}_1,\mathbf{z}_2) = \varphi(\mathbf{z}_1)^T\varphi(\mathbf{z}_2)$, namely, an inner product form of the mapping data points.

4.4.1 Foundation of Kernelization for M^4

In the following, we demonstrate that the kernelization trick indeed works in M^4, provided suitable estimates of means and covariance matrices are applied therein.

Corollary 16 If the estimates of means and covariance matrices are given in M^4 as the following estimates:

$$\overline{\varphi(\mathbf{x})} = \sum_{i=1}^{N_{\mathbf{x}}}\lambda_i\varphi(\mathbf{x}_i), \quad \overline{\varphi(\mathbf{y})} = \sum_{j=1}^{N_{\mathbf{y}}}\omega_j\varphi(\mathbf{y}_j)\ ,$$

$$\Sigma_{\varphi(\mathbf{x})} = \rho_{\mathbf{x}}\mathbf{I}_n + \sum_{i=1}^{N_{\mathbf{x}}}\Lambda_i(\varphi(\mathbf{x}_i)-\overline{\varphi(\mathbf{x})})(\varphi(\mathbf{x}_i)-\overline{\varphi(\mathbf{x})})^T\ ,$$

$$\Sigma_{\varphi(\mathbf{y})} = \rho_{\mathbf{y}}\mathbf{I}_n + \sum_{j=1}^{N_{\mathbf{y}}}\Omega_j(\varphi(\mathbf{y}_j)-\overline{\varphi(\mathbf{y})})(\varphi(\mathbf{y}_j)-\overline{\varphi(\mathbf{y})})^T\ ,$$

where $\mathbf{I}_n$ is the identity matrix of dimension n then the optimal $\boldsymbol{\gamma}$ in (4.37-4.39) lies in the space spanned by the training points.

Proof: We write $\boldsymbol{\gamma} = \boldsymbol{\gamma}_p + \boldsymbol{\gamma}_d$, where $\boldsymbol{\gamma}_p$ is the projection of $\boldsymbol{\gamma}$ in the vector space spanned by all the training data points and $\boldsymbol{\gamma}_d$ is the orthogonal

component to this span space. By using $\boldsymbol{\gamma}_d^T\varphi(\mathbf{x}_i) = 0$ and $\boldsymbol{\gamma}_d^T\varphi(\mathbf{y}_j) = 0$, one can easily verify that the optimization (4.37-4.39) changes to:

$$\begin{aligned}
&\max_{\rho,\{\boldsymbol{\gamma}_p,\boldsymbol{\gamma}_d\}\neq\mathbf{0},b} \quad \rho \quad \text{s.t.} \\
&\frac{-(\boldsymbol{\gamma}_p^T\varphi(\mathbf{x}_i)+b)}{\sqrt{\boldsymbol{\gamma}_p^T\sum_{i=1}^{N_\mathbf{x}}\Lambda_i(\varphi(\mathbf{x}_j)-\overline{\varphi(\mathbf{x})})(\varphi(\mathbf{x}_i)-\overline{\varphi(\mathbf{x})})^T\boldsymbol{\gamma}_p+\rho_\mathbf{x}(\boldsymbol{\gamma}_p^T\boldsymbol{\gamma}_p+\boldsymbol{\gamma}_d^T\boldsymbol{\gamma}_d)}} \geq \rho\ , \\
&\frac{-(\boldsymbol{\gamma}_p^T\varphi(\mathbf{y}_j)+b)}{\sqrt{\boldsymbol{\gamma}_p^T\sum_{j=1}^{N_\mathbf{y}}\Omega_j(\varphi(\mathbf{y}_j)-\overline{\varphi(\mathbf{y})})(\varphi(\mathbf{y}_j)-\overline{\varphi(\mathbf{y})})^T\boldsymbol{\gamma}_p+\rho_\mathbf{y}(\boldsymbol{\gamma}_p^T\boldsymbol{\gamma}_p+\boldsymbol{\gamma}_d^T\boldsymbol{\gamma}_d)}} \geq \rho\ ,
\end{aligned}$$

where $i = 1, \ldots, N_\mathbf{x}$, $j = 1, \ldots, N_\mathbf{y}$. Since we intend to maximize the margin ρ, the denominators in the above two constraints need to be as small as possible. This would lead to $\boldsymbol{\gamma}_d = \mathbf{0}$. In other words, the optimal $\boldsymbol{\gamma}$ lies in the vector space spanned by all the training data points. Note that the above discussion is assumed in the feature space. ■

According to Corollary 16, if we use the plug-in estimates to approximate the means and covariance matrices, we can write $\boldsymbol{\gamma}$ as a linear combination form of training data points:

$$\boldsymbol{\gamma} = \sum_{i=1}^{N_\mathbf{x}}\mu_i\varphi(\mathbf{x}_i) + \sum_{j=1}^{N_\mathbf{y}}\upsilon_j\varphi(\mathbf{y}_j)\ , \tag{4.40}$$

where the coefficients μ_i, $\upsilon_j \in \mathbb{R}, i = 1, \ldots, N_\mathbf{x}$, $j = 1, \ldots, N_\mathbf{y}$.

4.4.2 The Kernelization Result

We present the kernelization result as the following theorem.

Theorem 17 [**Kernelization Theorem of** M^4] The optimal decision hyperplane for M^4 involves solving the following optimization problem:

$$\begin{aligned}
&\max_{\rho,\boldsymbol{\eta}\neq\mathbf{0},b} \quad \rho \quad \text{s.t.} \\
&\frac{(\boldsymbol{\eta}^T\mathbf{K}_i + b)}{\sqrt{\frac{1}{N_\mathbf{x}}\boldsymbol{\eta}^T\tilde{\mathbf{K}}_\mathbf{x}^T\tilde{\mathbf{K}}_\mathbf{x}\boldsymbol{\eta}}} \geq \rho, \quad i = 1, 2, \ldots, N_\mathbf{x}\ , \\
&\frac{-(\boldsymbol{\eta}^T\mathbf{K}_{j+N_\mathbf{x}} + b)}{\sqrt{\frac{1}{N_\mathbf{y}}\boldsymbol{\eta}^T\tilde{\mathbf{K}}_\mathbf{y}^T\tilde{\mathbf{K}}_\mathbf{y}\boldsymbol{\eta}}} \geq \rho, \quad j = 1, 2, \ldots, N_\mathbf{y}\ .
\end{aligned}$$

Proof: The theorem can easily be proved by simply substituting the plug-in estimations of means and covariances matrices and (4.40) into (4.37)-(4.39). ■

The optimal decision hyperplane can be represented as a linear form in the kernel space

$$f(\mathbf{z}) = \sum_{i=1}^{N_\mathbf{x}} \boldsymbol{\eta}_{*i}\mathbf{K}(\mathbf{z}, \mathbf{x}_i) + \sum_{i=1}^{N_\mathbf{y}} \boldsymbol{\eta}_{*N_\mathbf{x}+i}\mathbf{K}(\mathbf{z}, \mathbf{y}_i) + b_*\ ,$$

where $\boldsymbol{\eta}_*$ and b_* are the optimal parameters obtained by the optimization procedure. The notations in the above are defined similar to Chapter 3. However, for an easy reference, we also summarize them in Table 4.1.

4.5 Experiments

In this section, we present the evaluation results of M^4 in comparison with SVM and MPM on both synthetic toy data sets and real world benchmark data sets. SOCP problems are solved based on the general software named Sedumi [144, 145]. The covariance matrices are given by the plug-in estimates.

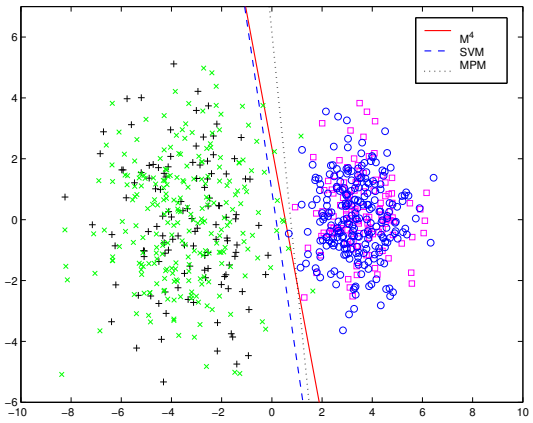
Notation		
$\mathbf{z} \in \mathbb{R}^{N_\mathbf{x}+N_\mathbf{y}}$	$\mathbf{z}_i :=$	$\mathbf{x}_i \quad i = 1, 2, \ldots, N_\mathbf{x}$.
	$\mathbf{z}_i :=$	$\mathbf{y}_{i-N_\mathbf{x}} \quad i = N_\mathbf{x}+1, N_\mathbf{x}+2, \ldots, N_\mathbf{x}+N_\mathbf{y}$.
$\boldsymbol{\eta} \in \mathbb{R}^{N_\mathbf{x}+N_\mathbf{y}}$	$\boldsymbol{\eta} :=$	$[\mu_1, \ldots, \mu_{N_\mathbf{x}}, \upsilon_1, \ldots, \upsilon_{N_\mathbf{y}}]^T$.
$\mathbf{K}$ is Gram matrix	$\mathbf{K}_{i,j} :=$	$\varphi(\mathbf{z}_i)^T \varphi(\mathbf{z}_j)$.
	$\mathbf{K}_\mathbf{x} :=$	$\begin{pmatrix} \mathbf{K}_{1,1} & \mathbf{K}_{1,2} & \cdots & \mathbf{K}_{1,N_\mathbf{x}+N_\mathbf{y}} \\ \mathbf{K}_{2,1} & \mathbf{K}_{2,2} & \cdots & \mathbf{K}_{2,N_\mathbf{x}+N_\mathbf{y}} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{K}_{N_\mathbf{x},1} & \mathbf{K}_{N_\mathbf{x},2} & \cdots & \mathbf{K}_{N_\mathbf{x},N_\mathbf{x}+N_\mathbf{y}} \end{pmatrix}$.
	$\mathbf{K}_\mathbf{y} :=$	$\begin{pmatrix} \mathbf{K}_{N_\mathbf{x}+1,1} & \mathbf{K}_{N_\mathbf{x}+1,2} & \cdots & \mathbf{K}_{N_\mathbf{x}+1,N_\mathbf{x}+N_\mathbf{y}} \\ \mathbf{K}_{N_\mathbf{x}+2,1} & \mathbf{K}_{N_\mathbf{x}+2,2} & \cdots & \mathbf{K}_{N_\mathbf{x}+2,N_\mathbf{x}+N_\mathbf{y}} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{K}_{N_\mathbf{x}+N_\mathbf{y},1} & \mathbf{K}_{N_\mathbf{x}+N_\mathbf{y},2} & \cdots & \mathbf{K}_{N_\mathbf{x}+N_\mathbf{y},N_\mathbf{x}+N_\mathbf{y}} \end{pmatrix}$.
$\tilde{\mathbf{k}}_\mathbf{x}, \tilde{\mathbf{k}}_\mathbf{y} \in \mathbb{R}^{N_\mathbf{x}+N_\mathbf{y}}$	$[\tilde{\mathbf{k}}_\mathbf{x}]_i :=$	$\frac{1}{N_\mathbf{x}} \sum_{j=1}^{N_\mathbf{x}} \mathbf{K}(\mathbf{x}_j, \mathbf{z}_i)$.
	$[\tilde{\mathbf{k}}_\mathbf{y}]_i :=$	$\frac{1}{N_\mathbf{y}} \sum_{j=1}^{N_\mathbf{y}} \mathbf{K}(\mathbf{y}_j, \mathbf{z}_i)$.
$\mathbf{1}_{N_\mathbf{x}} \in \mathbb{R}^{N_\mathbf{x}}$	$\mathbf{1}_i :=$	$1 \quad i = 1, 2, \ldots N_\mathbf{x}$.
$\mathbf{1}_{N_\mathbf{y}} \in \mathbb{R}^{N_\mathbf{y}}$	$\mathbf{1}_i :=$	$1 \quad i = 1, 2, \ldots N_\mathbf{y}$.
$\tilde{\mathbf{K}} :=$	$\begin{pmatrix} \tilde{\mathbf{K}}_\mathbf{x} \\ \tilde{\mathbf{K}}_\mathbf{y} \end{pmatrix} :=$	$\begin{pmatrix} \mathbf{K}_\mathbf{x} - \mathbf{1}_{N_\mathbf{x}} \tilde{\mathbf{k}}_\mathbf{x}^T \\ \mathbf{K}_\mathbf{y} - \mathbf{1}_{N_\mathbf{y}} \tilde{\mathbf{k}}_\mathbf{y}^T \end{pmatrix}$.

Table 4.1: Notations used in Kernelization

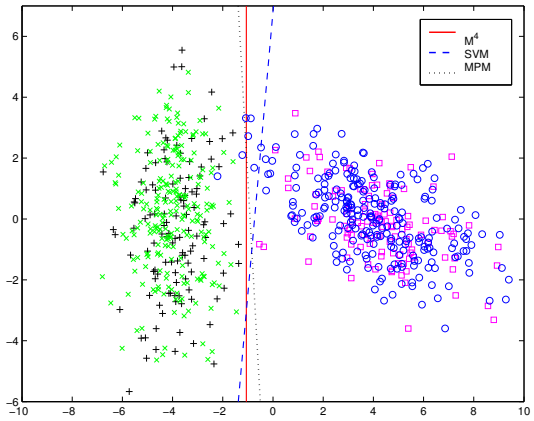
4.5.1 Evaluations on Three Synthetic Toy Data Sets

We demonstrate the advantages of our approach in comparison with SVM and MPM in the following synthetic toy data sets first.

As illustrated in Figure 4.6, we generate two types of data with the same data orientations but with different data magnitudes in Figure 4.6 (a), while we generate two types of data with the same data magnitudes but with different data orientations in Figure 4.6 (b). In (a), the $\mathbf{x}$ data are randomly sampled from the Gaussian distribution with the mean as $[-3.5,\ 0]^T$ and the covariance as $[3,\ 0; 0,\ 4.5]$, while the $\mathbf{y}$ data are randomly sampled from another Gaussian distribution with the mean and the covariance as $[3.5,\ 0]^T$ and $[1,\ 0; 0,\ 1.5]$ respectively. In (b), the $\mathbf{x}$ data are randomly sampled from the Gaussian distribution with the mean as $[-4,\ 0]^T$ and the covariance as $[1,\ 0; 0,\ 5]$, while the $\mathbf{y}$ data are randomly sampled from another distribution with the mean and the covariance as $[4,\ 0]^T$ and $[1,\ 0; 0,\ 5]$ respectively. Moreover, to generate different data orientation, in this figure, the $\mathbf{y}$ data are rotated anti-clockwise at the angle of $-\frac{7}{8}\pi$. In both (a) and (b), training (test) data, consisting of 120 (250) data points for each class, are presented as o's (+'s) and $\times$'s



(a)



(b)

Figure 4.6: The first two synthetic toy examples to illustrate M^4. Training (test) data, consisting of 120 (250) data points for each class are presented as o's (+'s) and $\times$'s ($\square$'s) for $\mathbf{x}$ and $\mathbf{y}$ respectively. Subfigure (a) demonstrates SVM omits the data compactness information and (b) demonstrates SVM discards the data orientation information, while M^4 achieves the ideal decision boundary, which considers data both locally and globally.

(□'s) for $\mathbf{x}$ and $\mathbf{y}$ respectively. Observed from Figure 4.6, M^4 demonstrates its advantages over SVM. More specifically, in Figure 4.6 (a), SVM discards the information of the data magnitudes, whose decision hyperplane lies basically in the middle of boundary points of two types of data, while M^4 successfully utilizes this information, i.e., its decision hyperplane lies closer to the compact class ($\mathbf{y}$ data), which is more reasonable. Similarly, in Figure 4.6 (b), M^4 takes advantage of the information of the data orientation, while SVM simply overlooks this information, which results in a lot of points incorrectly classified.

When MPM is compared with M^4, since in the above two data sets, the global information, i.e., the mean and the covariance can be reliably estimated from data, they achieve similar performance. To see the difference between M^4 and MPM, we generate another data set as illustrated in Figure 4.7, where we intentionally generate a very small number of training data, i.e., only 20 training points. Similarly, the data are generated under two Gaussian distributions: the $\mathbf{x}$ data are randomly sampled from the Gaussian distribution with the mean as $[-3,\ 0]^T$ and the covariance as $[0.5,\ 0; 0,\ 8]$, while the $\mathbf{y}$ data are randomly sampled from another distribution with the mean and the covariance as $[4,\ 0]^T$ and $[6,\ 0; 0,\ 1]$ respectively. Training data and test data are represented using similar symbols to Figure 4.6. From Figure 4.7, once again M^4 achieves the ideal decision boundary, which considers data both locally and globally; whereas SVM obtains the local boundary just in the middle of the support vectors, which discards the global information, namely the statistical "trend" of data occurrence. For MPM, its decision hyperplane is exclusively dependent on the mean and covariance matrices. Thus we can see that this hyperplane coincides with the data shape, i.e., the long axis of training data of $\mathbf{x}$ is nearly in the same direction as the MPM decision hyperplane. However, the estimated mean and covariance is inaccurate due to the small number of data points. This results in a relatively lower test accuracy as illustrated in Figure 4.7(b). In comparison, M^4 incorporates the information of the local

points to neutralize the effect caused by inaccurate estimations. The test accuracies for the above three toy data sets listed in Table 4.2 also demonstrates the advantages of M^4.

Dataset	M^4	SVM	MPM
I(%)	**98.8**	96.8	**98.8**
II(%)	**98.8**	97.2	**98.8**
III(%)	**98.3**	97.5	95.8

Table 4.2: Comparisons of classification accuracies between M^4, SVM, and MPM on the toy data sets.

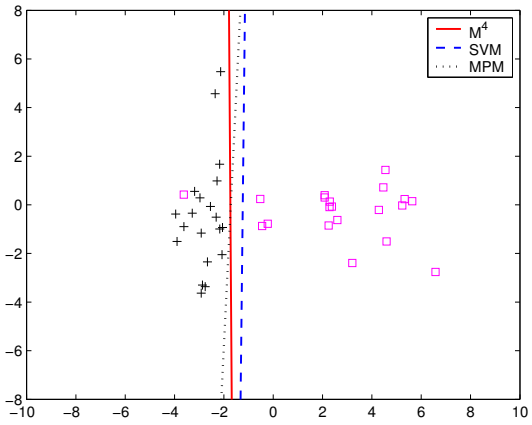
4.5.2 Evaluations on Benchmark Data Sets

We perform evaluations on seven standard data sets. Data for Twonorm problem were synthetically generated according to [16]. The remaining six data sets were real world data obtained from the UCI machine learning repository [12]. We compared M^4 with SVM and MPM engaging both the linear and Gaussian kernels. The parameter C for both M^4 and SVM was tuned via cross validations [77], so were the width parameter in the Gaussian kernel for all three models. The final performance results were obtained via the $10-$fold cross validation. Table 4.3 summarizes the evaluation results.

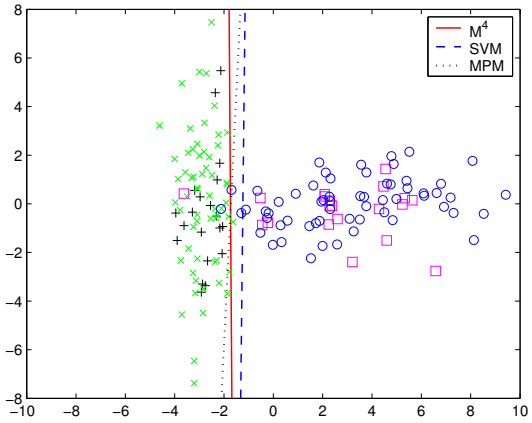
Data set	Linear kernel			Gaussian kernel		
	M^4	SVM	MPM	M^4	SVM	MPM
Twonorm(%)	96.5 ± 0.6	95.1 ± 0.7	$\mathbf{97.6 \pm 0.5}$	96.5 ± 0.7	96.1 ± 0.4	$\mathbf{97.6 \pm 0.5}$
Breast(%)	$\mathbf{97.5 \pm 0.7}$	96.6 ± 0.5	96.9 ± 0.8	$\mathbf{97.5 \pm 0.6}$	96.7 ± 0.4	96.9 ± 0.8
Ionosphere(%)	$\mathbf{87.7 \pm 0.8}$	86.9 ± 0.6	84.8 ± 0.8	$\mathbf{94.5 \pm 0.4}$	94.2 ± 0.3	92.3 ± 0.6
Pima(%)	77.7 ± 0.9	$\mathbf{77.9 \pm 0.7}$	76.1 ± 1.2	77.6 ± 0.8	$\mathbf{78.0 \pm 0.5}$	76.2 ± 1.2
Sonar(%)	$\mathbf{77.6 \pm 1.2}$	76.2 ± 1.1	75.5 ± 1.1	84.9 ± 1.2	86.5 ± 1.1	$\mathbf{87.3 \pm 0.8}$
Vote(%)	$\mathbf{96.1 \pm 0.5}$	95.1 ± 0.4	94.8 ± 0.4	$\mathbf{96.2 \pm 0.5}$	95.9 ± 0.6	94.6 ± 0.4
Heart-disease(%)	$\mathbf{86.6 \pm 0.8}$	84.1 ± 0.7	83.2 ± 0.8	$\mathbf{86.2 \pm 0.8}$	83.8 ± 0.5	83.1 ± 1.0

Table 4.3: Comparisons of classification accuracies among M^4, SVM, and MPM.

From the results, we observe that M^4 achieves the best overall performance. In comparison with SVM and MPM, M^4 wins five cases in the linear kernel and four in the Gaussian kernel. The evaluations on these standard benchmark data sets demonstrate that it is worth considering data both locally and



(a)

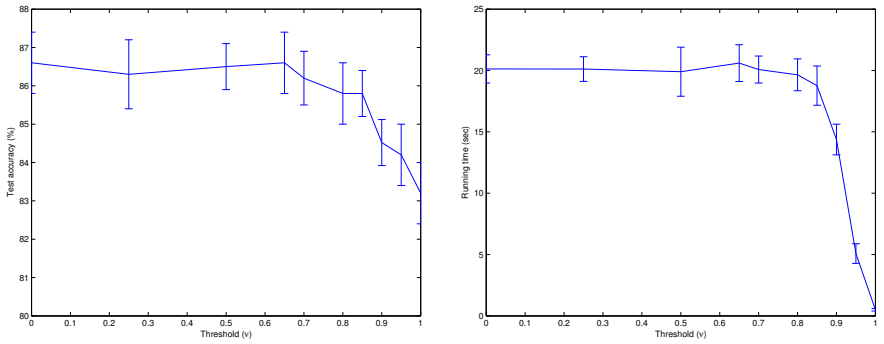


(b)

Figure 4.7: The third synthetic toy example to illustrate M^4. Training (test) data, consisting of 20 (60) data points for each class are presented as o's (+'s) and ×'s (□'s) for $\mathbf{x}$ and $\mathbf{y}$ respectively. Subfigure (a) demonstrates the decision boundaries derived from training data, while (b) illustrates the performance of these hyerplanes on the test set. The M^4 achieves the ideal decision boundary, which considers data both locally and globally.

globally, which is emphasized in M^4. Inspecting the differences between M^4 with SVM, the kernelized M^4 appears marginally better than the kernelized SVM, while the linear M^4 demonstrates a distinctive advantage over the linear SVM. This phenomenon may be explained on two hands. On one hand, this can be explained from the fact that the data points are very sparse in the kernelized space or feature space (compared with the huge dimensionality in the Gaussian kernel). Thus the plug-in estimates of the covariance matrices may not accurately represent the data information in this case. On the other hand, it is well-known that the kernelization will not keep the structure information in the feature space. One direct consequence is that maximizing the margin in the feature space does not necessarily maximize the margin in the original space [151]. Therefore, without building some connections between the original space and the feature space, utilizing the structure information, e.g., covariance matrices in the feature space seem not to do much help in this sense. Inspecting these two points, one interesting topic in the future is to consider forcing constraints on the mapping function so as to maintain the data topology in the kernelization process.

In the above, we do not perform the reduction on these data sets. To illustrate how the reduction algorithm works for decreasing the computation time while maintaining the test accuracy, we implement it on the Heart-disease data set. We perform the reduction in training sets and then keep test sets unchanged. We repeat this process for different thresholds ν. We then plot the curve of the cross validation accuracy against the threshold ν. Moreover, we also plot the curve of the computation time against the threshold. This can be seen in Figure 4.8. From this figure, we can see both the computation time and the test accuracy change insensitive against ν when ν is set to some small values, e.g. $\nu \leq 0.7$. If looking into the Heart-disease data set, we find that most data points are faraway from their corresponding class center in terms of the Manhalanobis distance. Thus setting small values to ν does not actually



(a) (b)

Figure 4.8: Reduction on the Heart-disease data set

reduce many data points. This generates both a relatively flat changing curve in the test accuracy and the computation time in this range. As ν is changing larger, the computation time decreases fast as more and more data points are removed, while the test accuracy goes down slowly. When the threshold is set to 1, the M^4 degrades to the MPM model, yielding the test accuracy of M^4 achieves the same value of MPM. This demonstrates how the proposed reduction algorithms can decrease the computation time while maintaining good performance. When used in practice, the threshold can be set according to the required response time.

4.6 Discussions and Future Work

We will discuss several important issues in this section. First, although M^4 can be solved in polynomial time, the large computation time is still one of its biggest limitations. This may cause problems especially in its kernelization version. Note that the proposed reduction algorithm in this chapter does not completely solve this problem, since removing points will inevitably lose information. In this sense, it is crucial to develop some special algorithms for M^4. Due to the sparsity of M^4 (it also contains support vectors), it is

therefore very interesting to investigate whether decomposable methods or an analogy to the Sequential Minimal Optimization [119] designed for SVM can also be applied in training M^4. We believe there is much to gain from such explorations. Certainly, this is a high worthy research direction in the future.

Second, although we have derived an error bound for M^4, digging out the direct connection or perform empirical comparison of this bound with those of its special cases, namely, SVM and MPM maintains an interesting problem. Especially, it is an open problem whether there exists an unified form of the bounds for M^4, SVM, and MPM. This interesting subject deserves future deep explorations.

Third, since in this chapter we mainly discuss M^4 for two-category classifications, how to extend its application to multi-way classifications is also an important topic in the future.

4.7 Summary

Local learning approaches, e.g., large margin machines have demonstrated their advantages in machine learning and pattern recognition. However, they derive the decision boundary only in a local way. For example, the most popular large margin classifier, Support Vector Machine obtains the decision hyperplane by focusing on considering some critical local points called support vectors, while discarding all other points; on the other hand, global learning models e.g., Minimax Probability Machine obtain the classifier only based on global information, i.e., the mean and covariance information in MPM, while ignoring all individual local points. Differently, our proposed model is constructed based on both a local and a global view of data. This new model is theoretically important in the sense that SVM and MPM can both be considered as its special case. Furthermore, the optimization of M^4 can be cast as a sequential Conic Programming problem, which can be solved in polynomial

time.

We have provided a clear geometrical interpretation, and established detailed connections among our model and other models such as Support Vector Machine, Minimax Probability Machine, Fisher Discriminant Analysis, and Minimum Error Minimax Probability Machine. We have also shown to exploit Mercer kernels to extend our model to build up nonlinear decision boundaries. In addition, we have also proposed a reduction method to decrease the computation time. Experimental results on both synthetic data sets and real world benchmark data sets have demonstrated the advantages of M^4 over Support Vector Machine and Minimax Probability Machine.

Chapter 5

Extension I: BMPM for Imbalanced Learning

We consider the problem of the binary classification on imbalanced data, in which nearly all the instances are labelled as one class, while far fewer instances are labelled as the other class, usually the more important class. Traditional machine learning methods seeking an accurate performance over a full range of instances are not suitable to deal with this problem, since they tend to classify all the data into the majority class, usually the less important class. Moreover, many current methods have tried to utilize some intermediate factors, e.g., the distribution of the training set, the decision thresholds or the cost matrix, to impose a bias towards the important class. However, it remains uncertain whether these roundabout methods can improve the performance in a systematic way. In this chapter, we apply Biased Minimax Probability Machine, one of the special case of Minimum Error Minimax Probability Machine to deal with the imbalanced learning tasks. Different from previous methods, this model achieves in a worst-case scenario, to derive the biased classifier by directly controlling the classification accuracy on each class. More precisely, BMPM builds up an explicit connection between the classification accuracy and the bias, which thus provides a rigorous treatment on imbalanced data. We examine different models and compare BMPM with three other competitive

methods, i.e., the Naive Bayesian classifier, the k-Nearest Neighbor method, and the decision tree method C4.5. The experimental results demonstrates the superiority of this model.

This chapter is organized as follows. In the next section, we briefly present an introduction to the imbalanced learning. We then reiterate in a tight version the theoretical foundation of this chapter, namely the BMPM model. Following that, we then in Section 5.3 apply the BMPM model to deal with the imbalanced learning tasks. Following that, we evaluate the BMPM model based on a series of experiments. In Section 5.5, we make discussions and present future work. Finally, we summarize this chapter in Section 5.6.

5.1 Introductions to Imbalanced Learning

Learning classifiers from imbalanced or skewed data sets is an important topic, arising very often in practice in classification problems. In such problems, almost all the instances are labelled as one class, while far fewer instances are labelled as the other class, usually the more important class. It is obvious that traditional classifiers seeking an accurate performance over a full range of instances are not suitable to deal with imbalanced learning tasks, since they tend to classify all the data into the majority class, which is usually the less important class.

To cope with imbalanced data sets, there are types of methods, such as the methods of sampling [21, 92, 81], the methods of moving the decision thresholds [99, 122], and the methods of adjusting the cost matrix [18, 99]. The first school of methods aims to reduce the data imbalance by "down-sampling" (removing) instances from the majority class or "up-sampling" (duplicating) the training instances from the minority class or both. The second school of methods tries to adapt the decision threshold to impose a bias on the minority class. Similarly, the third school of methods improves the prediction performance by

adjusting the weight (cost) for each class.

A common problem for all the three families of methods is that they lack a rigorous and systematic treatment on imbalanced data. For the sampling method, either up- or down-sampling is unsuitable: up-sampling will introduce noise, while down-sampling the data will lose information. Moreover, to incorporate a good bias, it is usually difficult to know what a proportion should be sampled. For these reasons, Provost states it as an open problem whether simply varying the skewness of the data distribution can improve prediction performance systematically [122]. For the method of adjusting the cost matrix or adapting weights, similar problems are also encountered, i.e., they are hard to build direct connections between the cost matrix or the weights and the biased classification quantitatively. To impose a suitable bias towards the important class, they have to adapt these factors by trials. Therefore, these methods cannot rigorously handle imbalanced data.

In this chapter, we apply Biased Minimax Probability Machine (BMPM) to handle the tasks of learning from imbalanced data. Different from the sampling methods, BMPM does not remove or duplicate data. When compared with the methods of changing the thresholds or weights, our model builds up an explicit connection between the classification accuracy and the bias. It thus offers an elegant way to incorporate the bias into classification by directly controlling the real accuracy.

5.2 Biased Minimax Probability Machine

Suppose two random n-dimensional vectors $\mathbf{x}$ and $\mathbf{y}$ represent two classes of data, where $\mathbf{x}$ belongs to the family of distributions with a given mean $\bar{\mathbf{x}}$ and a covariance $\Sigma_{\mathbf{x}}$, denoted as $\mathbf{x} \sim (\bar{\mathbf{x}}, \Sigma_{\mathbf{x}})$; similarly, $\mathbf{y}$ belongs to the family of distributions with a given mean $\bar{\mathbf{y}}$ and a covariance $\Sigma_{\mathbf{y}}$, denoted as $\mathbf{y} \sim (\bar{\mathbf{y}}, \Sigma_{\mathbf{y}})$. Here $\mathbf{x}$, $\mathbf{y}$, $\bar{\mathbf{x}}$, $\bar{\mathbf{y}} \in \mathbb{R}^n$, and $\Sigma_{\mathbf{x}}$, $\Sigma_{\mathbf{y}} \in \mathbb{R}^{n \times n}$. In this chapter,

the class $\mathbf{x}$ also represents the important or minority class and the class $\mathbf{y}$ represents the corresponding less important or majority class.

The Biased Minimax Probability Machine can be described as follows[1]:

$$\max_{\alpha,\beta,b,\mathbf{w}\neq\mathbf{0}} \quad \alpha \quad \text{s.t.} \qquad \inf_{\mathbf{x}\sim(\bar{\mathbf{x}},\Sigma_{\mathbf{x}})} \Pr\{\mathbf{w}^T\mathbf{x} \geq b\} \geq \alpha \,, \tag{5.1}$$

$$\inf_{\mathbf{y}\sim(\bar{\mathbf{y}},\Sigma_{\mathbf{y}})} \Pr\{\mathbf{w}^T\mathbf{y} \leq b\} \geq \beta \,, \tag{5.2}$$

$$\beta \geq \beta_0 \,. \tag{5.3}$$

Here α means the lower bound of the probability (accuracy) for the classification of future cases of the class $\mathbf{x}$ with respect all distributions with the mean and covariance as $(\mathbf{x},\ \Sigma_{\mathbf{x}})$; in other words, α is the worst-case accuracy for the class $\mathbf{x}$. Similarly, β is the lower bound of the accuracy of the class $\mathbf{y}$. This optimization achieves to maximize the accuracy (the probability α) for the biased class $\mathbf{x}$ while simultaneously maintaining the class $\mathbf{y}$'s accuracy at an acceptable level β_0 by setting a lower bound as (5.3). In comparison, the Minimax Probability Machine (MPM) in [84, 85] considers the balanced data set; therefore, it makes α equal to β.

This optimization setting seems to be more useful in incorporating a bias into classifications for imbalanced learning problems. A typical example can be seen in the epidemic disease diagnosis problem, which is usually an imbalanced classification problem as well. The "ill" cases are usually much fewer than the healthy cases. However, misclassification of the "ill" class results in more serious consequence than misclassification of the "healthy" case. Thus an unequal treatment on different classes is obviously necessary.

We summarize the advantages of our biased model in the following. First,

[1]Note that, for easy explanations, the model description is in the slightly different but essentially the same form as the one introduced in Chapter 3.

this method provides a different treatment on different classes, i.e, the hyperplane $\mathbf{w}^{*T}\mathbf{z} = b^*$ given by the solution of this optimization favors the classification of the important class $\mathbf{x}$ over the less important class $\mathbf{y}$. Second, given reliable mean and covariance matrices, the derived decision hyperplane is directly associated with two real accuracy indicators, i.e., α and β, for each class. Thus, by varying the lower bound of β, i.e., β_0 and deriving the corresponding classifier, we can quantitatively incorporate a bias into the classification. Third, this model contains a distribution-free feature. With no distribution assumption on data, the derived hyperplane seems to be more general and valid than a large family of classifiers, namely the generative classifiers [62, 132] including the Naive Bayesian classifier [88], which has to make specific distribution assumptions. Fourth, as shown shortly in Section 5.3, either we can simply modify this BMPM optimization to automatically search the best β_0 in terms of some standard criteria, or slightly different from the current setting, we can quantitatively generate the trade-off curve between the accuracies on different classes and leave the task of choosing the best β_0 to the users. Finally, although the BMPM contains the above advantages, it does not trade them for efficiency. It is shortly shown that the optimization of BMPM can be cast as a Fractional Programming (FP) problem and thus can be solved efficiently. In short, with these important features, BMPM appears to offer a more direct and rigorous scheme to handle biased classification tasks, especially the imbalanced classifications, where the importance or cost for each class is unequal.

5.3 Learning from Imbalanced Data by Using BMPM

In this section, we apply the novel BMPM model into the tasks of learning from imbalanced data. We first review four standard imbalanced learning criteria. We then, based on two of them, apply BMPM into imbalanced learning tasks.

5.3.1 Four Criteria to Evaluate Learning from Imbalanced Data

In general, four criteria are used to evaluate the imbalanced learning. They are (1) the criterion of Minimum Cost (MC), (2) the criterion of Maximum Geometry Mean (MGM) of the accuracies on the majority class and the minority class, (3) the criterion of the Maximum Sum (MS) of the accuracies on the majority class and the minority class, and (4) the criterion of Receiver Operating Characteristic (ROC) analysis. We review these criteria as follows.

Aiming to solve the problems caused by maximizing the accuracy over a full range of data, instead, Grzymala-Busse [52] et al. maximized the sum of the accuracies on the minority class and the majority class (or maximized the difference between the true positive and false positive accuracy). This criterion is also widely used in other fields, e.g., graph detection, especially line detection and arc detection, where it is called Vector Recovery Index [35, 93]. Similarly, Kubat et al. [80] proposed to use the geometric mean instead of the sum of the accuracies. However, compared to maximizing the sum, this criterion has a nonlinear form, which is not easy to be automatically optimized. On the other hand, when the cost of misclassification is known, a minimum cost measure defined as (5.4) should be used [14]:

$$Cost = F_p \cdot C_{F_p} + F_n \cdot C_{F_n} \ , \tag{5.4}$$

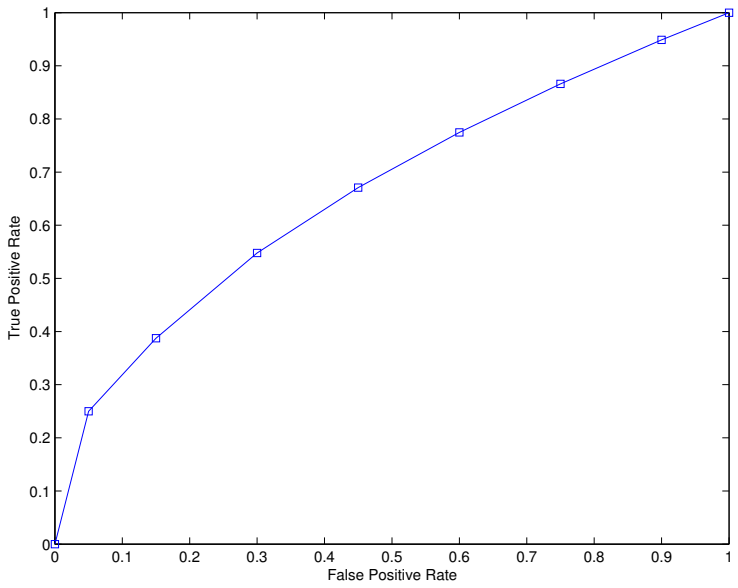


Figure 5.1: An artificially generated Receiver Operating Characteristic (ROC) Curve

where F_p is the number of the false positive, C_{F_p} is the cost of a false positive, F_n is the number of the false negative, and C_{F_n} is the cost of a false negative. However, because the cost of misclassification is generally unknown in real cases, the usage of this measure is somewhat restricted. Considering this point, some researchers introduced the ROC analysis [98, 99, 147]. This criterion plots a so-called ROC curve to visualize the tradeoff between the false positive rate and the true positive rate and leaves the task of the selection of a specific tradeoff to the practitioners. Fig. 5.1 illustrates an artificially generated ROC curve. It has been suggested that the area beneath an ROC curve can be used as a measure of accuracy in many applications [123, 146]. Thus, a good classifier for imbalanced learning should have a larger area.

Based on the above review, in this chapter, we will focus on using the criterion of MS and the ROC curve analysis to evaluate the classifiers.

5.3.2 BMPM for Maximizing the Sum of the Accuracies

In the following, we first modify the formulation of BMPM to maximize the sum of the accuracies for two classes. Next, we make an analysis on the solvability of the modification version. Finally we present the optimization method.

Model Modification

When using BMPM for the criterion of MS, we can modify the formulation of BMPM as follows:

$$\max_{\alpha,\beta,b,\mathbf{w}\neq\mathbf{0}} \quad \alpha+\beta \quad \text{s.t.} \tag{5.5}$$

$$\inf_{\mathbf{x}\sim\{\overline{\mathbf{x}},\Sigma_{\mathbf{x}}\}} \Pr\{\mathbf{w}^T\mathbf{x}\geq b\}\geq\alpha\ , \tag{5.6}$$

$$\inf_{\mathbf{y}\sim\{\overline{\mathbf{y}},\Sigma_{\mathbf{y}}\}} \Pr\{\mathbf{w}^T\mathbf{y}\leq b\}\geq\beta\ . \tag{5.7}$$

The above formulation directly maximizes the sum of the lower bounds of the accuracies so as to maximize the sum of the accuracies. In comparison, to achieve the maximum sum of the accuracies, some other approaches, e.g., the methods of sampling or the methods of adapting the weights have to search the best sampling proportion or the best weights by trials, which are in general very time-consuming. Since the above optimization is in fact nearly the same as the Minimum Error Probability Machine, it can be similarly solved by the Sequential Biased Minimax Probability Machine optimization method as introduced in Chapter 3. We thus do not elaborate it here.

5.3.3 BMPM for ROC Analysis

It is straightforward to apply the BMPM model to plot the ROC curve, since the lower bounds α and β directly and quantitatively control the accuracies for two classes. We only need to adapt the acceptable level for β, namely β_0,

from 0 to 1, to obtain a sequence of trade-offs between the accuracy of the important class and the negative class. We address again, since β_0 represents the lower bound of the accuracy of the less important class, varying β_0 provides a direct and quantitative way to move the decision plane with different trade-offs. Directly associating accuracies with the moving of the hyperplane while assuming no distribution is one of advantages of BMPM over the other methods by adapting the weights or thresholds.

5.4 Experimental Results

In this section, we first illustrate the BMPM model with a toy example. We then evaluate the performance of BMPM on two real world imbalanced data sets, namely the recidivism data set and the rooftop data set in comparison with the Naive Bayesian (NB) classifier, the k-Nearest Neighbor (k-NN) method [1], and the decision tree classifier C4.5 [125].

5.4.1 A Toy Example

We present a toy example to illustrate the BMPM model in this section. Suppose 15 data points of the class $\mathbf{x}$ are generated from a 2-dimensional Gaussian distribution with the mean and covariance matrix as $\overline{\mathbf{x}} = [0\ 1.5]^T$ and $\Sigma_{\mathbf{x}} = [0.5\ 0; 0\ 0.5]$ and 65 data points of the class $\mathbf{y}$ from another 2-dimensional Gaussian distribution with $\overline{\mathbf{y}} = [0\ 0]^T$ and $\Sigma_{\mathbf{y}} = [0.5\ 0; 0\ 0.5]$.

By adapting the lower bound accuracy β_0 for the class $\mathbf{y}$, with optimizing the corresponding BMPM, we obtain a series of decision boundaries for the toy example when using the Gaussian kernel $e^{-\|\mathbf{x}-\mathbf{y}\|^2/\sigma}$ with the parameter σ as 5. These boundaries are illustrated in Fig. 5.2. Green regions are classified as the class $\mathbf{x}$ represented by black +'s, whereas those outside green regions are judged as the class $\mathbf{y}$ plotted as magenta $\square$'s. It is clear to observe that the lower bound β_0 directly controls the accuracy of the class $\mathbf{y}$. More specifically, when

β_0 is set to small values such as 10.00%, 60.00% and 95.00%, the boundary is biased towards the class $\mathbf{x}$. When β_0 is set to larger values such as 99.00%, the classification is biased towards the class $\mathbf{y}$. Moreover we demonstrate in Table 5.1 that the lower bounds β_0 and α can serve as the accuracy indictors. It is observed that these lower bounds keep well, i.e., the corresponding accuracies are slightly higher than the lower bounders except in the case when $\beta_0 = 0.95$. The exception, i.e., that the value of α, 99.16% is greater than the real accuracy 93.33%, is understandable due to the relatively smaller number of training samples: one single misclassification will influence the classification results significantly. This toy example demonstrates that, by changing β_0, BMPM provides an elegant and direct way to incorporate the bias into the classification.

β_0(%)	True Negative Rate(%)	α(%)	True Positive Rate(%)
10.00	13.85	100.00	100.00
60.00	63.08	100.00	100.00
95.00	95.38	99.16	93.33
99.00	100.00	81.94	86.67

Table 5.1: Lower bounds of accuracies, α, β_0 and the real accuracies

5.4.2 Evaluations on Real World Imbalanced Datasets

In this section, we evaluate our novel BMPM model in comparison with three competitive classification methods, namely the Naive Bayesian classifier, the k-Nearest Neighbor methods, and the decision tree, C4.5, on two real world imbalanced data sets, the recidivism data set and the rooftop data set. Before we go into the experimental details, we first introduce these three techniques and adapt them to learn from imbalanced data sets according to previous research results [90, 99].

Modifying Three Learning Techniques

We investigate and modify three learning techniques, the Naive Bayesian classifier, the k-Nearest Neighbor method, and the decision tree, C4.5 in the following.

The Naive Bayesian classifier [63, 88] is proposed based on a very simple assumption, i.e., each attribute is conditionally independent of each other when given the class variable. The decision in a two-category prediction task is made according to the calculation of the posterior probability $p(C|\mathbf{z})$, where C is the class variable and $\mathbf{z}$ represents the observation. When $p(C_1|\mathbf{z}) \geq 0.5$ or another equivalent yet more convenient rule is satisfied, i.e., $p(C_1)p(\mathbf{z}|C_1) \geq p(C_2)p(\mathbf{z}|C_2)$, $\mathbf{z}$ is classified into C_1; otherwise, it is judged as C_2. Even with the strong conditionally independency assumption, the Naive Bayesian classifier demonstrates a surprisingly good performance when compared with state-of-art classifiers [46, 89] such as Support Vector Machines [154] and C4.5 in many domains. By simply introducing a parameter τ into the decision rule $p(C_1)p(\mathbf{z}|C_1) \geq \tau p(C_2)p(\mathbf{z}|C_2)$, Naive Bayesian classifiers can be adapted into the imbalanced learning. For example, specifying $\tau < 1$ imposes a bias towards the C_1 class, whereas specifying $\tau > 1$ imposes a bias towards the C_2 class.

In the k-Nearest Neighbor classification [1], based on some distance measure, e.g, the Euclidean distance measure, k data points, which are the closest to the query point, are selected out. It then labels the query point as the most frequent class among the chosen k points. Although this method is very simple and may suffer from difficulties in high dimensions, it achieves satisfactory performance in many real domains. Following [99], we alter the distance measure δ_j for the class C_j to handle imbalanced learning tasks according to (5.8)

$$\delta_j = d_E(\mathbf{z}, \mathbf{z_j}) - \tau_j d_E(\mathbf{z}, \mathbf{z_j}) \ , \tag{5.8}$$

where $\mathbf{z_j}$ is the closest point from class C_j to the query point, and $d_E(\mathbf{z}, \mathbf{z_j})$ represents the Euclidean distance measure. Similar to the Naive Bayesian classifier, by modifying τ_j, the Nearest Neighbor method can build biased classifiers.

C4.5 is a kind of algorithm introduced by Quinlan for inducing classification models, also called decision trees, from data [125]. By selecting the attributes according to the gain ratios criterion, an information measure of homogeneity, C4.5 builds up a decision tree, where each path from the root to a leaf represents a specific classification rule. We adapt C4.5 to learn from imbalanced data set based on the similar method to [99], i.e., by changing the prior probability to bias the classification.

Evaluations on the Recidivism Dataset

The recidivism data set was obtained from a cohort of releasees of the North Carolina prison system during the time period from July 1, 1977 to June 30, 1978. There are totally $4,618$ individuals in this data set, including a training set with $1,540$ individuals and a test set with $3,078$ individuals. In the training set, 570 (27.5%) individuals were recidivists and 970 (72.5%) were not. In the test set, $1,151$ individuals were recidivists and $1,927$ were not. Although this data set is not skewed as severely as other reported data sets, for example, the fog-data set [113] and the rooftop data set used in the next subsection, it is enough to use this data set to evaluate the performance of the imbalanced learning [99].

We use the same processing method [135] to select and scale nine attributes appeared in Table 5.2, while six other attributes were dropped based on an insignificant test at the 5% level.

We compare the performance of our proposed Biased Minimax Probability Machine model, in both the linear (BMPML) and the Gaussian kernel setting (BMPMG), with the Naive Bayesian classifier, C4.5 and the k-Nearest

Attribute	Description
TSERVED	Time served (in months)
AGE	Age (in months) at the time of release
PRIORS	Number of previous incarcerations
WHITE	Is the individual Caucasian?
FELON	Was the sentence for a felony?
LCHY	Does individual's record indicate a serious problem with alcohol?
JUNKY	Does individual's record indicate a serious problem with hard drugs?
PROPTY	Was individual's sentence for a crime against property?
MALE	Is the individual male?

Table 5.2: Attribute description in the recidivism data set

Neighbor method. These methods are modified into the imbalanced learning according to the methods introduced in the previous section. We run k-NN methods for $k = 1, 3, 5, \ldots, 21$, but we only present the best three results for brevity. The width parameter for the Gaussian kernel is tuned via cross validation methods [77].

We first present the experimental results based on the MS criterion in Table 5.3. To be more comparable, we show the average of the accuracy for each class when each classifier attains the point of the maximum sum. The BMPML achieves an average accuracy of 0.6391 and the BMPMG achieves an average accuracy of 0.6490, while the highest average accuracy among other classifiers is given as 0.6272 by NB. Therefore, in this data set, BMPML and BMPMG outperform other methods in terms of the MS criterion.

Let us next present the experimental results based on the ROC analysis. By setting the thresholds or costs by trials for NB, k-NN, and C4.5, the ROC curves are generated with good shapes as evenly distributed along their length as possible. As discussed in [99], although this generation method may increase the running time for some methods, e.g., k-NN, it works well in C4.5 and NB and is sufficient to evaluate the performance of imbalanced learning. For the BMPM model, since the lower bound β_0 serves as the accuracy indicators, we simply vary it from 0 to 1 to generate the corresponding ROC curve. The ROC curves are shown in Fig. 5.3(a). As seen in this figure, the performances of BMPML and BMPMG are once again superior to those of other methods, since

their ROC curves cover those of other models in most parts. To quantitatively demonstrate the difference, in Table 5.4, we also show the areas beneath the ROC curves approximated by using the trapezoid rule. The BMPML and BMPMG show a consistent superiority to NB, the best of the other three methods.

In addition, in real applications, not all the portions of the ROC curve are of great interest [102]. Usually, those with a small false positive rate and a high true positive rate should be more of interest and importance [159]. We thus especially show the portion of the ROC curve in the range when the false positive rate FP$\in$ [0, 0.5] and the true positive rate TP$\in$ [0.5, 1]. As shown in Fig. 5.3(b), in this range, the superiority of the BMPL and BMPMG is more obvious than the whole ROC curve analysis. This again demonstrates our model's advantages over other methods.

Method	True Negative	True Positive	(True Positive +True Negative) / 2
NB	0.6177	0.6377	0.6272
k-NN(9)	0.6255	0.5464	0.5860
k-NN(11)	0.6238	0.5542	0.5890
k-NN(13)	0.5569	0.6201	0.5885
C4.5	0.7405	0.4900	0.6153
BMPML	0.7037	0.5745	**0.6391**
BMPMG	0.7203	0.5778	**0.6490**

Table 5.3: Performance on a recidivism prediction task based on the MS criterion

Method	Area under ROC Curve
NB	0.6646
k-NN(11)	0.6155
k-NN(13)	0.6189
k-NN(17)	0.6148
C4.5	0.6383
BMPML	**0.6842**
BMPMG	**0.6798**

Table 5.4: Performance on a recidivism prediction task based on the area of ROC curve

Evaluations on the Rooftop Dataset

The rooftop data set consists of $17,829$ overhead images of Fort Hood, Texas, collected as part of the RADIUS project [44], which are of a military base. Depending on whether they are buildings (with a detected rooftop) or not, 781 images in this data set are labelled as positive examples while $17,048$ images are labelled as negative examples. It is clearly observed that this is a severely skewed data set. According to [44, 99], these images were taken from two different viewpoints, i.e., a nadir aspect and an oblique aspect and covered three different areas. Following [91, 99], we represent each of these images in nine continues attributes which are extracted based on various image analysis. The detailed information about this data set is summarized in Table 5.5 and Table 5.6.

Sub-data set	Location	Image Size	Aspect	#Positive	#Negative
1	A	2055×375	Nadir	71	2645
2	A	1803×429	Oblique	74	3349
3	B	670×645	Nadir	197	982
4	B	704×568	Oblique	238	1955
5	C	1322×642	Nadir	87	3722
6	C	1534×705	Oblique	114	4395

Table 5.5: Description of images in the rooftop data set

Attribute	Description
1	Evaluation of the edge support
2	Evaluation of the corner support
3	Evaluation of the parallel support
4	Evaluation of the OTV (Orthogonal Trihedral Vertex) support
5	Evaluation of the shadow corner support
6	Evaluation of gap overlap
7	Evaluation of displacement of edge support
8	Evaluation of crossing lines on any side of the hypothesis
9	Evaluation of existence of T-junction or L-junction on any side

Table 5.6: Description of the attributes in the rooftop data set

We randomly split the rooftop data into a training set with 60% data and a test set with 40% data. We then construct classifiers from imbalanced data based on the training data set and perform evaluations on the test data set. We repeat this procedure ten times and use the average of the results as the

performance metric. In such a setup, we compare our BMPM with other three approaches, i.e., NB, C4.5, and k-NN. Similar to the case in the recidivism data set, NB, C4.5, and k-NN are modified to handle imbalanced data. The width parameter σ is chosen by cross validation methods again. Moreover, we still run k-NN with $k = 1, 3, 5, ..., 21$ and present the best three for brevity.

The results are summarized in Table 5.7 based on the MS criterion, and Fig. 5.4 and Table 5.8 based on the ROC analysis. As is clearly observed, for both criteria, the BMPM method demonstrates its superiority to the other methods, since they have higher sums of the accuracies and larger areas under the ROC curves. Similar to what we do in the recivisim data set, we also plot the more critical portion of the ROC curve in Fig. 5.4(b). The predominance of the BMPML and the BMPMG is even more obvious. To evaluate the performance more reliably, we perform a significance test based on both LabMRMC [34, 97] and a t-test. The analysis shows that the accuracies of BMPML and BMPMG are significantly different from those of other methods at $p \leq 0.05$, both in terms of the MS criterion and the ROC curve criterion.

Method	True Negative	True Positive	(True Positive + True Negative) / 2
BMPML	0.8015 ± 0.0058	0.8231 ± 0.0063	$\mathbf{0.8123 \pm 0.0060}$
BMPMG	0.7997 ± 0.0087	0.8405 ± 0.0100	$\mathbf{0.8201 \pm 0.0091}$
k-NN(7)	0.7510 ± 0.0055	0.8069 ± 0.0062	0.7789 ± 0.0052
k-NN(13)	0.7409 ± 0.0051	0.8140 ± 0.0083	0.7774 ± 0.0061
k-NN(15)	0.7433 ± 0.0067	0.8211 ± 0.0072	0.7822 ± 0.0072
NB	0.7969 ± 0.0043	0.8177 ± 0.0080	0.8073 ± 0.0066
C4.5	0.8176 ± 0.0040	0.7942 ± 0.0063	0.8059 ± 0.0051

Table 5.7: Performance on the rooftop data set based on the MS Criterion

Method	Area under ROC Curve
BMPML	$\mathbf{0.8791 \pm 0.0061}$
BMPMG	$\mathbf{0.8819 \pm 0.0087}$
k-NN(9)	0.8601 ± 0.0091
k-NN(11)	0.8569 ± 0.0058
kNN(15)	0.8582 ± 0.0063
NB	0.8678 ± 0.0060
C4.5	0.8744 ± 0.0062

Table 5.8: Performance on the rooftop Dataset based on the Area of ROC Curve

5.4.3 Evaluations on Disease Data Sets

Diagnosing diseases contains a very similar characteristic to the imbalanced learning, since one class, usually the disease class needs to be given more bias than the other class. Therefore, the above discussed model modifications will be automatically applicable for this kind of tasks. In the following, we evaluate the performance of BMPM on two disease data sets, namely, the Breast-cancer data and the Heart-disease data set, which are obtained from UCI machine learning repository. In the context of diagnosing diseases, the true positive rate is usually called sensitivity, while the true negative rate is called specificity. Therefore, we should maximize the sensitivity while maintaining the specificity acceptable. In the following, we present the experimental results still compared with the modified the Naive Bayesian classifier, the best three k-NN, and C4.5. We randomly split the data for each data set into a training set with 80% data and a test set with 20% data. We then construct classifiers based on the training data set and perform evaluations on the test data set. We repeat this procedure ten times and use the average of the results as the performance metric.

We present the results based on the MS criterion in Table 5.9 for the breast-cancer data set and Table 5.10 for the heart disease data set. Obsereved from these two table, the BMPM model also demonstrates a superiority to other three models. In addition, the T-test also shows that the accuracies of BMPML and BMPMG are significantly different from those of other three classifiers at $p \leq 0.05$.

We next present the experimental results based on the ROC analysis in Fig. 5.5(a), and Fig. 5.6(a). It is observed that the BMPML and BMPMG perform better than other classifiers for both datasets, since in most parts, the BMPM curves dominate those of other methods. More specifically, we calculate the areas under the ROC curves as illustrated in Table 5.11, based

Method	Specificity	Sensitivity	(Specificity+Sensitivity)/2
BMPML	0.9684 ± 0.0029	0.9872 ± 0.0015	**0.9778 ± 0.0021**
BMPMG	0.9612 ± 0.0018	0.9915 ± 0.0011	**0.9764 ± 0.0016**
k-NN(11)	0.9900 ± 0.0047	0.9620 ± 0.0034	0.9760 ± 0.0029
k-NN(17)	0.9862 ± 0.0081	0.9664 ± 0.0058	0.9762 ± 0.0050
k-NN(7)	0.9721 ± 0.0071	0.9752 ± 0.0049	0.9737 ± 0.0058
NB	0.9366 ± 0.0059	0.9719 ± 0.0049	0.9543 ± 0.0051
C4.5	0.9378 ± 0.0074	0.9582 ± 0.0067	0.9480 ± 0.0072

Table 5.9: Comparison of the model performance based on the MS criterion on the breast-cancer data set

Method	Specificity	Sensitivity	(Specificity+Sensitivity)/2
BMPML	0.8549 ± 0.0042	0.8158 ± 0.0013	**0.8354 ± 0.0035**
BMPMG	0.8403 ± 0.0053	0.8572 ± 0.0017	**0.8488 ± 0.0026**
k-NN(17)	0.7654 ± 0.0029	0.8837 ± 0.0018	0.8246 ± 0.0027
k-NN(7)	0.7754 ± 0.0038	0.8844 ± 0.0042	0.8299 ± 0.0037
k-NN(15)	0.7512 ± 0.0028	0.8653 ± 0.0037	0.8082 ± 0.0036
NB	0.7862 ± 0.0052	0.8024 ± 0.0031	0.7943 ± 0.0040
C4.5	0.8831 ± 0.0022	0.7065 ± 0.0018	0.7948 ± 0.0021

Table 5.10: Comparison of the model performance based on the MS criterion on the heart disease data set

on the trapezoid rule. For the breast-cancer dataset, it produces a curve with an area of 0.9953 in the linear setting and a curve with an area of 0.9963 in the Gaussian kernel, whereas the k-NN with $k = 11$ forms a curve with a smaller area equal to 0.9908, the best result of the k-NN, the NB, and C4.5. For the heart disease dataset, the BMPM shows a curve with an area of 0.8814 in the linear setting and a curve with an area of 0.8932 in the Gaussian kernel setting. These two areas are both greater than those of the other methods, i.e., the k-NN classifier, NB and C4.5. In summary, the evaluations based on the area of the ROC curve quantitatively demonstrate the superiority of our BMPM model for both datasets.

In addition, as illustrated in Fig. 5.5(b) and Fig. 5.6(b), we show the critical portion of Fig. 5.5(a) and Fig. 5.6(a) respectively when the false positive rate is in the range of 0.0 to 0.5 and the true positive rate is in the range of 0.5 to 1.0. In this critical region, most parts of the ROC curves of the BMPM cover the corresponding curves of other models in both datasets, which again demonstrates the superiority of the BMPM model.

breast-cancer		heart	
Method	Area under ROC Curve	Method	Area under ROC Curve
BMPML	**0.9953 ± 0.0018**	BMPML	**0.8814± 0.0056**
BMPMG	**0.9963 ± 0.0016**	BMPMG	**0.8932± 0.0043**
k-NN(11)	0.9908 ± 0.0060	k-NN(17)	0.8701 ± 0.0038
k-NN(17)	0.9902 ± 0.0100	k-NN(7)	0.8689± 0.0050
k-NN(7)	0.9887 ± 0.0080	k-NN(15)	0.8596 ± 0.0038
NB	0.9841 ± 0.0060	NB	0.8162 ± 0.0034
C4.5	0.9762 ± 0.0120	C4.5	0.8301± 0.0038

Table 5.11: Comparison of the model performance based on the ROC analysis

5.5 When the Cost for Each Class Is Known

There exists cases in which the cost for each class can be given by experts. In the following, we show that the BMPM model can naturally be adapted into this type of tasks.

Assuming $\mathbf{x}$ and $\mathbf{y}$ are the minority class and the majority class respectively, it is easily verified that minimizing the optimization function given by (5.4) is equivalent to maximizing the following formulation:

$$\max \quad r_{\mathbf{x}} K_{\mathbf{x}} + r_{\mathbf{y}} K_{\mathbf{y}},$$

where $r_{\mathbf{x}}$ is the true positive rate or the accuracy of the class $\mathbf{x}$, $r_{\mathbf{y}}$ is the true negative rate or the accuracy of the class $\mathbf{y}$, $K_{\mathbf{x}}$ and $K_{\mathbf{y}}$ are two constants, which are equal to $C_{F_p} N_{\mathbf{y}}$ and $C_{F_n} N_{\mathbf{x}}$ respectively ($N_{\mathbf{x}}$, $N_{\mathbf{y}}$ are respectively the number of data points labelled as the class $\mathbf{x}$ and $\mathbf{y}$). Similar to the optimization procedure of MS, we can naturally modify the BMPM model into the following formulation:

$$\max_{\alpha,\beta,b,\mathbf{w}\neq\mathbf{0}} \quad K_{\mathbf{x}}\alpha + K_{\mathbf{y}}\beta \quad \text{s.t.} \quad \begin{array}{l} \inf\limits_{\mathbf{x}\sim\{\overline{\mathbf{x}},\Sigma_{\mathbf{x}}\}} \Pr\{\mathbf{w}^T\mathbf{x} \geq b\} \geq \alpha \ , \\ \inf\limits_{\mathbf{y}\sim\{\overline{\mathbf{y}},\Sigma_{\mathbf{y}}\}} \Pr\{\mathbf{w}^T\mathbf{y} \leq b\} \geq \beta \ . \end{array}$$

The above optimization derives the classification boundary by maximizing the weighted lower bound of the real accuracies or the weighted worst-case real

accuracies so as to minimize the overall classification risk. Moreover, similar to the MS case, it is easily validated that this optimization problem can be cast a sequential BMPM problem. Hence, it can similarly be solved based on the method presented in Chapter 3.

5.6 Summary

In this chapter, we have applied a novel model named Biased Minimax Probability Machine to deal with the task of learning from imbalanced data sets. Given reliable estimation of the mean and covariance of data, this model constructs the classification boundary by directly controlling the lower bound of the real accuracy and thus provides a systematic and rigorous treatment on skewed data. We have evaluated the BMPM model on two real world imbalanced data sets and two disease data sets in terms of two criteria. In both criteria, the performances are shown to be the best when compared with other competitive methods such as the Naive Bayesian classifier, the k-Nearest Neighbor method, and the decision tree classifier, C4.5.

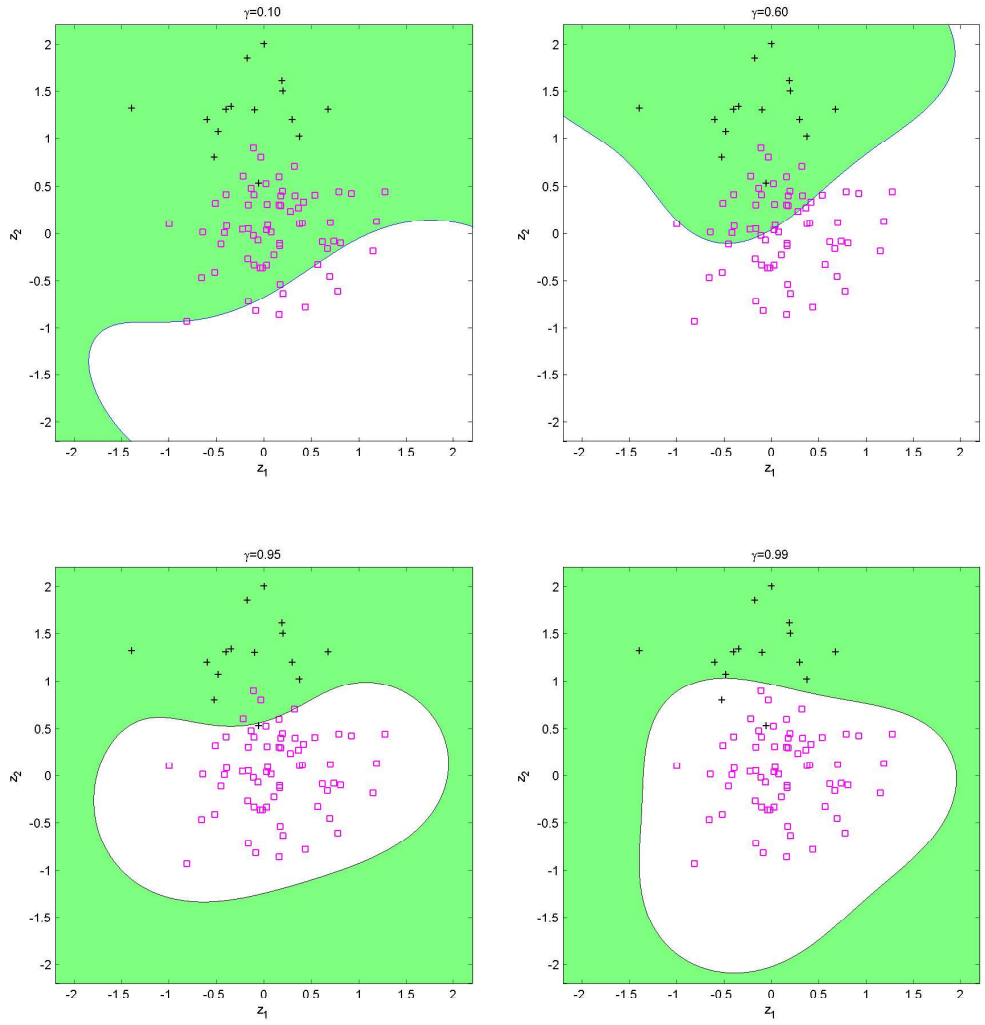
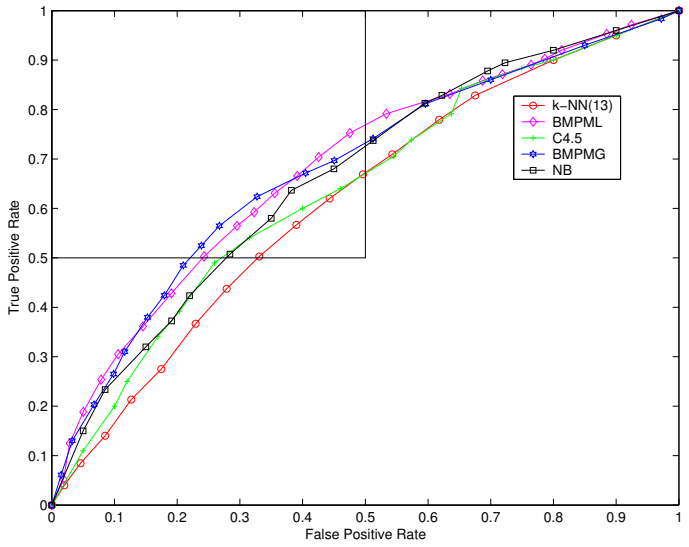
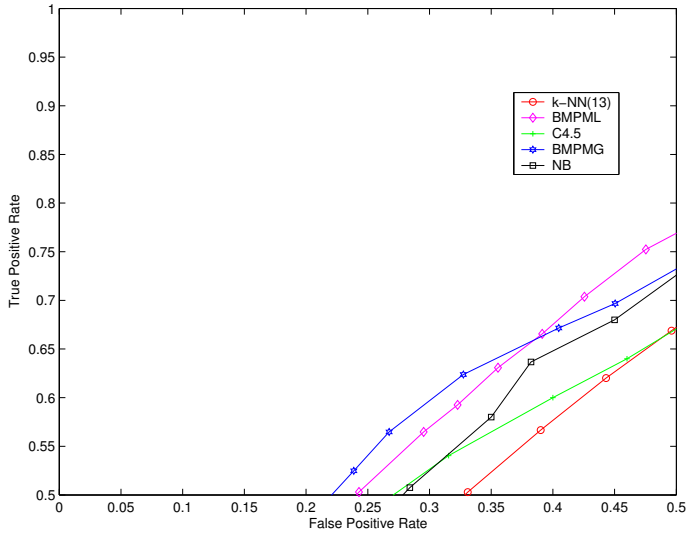


Figure 5.2: A toy example to illustrate BMPM. Data of the class $\mathbf{x}$ is plotted as black +'s, and data of class $\mathbf{y}$ as magenta □'s. The green area represents the classification region of the class $\mathbf{x}$, while the area outside the green region is classified as the class $\mathbf{y}$.

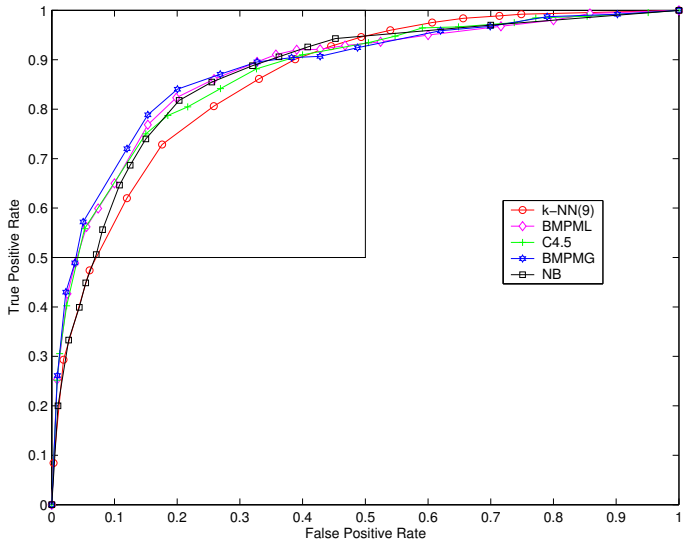


(a)

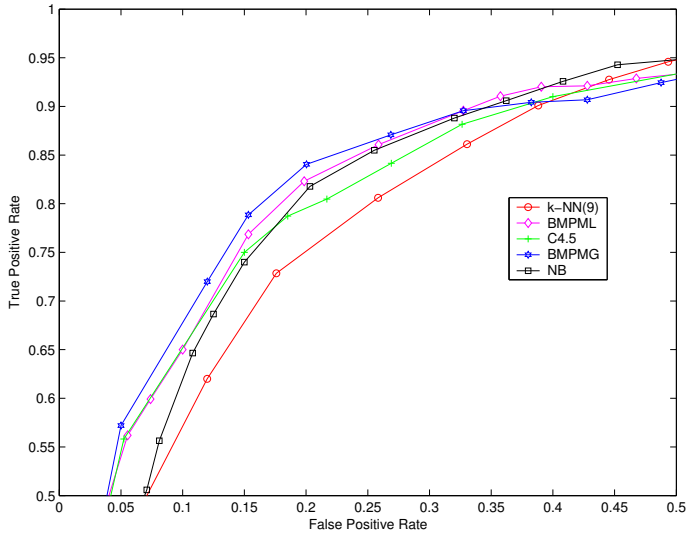


(b)

Figure 5.3: ROC curves for the recidivism data set. Subfigure (a) shows a full range of the ROC curve, while (b) shows a critical proportion of the ROC curve, which is of more interest in real applications. Both figures demonstrate the superiority of the BMPM model, since the curves of BMPML and BMPMG cover those of other models in most parts and thus have a larger area.

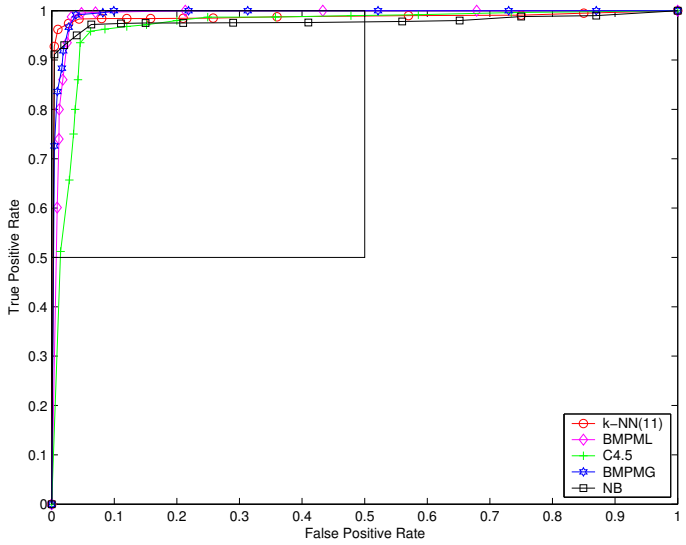


(a)

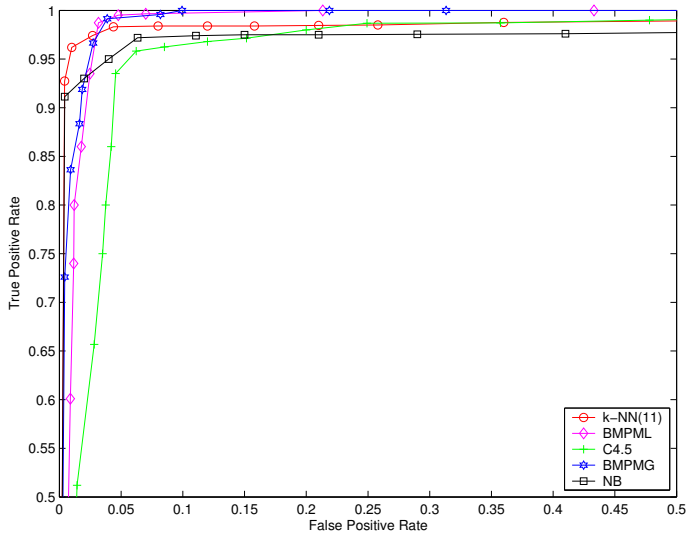


(b)

Figure 5.4: ROC curves for the rooftop data set. We ran each method by randomly partitioning the data set into a training data set (60%) and a test data set (40%). The evaluations were iterated 10 times. We then average the true positive rate and false positive rate to generate the ROC curves. Subfigure (a) shows a full range of the ROC curve, while (b) shows a critical proportion of the ROC curve, which is of more interest in real applications. Both figures demonstrate the superiority of the BMPML and BMPMG model to other models, since the curves of BMPML and BMPMG cover those of other models in most parts and thus have a larger area.

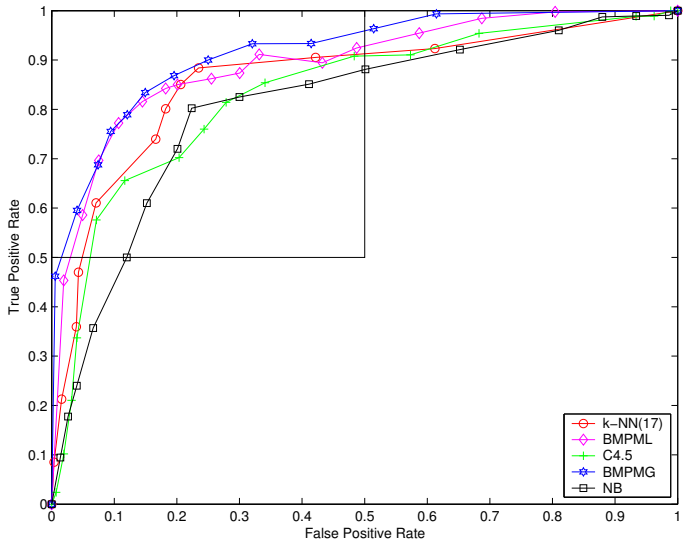


(a)

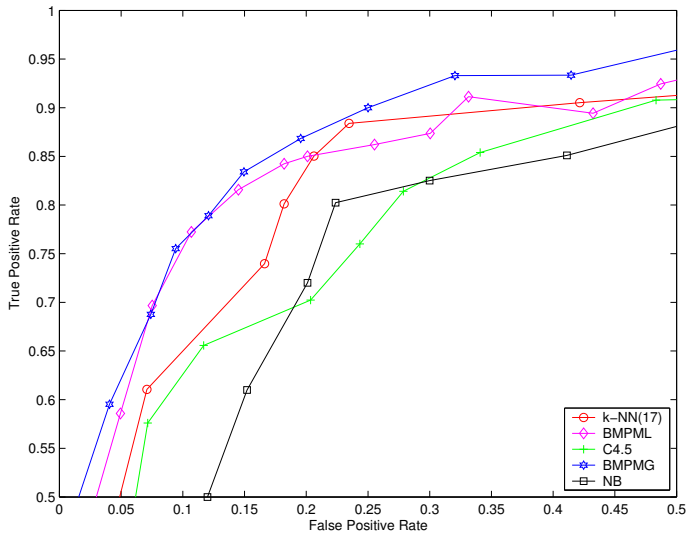


(b)

Figure 5.5: ROC curves for the breast-cancer data set. The ROC curves of the BMPML and the BMPMG dominate those of other models and the BMPMG yields the largest area under the ROC curve.



(a)



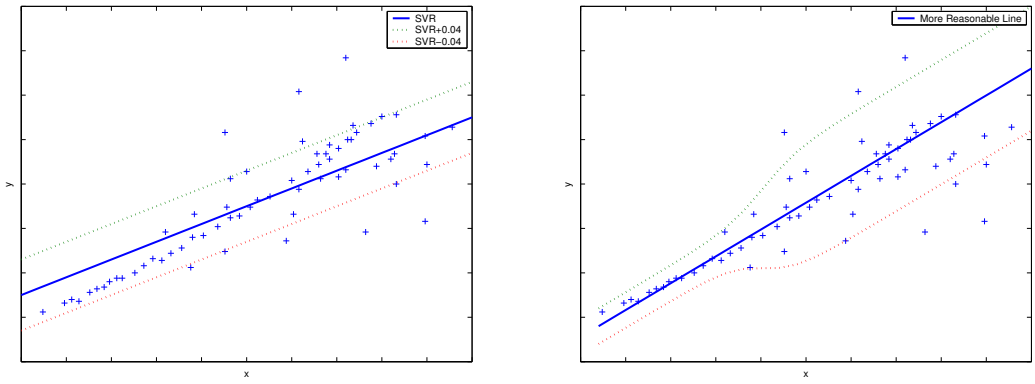
(b)

Figure 5.6: ROC curves for the heart disease data set. The ROC curves of the BMPML and the BMPMG dominate those of other models and the BMPMG yields the largest area under the ROC curve.

Chapter 6

Extension II: A Regression Model from M^4

In this chapter, we present a novel regression model, which is directly motivated from the Maxi-Min Margin Machine model described in Chapter 4. Regression is one of the problems in supervised learning. The objective is to learn a model from a given data set, $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, and then based on the learned model, to make accurate predictions of y for future values of $\mathbf{x}$. Support Vector Regression (SVR), a successful method in dealing with this problem contains the good generalization ability [154, 140, 53, 37]. The standard SVR adopts the ℓ_2-norm to control the functional complexity and chooses an ϵ-insensitive loss function with a fixed tube (margin) to measure the empirical risk. By introducing the ℓ_2-norm, the optimization problem in SVR can be transformed to a quadratic programming problem. On the other hand, the ϵ-tube has the ability to tolerate noise in data and fixing the tube enjoys the advantages of simplicity. These settings are in a global fashion and are effective in common applications, but they lack the ability and the flexibility to capture the local trend in some applications. For example, in stock markets, the data are highly volatile and the associated variance of noise varies over time. In such cases, fixing the tube cannot capture the local trend of data and cannot tolerate the noise adaptively.



(a) (b)

Figure 6.1: Illustration of the ϵ-insensitive loss function with fixed and non-fixed margins in the feature space. In (b), a non-fixed margin setting is more reasonable. It can moderate the effect of the noise by enlarging (shrinking) the margin width in the local area with large (small) variance of noise.

One typical illustration can be seen in Fig. 6.1. In this figure, the data contain larger noise as the x value of the data becomes larger. However, the SVR cannot flexibly and suitably handle it. As shown in Fig. 6.1(a), with a fixed ϵ-margin (set to 0.04), SVR considers the data globally and equally: The derived approximating function in SVR deviates from the actual data trend. On the other hand, as illustrated in Fig. 6.1(b), if we adequately consider the local volatility of data by adaptively and automatically setting a small margin in low volatile regions and a larger margin in high volatile areas, the resulting approximating function (the blue solid line in Fig. 6.1(b)) would be more suitable and reasonable.

Targeting to solve these problems, we propose the Local Support Vector Regression (LSVR) model. we will show that, with consideration of the local data trend, our model provides a systematic and automatic scheme to locally and flexibly adapt the margin. Moreover, we will also demonstrate that this novel LSVR model can derive special cases, containing a very similar physical meaning to the standard SVR. Another critical feature of our model is that

the associated optimization of LSVR can be cast as a Second Order Cone Programming (SOCP) problem, which can be efficiently solved in polynomial time [94]. The margin setting in the novel LSVR model is different from that in our previous work [162]. Concretely, the tube here is adapted directly based on the functional complexity and the local trend of data. This hence provides a more systematic and more rigorous way to moderate the margin automatically. This model can be seen as an extension into the regression model of Maxi-Min Margin Machine. In Maxi-Min Margin Machine, the main purpose is to build a classification boundary for different classes, while in LSVR, the goal is to model a function approximating the data. Therefore, M^4 considers different data trends for different classes, while LSVR focuses on employing the different data trends in different data regions. This is more valuable with the framework of regression tasks.

The rest of this chapter is organized as follows: the linear LSVR model with its theoretical background is presented in Section 6.1. In Section 6.2, we demonstrate how the standard SVR can be considered as the special case of our proposed model. In Section 6.3, we show the link between our proposed LSVR model and the general large margin classifier Maxi-Min Margin Machine. The kernelized LSVR is tackled by utilizing the Mercer's kernel in Section 6.5. Section 6.6 provides an additional interpretation on the issue of controlling the complexity of the LSVR model. Section 6.7 presents the experiments on both synthetic and real data. The chapter is concluded in Section 6.8.

6.1 A Local Support Vector Regression Model

In this section, we first present the problem and model definition of the LSVR model. We then detail its interpretation and its appealing characteristics. After that, we state its corresponding optimization method.

6.1.1 Problem and Model Definition

A basic idea to avoid overfitting in function approximation is to restrict the class of admissible solutions by a regularization term. A common method is to find a function, $f : \mathbb{R}^d \mapsto \mathbb{R}$, based on an N-instance data set $\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^d,\ y_i \in \mathbb{R},\ i = 1, \ldots, N\}$ by minimizing the following regularized functional risk

$$R_{reg}[f] = \Omega[f] + C \cdot R_{emp}[f],$$

where $C > 0$ is a regularization parameter used as the tradeoff between the minimal empirical risk, $R_{emp}[f]$, and the smoothness or functional complexity controlled by $\Omega[f]$.

Support Vector Regression is a successful regression model following this idea. It attempts to find an approximating function in the linear form:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b, \quad \mathbf{w}, \mathbf{x} \in \mathbb{R}^d, b \in \mathbb{R}. \tag{6.1}$$

For the complexity term, $\Omega[f]$, SVR selects ℓ_2-norm, or other ℓ_p-norm of $\mathbf{w}$. To measure the empirical risk, $R_{emp}[f]$, the standard SVR uses an ϵ-insensitive loss function [154].

In order to improve the flexibility of the standard SVR, we propose a new regression model, namely Local Support Vector Regression (LSVR). The objective is to learn the function in (6.1) approximating the data in $\mathcal{D}$ by making the function locally as less volatile as possible while keeping the error as small

as possible. We formulate this objective as follows:

$$\min_{\mathbf{w},b,\xi_i,\xi_i^*} \quad \frac{1}{N}\sum_{i=1}^{N}\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} + C\sum_{i=1}^{N}(\xi_i+\xi_i^*), \tag{6.2}$$

$$\text{s.t.} \quad y_i - (\mathbf{w}^T\mathbf{x}_i + b) \le \epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} + \xi_i,$$

$$(\mathbf{w}^T\mathbf{x}_i + b) - y_i \le \epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} + \xi_i^*, \tag{6.3}$$

$$\xi_i \ge 0, \quad \xi_i^* \ge 0, \quad i = 1, \ldots, N,$$

where ξ_i and ξ_i^* are the corresponding up-side and the down-side errors at the i-th point, respectively. ϵ is a positive constant. Σ_i is the covariance matrix formed by the i-th data point and those data points close to it.

6.1.2 Interpretations and Appealing Properties

In this section, beginning with stating the physical meaning of the term, $\mathbf{w}^T\Sigma_i\mathbf{w}$, we interpret our novel LSVR model.

Suppose $y_i = \mathbf{w}^T\mathbf{x}_i + b$ and $\bar{y_i} = \mathbf{w}^T\bar{\mathbf{x}}_\mathbf{i} + b$. We have the variance around the i-th data point as $\Delta_i = \frac{1}{2k+1}\sum_{j=-k}^{k}(y_{i+j} - \bar{y_i})^2 = \frac{1}{2k+1}\sum_{j=-k}^{k}(\mathbf{w}^T(\mathbf{x}_{i+j} - \bar{\mathbf{x}_i}))^2 = \mathbf{w}^T\Sigma_i\mathbf{w}$, where $2k$ is the number of data points closest to the i-th data point. Therefore, $\Delta_i = \mathbf{w}^T\Sigma_i\mathbf{w}$ actually captures the volatility in the local region around the i-th data point. In addition, Δ_i can also measure the local functional complexity around the i-th data, since it reflects the smoothness of the corresponding local region. This will be in details addressed later in section 6.6.

By using the first meaning of $\Delta_i = \mathbf{w}^T\Sigma_i\mathbf{w}$ (representing the local volatility), LSVR can systematically and automatically vary the tube: If the i-th data point lies in the area with a larger variance of noise, it will contribute to a larger $\epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}}$ or a larger local margin. This will result in reducing the impact of the noise around the point; on the other hand, in the case that the i-th data point is in the region with a smaller variance of noise, the local

margin (tube), $\epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}}$, will be smaller. Therefore, the corresponding point would contribute more in the fitting process. In comparison, the standard SVR adopts a fixed margin, which treats each point equally and therefore lacks the ability to tolerate the change in noise.

By engaging the second compelling property of $\Delta_i = \mathbf{w}^T\Sigma_i\mathbf{w}$, namely, a measure in describing the local functional complexity, LSVR controls the overall smoothness of the approximating function by minimizing the average of Δ_i as seen in (6.2). Intuitively, the margin around each point can be neither too large nor too small: If the margin is too large, the local data trend may not be captured for "over-tolerating" data; if the margin is too small, the local data trend may be "over-emphasized", resulting a highly zig-zag approximating curve. Therefore by adding the regularization term, a trade-off can be achieved via adapting the parameter C.

6.2 Connection with Support Vector Regression

We now analyze the connection of the LSVR model with the standard Support Vector Regression model. By considering the data trend globally and equally, i.e., setting $\Sigma_i = \Sigma$, for $i = 1, \ldots, N$, we can transform the optimization of (6.2) as follows:

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i,\xi_i^*} \quad & \sqrt{\mathbf{w}^T\Sigma\mathbf{w}} + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \\ \text{s.t.} \quad & y_i - (\mathbf{w}^T\mathbf{x}_i + b) \le \epsilon\sqrt{\mathbf{w}^T\Sigma\mathbf{w}} + \xi_i, \\ & (\mathbf{w}^T\mathbf{x}_i + b) - y_i \le \epsilon\sqrt{\mathbf{w}^T\Sigma\mathbf{w}} + \xi_i^*, \\ & \xi_i \ge 0, \quad \xi_i^* \ge 0, \quad i = 1, \ldots, N. \end{aligned} \tag{6.4}$$

Further, if $\Sigma = \mathbf{I}$, we obtain:

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i,\xi_i^*} \quad & \|\mathbf{w}\| + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), && (6.5) \\ \text{s.t.} \quad & y_i - (\mathbf{w}\mathbf{x}_i + b) \le \|\mathbf{w}\|\epsilon + \xi_i, \\ & (\mathbf{w}\mathbf{x}_i + b) - y_i \le \|\mathbf{w}\|\epsilon + \xi_i^*, && (6.6) \\ & \xi_i \ge 0, \quad \xi_i^* \ge 0, \quad i = 1, \ldots, N. \end{aligned}$$

The above optimization problem is very similar to the ℓ_1-norm SVR, except that it has a margin related to the complexity term. In the following, we will prove that the above optimization is actually equivalent to the ℓ_1-norm SVR in a meaningful sense.

Lemma 18 The LSVR model with setting $\Sigma_i = \mathbf{I}$ is equivalent to the ℓ_1-norm SVR in the sense that: (1) Assuming a unique ϵ_1^* exists for making ℓ_1-norm SVR optimal (i.e. setting ϵ to ϵ_1^* will make the objective function minimal), if for ϵ_1^* the ℓ_1-norm SVR achieves a solution $\{\mathbf{w}^*, b^*\} = \text{SVR}(\epsilon_1^*)$, then the LSVR can produce the same solution by setting the parameter $\epsilon = \frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}$, i.e., $\text{LSVR}(\frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}) = \text{SVR}(\epsilon_1^*)$; (2) Assuming a unique ϵ_2^* exists for making the special case of LSVR optimal (i.e. setting ϵ to ϵ_2^* will make the objective function minimal), if for ϵ_2^* the special case of LSVR achieves a solution $\{\mathbf{w}_2^*, b_2^*\} = \text{LSVR}(\epsilon_2^*)$, then the ℓ_1-norm SVR can produce the same solution by setting the parameter $\epsilon = \epsilon_2^*\|\mathbf{w}_2^*\|$, i.e., $\text{SVR}(\epsilon_2^*\|\mathbf{w}_2^*\|) = \text{LSVR}(\epsilon_1^*)$.

Proof: Since (1) and (2) are very similar statements, we only prove (1). When ϵ of the special case of LSVR is setting to $\frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}$, the value of the objective function of LSVR will be at least smaller than the one by simply setting $\{\mathbf{w}, b\} = \{\mathbf{w}_1^*, b_1^*\}$, since $\{\mathbf{w}_1^*, b_1^*\}$ is easily verified to satisfy the constraints of LSVR. Namely,

$$\text{LSVR}(\frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}) \succeq \text{SVR}(\epsilon_1^*), \tag{6.7}$$

where we use $\succeq$ to represent "superior to". We assume the solution for $\epsilon = \frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}$ in LSVR as $\{\mathbf{w}_2, b_2\}$. Similarly, by setting $\epsilon = \epsilon_1^* \frac{\|\mathbf{w}_2\|}{\|\mathbf{w}_1^*\|}$ in SVR, we have:

$$\text{SVR}(\epsilon_1^* \frac{\|\mathbf{w}_2\|}{\|\mathbf{w}_1^*\|}) \succeq \text{LSVR}(\frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}). \tag{6.8}$$

Combining (6.7) and (6.8), we have:

$$\text{SVR}(\epsilon_1^* \frac{\|\mathbf{w}_2\|}{\|\mathbf{w}_1^*\|}) \succeq \text{LSVR}(\frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}) \succeq \text{SVR}(\epsilon_1^*) \tag{6.9}$$

Since ϵ_1^* is the unique ϵ making the objective of SVR minimal, (6.9) implies that $\mathbf{w}_2 = \mathbf{w}_1^*$. ■

In addition, if in LSVR we use the item of $\mathbf{w}^T\Sigma\mathbf{w}$ instead of its square root form as the structure risk or complexity risk, a similar proof will also be applicable that the ℓ_2-norm SVR is equivalent to the special case of LSVR with $\Sigma_i = \Sigma$. In summary, we can see that the LSVR model actually contain the standard SVR model as special cases.

6.3 Link with Maxi-Min Margin Machine

The LSVR model can also be considered as an extension of the general large margin classifier, Maxi-Min Margin Machine (M^4) presented previously in this thesis or [64]. Within the framework of binary classifications for class $\mathbf{x}$ and $\mathbf{y}$, the M^4 is formulated as follows:

$$\max_{\rho, \mathbf{w} \neq \mathbf{0}, b} \quad \rho \quad \text{s.t.} \tag{6.10}$$

$$\frac{(\mathbf{w}^T\mathbf{x}_i + b)}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}} \geq \rho, \quad i = 1, 2, \ldots, N_{\mathbf{x}}\ , \tag{6.11}$$

$$\frac{-(\mathbf{w}^T\mathbf{y}_j + b)}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}} \geq \rho, \quad j = 1, 2, \ldots, N_{\mathbf{y}}\ , \tag{6.12}$$

where $\Sigma_{\mathbf{x}}$ and $\Sigma_{\mathbf{y}}$ refer to the covariance matrices of the $\mathbf{x}$ and the $\mathbf{y}$ data, respectively.

Within the framework of classifications, M^4 considers different data trends for different classes. Analogously, in the novel LSVR model, we allow different data trends for different regions, which is more suitable for the regression purpose.

6.4 Optimization Method

In order to solve the optimization problem of (6.2), we introduce auxiliary variables, $t_1,\ldots,t_N$, and transform the problem as follows:

$$\begin{aligned}
\min_{\mathbf{w},b,t_i,\xi_i,\xi_i^*} \quad & \frac{1}{N}\sum_{i=1}^{N} t_i + C\sum_{i=1}^{N}(\xi_i+\xi_i^*), && (6.13)\\
\text{s.t.} \quad & y_i - (\mathbf{w}^T\mathbf{x}_i + b) \le \epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} + \xi_i, \\
& (\mathbf{w}^T\mathbf{x}_i + b) - y_i \le \epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} + \xi_i^*, && (6.14)\\
& \sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} \le t_i, && (6.15)\\
& t_i \ge 0, \xi_i \ge 0, \xi_i^* \ge 0, i = 1,\ldots,N.
\end{aligned}$$

It is clear that (6.14) and (6.15) are non-convex constraints. This may present difficulties in optimizing the LSVR problems. In the following, we relax the optimization to a Second Order Cone Programming problem (SOCP)

problem [94] by replacing $\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}}$ with its upper bound t_i.

$$\begin{aligned}
\min_{\mathbf{w},b,t_i,\xi_i,\xi_i^*} \quad & \frac{1}{N}\sum_{i=1}^{N} t_i + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \\
\text{s.t.} \quad & y_i - (\mathbf{w}^T\mathbf{x}_i + b) \le \epsilon t_i + \xi_i, \\
& (\mathbf{w}^T\mathbf{x}_i + b) - y_i \le \epsilon t_i + \xi_i^*, \\
& \sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} \le t_i, \\
& t_i \ge 0, \xi_i \ge 0, \xi_i^* \ge 0, i = 1, \ldots, N.
\end{aligned}$$

Since t_i is closely related to $\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}}$, weighting the margin width with t_i will contain a meaning similar to the original motivation, i.e., adapting the margin flexibly. More importantly, the relaxed form is a linear programming problem under quadratic cone constraints, or more specifically it is a Second Order Cone Programming. Therefore, this problem can be solved in polynomial time by many general optimization packages, e.g., Sedumi [144, 145].

6.5 Kernelization

In this section we extend the above linear regression model to the non-linear one by using the Mercer's kernel. Suppose the training data are mapped into a kernel space or a feature space by the mapping function, $\varphi : \mathbb{R}^d \mapsto \mathbb{R}^f$. Then, the objective in the feature space is transformed as follows:

$$\begin{aligned}
\min_{\mathbf{w},b,t_i,\xi_i,\xi_i^*} \quad & \frac{1}{N}\sum_{i=1}^{N} t_i + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \\
\text{s.t.} \quad & y_i - (\mathbf{w}^T\varphi(\mathbf{x}_i) + b) \le \epsilon t_i + \xi_i, \\
& (\mathbf{w}^T\varphi(\mathbf{x}_i) + b) - y_i \le \epsilon t_i + \xi_i^*, \\
& \sqrt{\mathbf{w}^T\Sigma_i^{\varphi}\mathbf{w}} \le t_i, \\
& t_i \ge 0, \xi_i \ge 0, \xi_i^* \ge 0, i = 1, \ldots, N.
\end{aligned} \tag{6.16}$$

In order to utilize the Mercer's kernel, we first present the following theorem.

Theorem 19 If the corresponding local covariance Σ_i^{φ} can be estimated by the mapped training data, i.e., $\hat{\varphi}_i$, Σ_i^{φ} can be written as

$$\hat{\varphi}_i = \frac{1}{2k+1} \sum_{j=-k}^{k} \varphi(\mathbf{x}_{i+j}), \tag{6.17}$$

$$\Sigma_i^{\varphi} = \frac{1}{2k+1} \sum_{j=-k}^{k} (\varphi(\mathbf{x}_{i+j}) - \hat{\varphi}_i)(\varphi(\mathbf{x}_{i+j}) - \hat{\varphi}_i)^T, \tag{6.18}$$

where we just consider $2k$ data points which are the closest to the i-th data, then the optimal $\mathbf{w}$ lies in the span of the mapped training data.

Proof: Suppose $\mathbf{w} = \mathbf{w}_p + \mathbf{w}_o$, where $\mathbf{w}_p$ is the projection of $\mathbf{w}$ in the span of the mapped training data, $\mathbf{w}_o$ is the orthogonal component to the span. Since $\mathbf{w}_o^T \varphi(\mathbf{x}_i) = 0$, $i = 1, \ldots, N$, we can easily know that

$$\begin{aligned} \mathbf{w}^T \varphi(\mathbf{x}_i) &= \mathbf{w}_p^T \varphi(\mathbf{x}_i) \\ \mathbf{w}^T \Sigma_i^{\varphi} \mathbf{w} &= \mathbf{w}_p^T \Sigma_i^{\varphi} \mathbf{w}_p. \end{aligned}$$

Therefore, we can omit $\mathbf{w}_o$ since it disappears in the optimization. We then set it to $\mathbf{0}$ and obtain $\mathbf{w} = \mathbf{w}_p$, i.e., the optimal $\mathbf{w}$ lies in the span of the mapped training data. ■

By using Theorem 19, we write $\mathbf{w}$ as $\sum_{j=1}^{N} \mu_j \varphi(\mathbf{x}_j)$ and substitute it into (6.17). By rewriting (6.17) in the kernel form by a kernel function $K(\mathbf{z}_1, \mathbf{z}_2) = \varphi(\mathbf{z}_1)^T \varphi(\mathbf{z}_2)$, we then obtain

$$\begin{aligned} \mathbf{w}^T \varphi(\mathbf{x}_i) &= \sum_{j=1}^{N} \mu_j K(\mathbf{x}_i, \mathbf{x}_j) = \mu^T K_i, \\ \mathbf{w}^T \Sigma_i^{\varphi} \mathbf{w} &= \mu^T \mathbf{L}_i^T \mathbf{L}_i \mu \end{aligned}$$

where $\mu = [\mu_1, \ldots, \mu_N]^T$, $K_i = [K(\mathbf{x}_1, \mathbf{x}_i), \ldots, K(\mathbf{x}_N, \mathbf{x}_i)]^T$, $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{L}_i = \frac{1}{\sqrt{2k+1}}(K_{[i-k:i+k,N]} - \mathbf{1}_{2k+1}\mathbf{l}_i^T)$, $K_{[i-k:i+k,N]} = \begin{pmatrix} K_{i-k,1}, & \ldots, & K_{i-k,N} \\ \vdots & \ddots & \vdots \\ K_{i+k,1}, & \ldots, & K_{i+k,N} \end{pmatrix}$, $(\mathbf{l}_i^T)_t = \frac{1}{2k+1}\sum_{j=-k}^{k} K(\mathbf{x}_{i+j}, \mathbf{x}_t)$, and $\mathbf{1}_{2k+1}$ is a column vector with ones of dimension $2k+1$.

Consequently, the corresponding objective in (6.16) becomes:

$$\begin{aligned} \min_{\mu,b,t_i,\xi_i,\xi_i^*} \quad & \frac{1}{N}\sum_{i=1}^{N} t_i + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \\ \text{s.t.} \quad & y_i - (\mu^T K_i + b) \le \epsilon t_i + \xi_i, \\ & (\mu^T K_i + b) - y_i \le \epsilon t_i + \xi_i^*, \\ & \sqrt{\mu^T \mathbf{L}_i^T \mathbf{L}_i \mu} \le t_i, \\ & t_i \ge 0, \xi_i \ge 0, \xi_i^* \ge 0, i = 1, \ldots, N. \end{aligned}$$

Hence we only need a kernel function in the optimization without knowing a specific mapping function and can be easily solved by the SOCP methods.

6.6 Additional Interpretation on $\mathbf{w}^T\Sigma_i\mathbf{w}$

We now interpret in terms of sparse approximation [22, 23, 49, 31, 26, 55, 114] why $\mathbf{w}^T\Sigma_i\mathbf{w}$ can be considered as the local complexity around the data point $\mathbf{x}_i$.

In [49], Girosi has demonstrated an equivalence between sparse approximation and support vector machines. In the view of sparse approximation, the regression can be regarded as the task of approximating data using linear superpositions of basis functions selected from a large, redundant set of basis functions, called dictionary [96]. A common sense in choosing a good approximating function is that one should not only approximate the given

data as accurately as possible, more importantly, one should use as few as possible basis functions. Therefore, a sparsity concept is invoked, i.e., the approximating function should be sparse in using the basis functions. When it is connected with Support Vector Regressions, the readers can regard a basis function is associated with each data point (note that the regression function can be represented as the linear combination form in the kernel space). The fact that SVR contains the property of sparsity, i.e., only a small fraction of data points (support vectors) makes contributions to the final approximating function, may therefore explain why it has achieved a great success. The measure of sparsity of the approximating function f, which is also regarded as the measure of complexity is formulated as follows:

$$\Omega[f] = (\sum_{i=1}^{N} \delta_i)^p, \tag{6.19}$$

$$\text{where, } \delta_i = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ appears ;} \\ 0 & \text{otherwise.} \end{cases} \tag{6.20}$$

It is well-known that the ℓ_0-norm of a vector counts the number of elements different from zero, the complexity term can also be described as:

$$\Omega[f] = \|\mathbf{w}\|_{\ell_0}^p. \tag{6.21}$$

However, due to involving in minimizing a combinatorial term as the above, it is extremely difficult to perform the optimization in practice. Therefore, instead, one often uses ℓ_1-norm as its approximated version, i.e.,

$$\Omega[f] = \|\mathbf{w}\|_{\ell_1}^p. \tag{6.22}$$

when p is set to 1, it therefore leads to the standard ℓ_1-norm SVR.

When one looks back into the LSVR model, minimizing $\frac{1}{N}\sum_{i=1}^{N}\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}}$

presents another approximated version to the sparsity, since it also tries to make $\mathbf{w}$ as sparse as possible.[1] Another advantage of using $\frac{1}{N}\sum_{i=1}^{N}\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}}$ is that it leads to an easy solving method as illustrated in Section 6.4.

6.7 Experiments

In this section, we report the experiments on both synthetic *sinc* data sets and real world data sets. The SOCP problem associated with our LSVR model is solved by a general software, Sedumi [144, 145]. The SVR algorithm is performed by the LIBSVM [20].

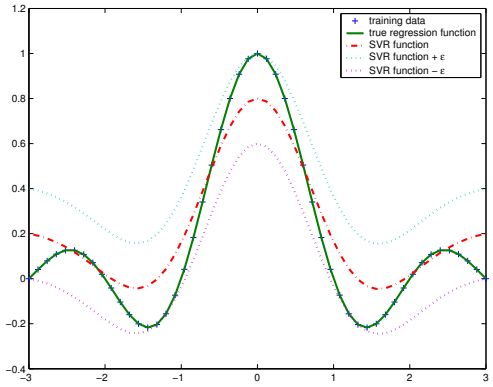
ϵ	Case I: $\sigma = 0.0$		Case II: Varying σ	
	LSVR	SVR	LSVR	SVR
0.0	0	0	0.1825±0.1011	0.3101±0.1165
0.2	0.0004	0.0160	0.2338±0.0888	0.2761±0.1111
0.4	0.0016	0.0722	0.1917±0.0726	0.2217±0.0840
0.6	0.0044	0.1695	0.1540±0.0687	0.2384±0.0867
0.8	0.0082	0.1748	0.1333±0.0674	0.2333±0.1096
1.0	0.0125	0.1748	0.1115±0.0597	0.2552±0.1218
2.0	0.0452	0.1748	0.0959±0.0421	0.2616±0.1517

Table 6.1: Experimental results (MSE±STD) of the LSVR model and the SVR algorithm on the *sinc* data with different ϵ values.

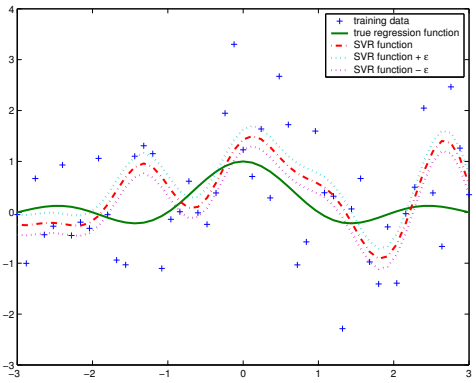
6.7.1 Evaluations on Synthetic Sinc Data

50 examples (x_i, y_i) are generated from a *sinc* function [136], where x_i are drawn uniformly from $[-3,\ 3]$, and $y_i = \sin(\pi x_i)/(\pi x_i)+\tau_i$, with τ_i drawn from a Gaussian with zero mean and variance σ^2. Two cases are evaluated. One is with $\sigma = 0$. The standard deviation of the data in the other case increases linearly from 0.5 at $x = -3$ to 1.5 at $x = 3$. It is clearly observed that in the second case, the variance of noise is different in different regions. We use the default parameters $C = 100$, the RBF kernel $K(u, v) = \exp(-\|u - v\|^2)$.

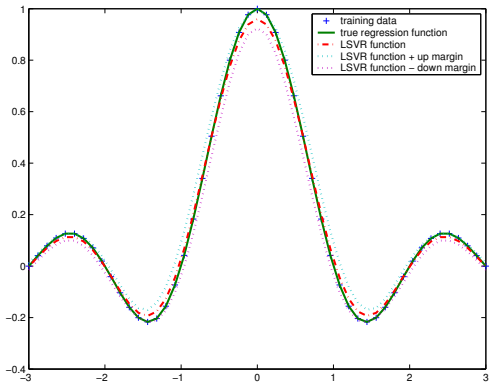
[1] Intuitively, when $\mathbf{w}$ is sparser, $\frac{1}{N}\sum_{i=1}^{N}\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}}$ would be smaller.



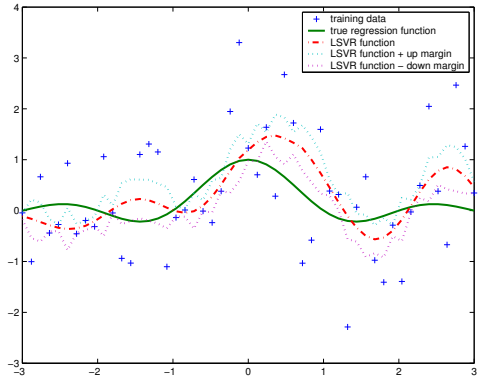
(a) SVR, σ=0.0



(b) SVR with varying σ



(c) LSVR, σ=0.0



(d) LSVR with varying σ

Figure 6.2: Experimental results on synthetic *sinc* data with ϵ=0.2.

Table 6.1 reports the average results over 100 random trails with different ϵ values. Fig. 6.2 illustrates the difference between the LSVR model and the SVR algorithm when $\epsilon = 0.2$. For the case I, $\sigma = 0.0$, the LSVR model can adjust the tube automatically to fit the data with a smaller Mean Square Error (MSE), which can be seen in Fig. 6.2(c). However, containing a fixed tube, the SVR algorithm lacks the flexibility (see Fig. 6.2(a)). This also leads that the MSE increases as ϵ increases. As reported in Table 6.1, when $\epsilon \geq 0.8$, there are no support vectors in SVR and the MSE is the largest. In the case II, the LSVR model has smaller MSE's and smaller STD's for all ϵ's. Figure 6.2(d) also shows that the obtained approximating function in LSVR is smoother than that in SVR.

Moments	DJIA		NASDAQ		S&P500	
	Train	Test	Train	Test	Train	Test
Mean	0.0000	-0.2850	-0.0000	-0.4819	0.0000	-0.3858
S.D.	1.0000	0.9957	1.0000	1.1312	1.0000	1.1298
Skew	-0.0678	0.1684	0.0928	0.3256	-0.1298	-0.0102
Kurt	2.5437	2.7706	2.6600	1.8631	2.5308	2.4124

Table 6.2: Summary statistics of normalized returns of DJIA, NASDAQ and S&P500 in the experiments. These indices show different statistical properties.

6.7.2 Evaluations on Real Financial Data

We evaluate our model on the financial time series data, which are highly volatile and non-stationary. The experimental data are three major indices: (1) the Dow Jones Industrial Average (DJIA), (2) the NASDAQ, and (3) the Standard & Poor 500 index (S&P500) in the period from January 2, 2004 to April 30, 2004. We choose this period of data because three indices data contain different statistical properties as reported in Table 6.7.1. Especially, one may note that the data in this period for three indices contain largely different skewness. In this way, the diversity in the data may not bias the comparison of the models.

Following the procedure in [120], we convert the daily closing prices (d_t)

ϵ	DJIA		NASDAQ		S&P500	
	LSVR	SVR	LSVR	SVR	LSVR	SVR
0.0	0.9204	1.3241	1.2897	1.3050	**1.2372**	1.2833
0.2	0.9835	1.1274	**1.2896**	1.3246	1.2399	1.2831
0.4	0.9341	0.9156	1.2898	1.3314	1.2442	1.2952
0.6	**0.9096**	0.9387	1.2901	1.3404	1.2540	1.2887
0.8	0.9273	0.9450	1.2904	1.3891	1.2788	1.2798
1.0	0.9434	0.9713	1.2908	1.4105	1.3044	1.2664
2.0	0.9666	1.0337	1.2928	1.3619	1.2643	1.3220

Table 6.3: Experimental results of the LSVR model and the SVR algorithm on the financial data with different ϵ values

of these indices to continuously compounded returns ($r_t = \log \frac{d_{t+1}}{d_t}$) and set the ratio of the number of the training return series to the number of test return series to 5 : 1. We perform normalization on these return series by $R_t = \frac{r_t - Mean(r_t)}{SD(r_t)}$, where the means and standard deviations are computed for each individual index in the training period.

We compare the performance of the LSVR model against the SVR. The predicted system is modelled as $\hat{R}_t = f(\mathbf{x}_t)$, where $\mathbf{x}_t$ takes the previous four days' normalized returns as indicators, i.e., $\mathbf{x}_t = (R_{t-4}, R_{t-3}, R_{t-2}, R_{t-1})$. Here this simple setting we employ is based on the suggestions in [120]: A suitable selection for the sequent values is four. We then apply the modelled function f to test the performance by one-step ahead prediction. The trade-off parameter C and the parameter of the RBF kernel ($K(\mathbf{u}, \mathbf{v}) = \exp(-\beta\|\mathbf{u} - \mathbf{v}\|^2)$), ($C$, β), are obtained by a five-fold cross-validation conducting the SVR on the following paired points: $[2^{-5}, 2^{-4}, \ldots, 2^{10}] \times [2^{-5}, 2^{-4}, \ldots, 2^{10}]$. We obtain the corresponding parameters as $(2^4, 2^{-3})$ for DJIA, $(2^{-3}, 2^1)$ for NASDAQ, and $(2^0, 2^2)$ for S&P500.

As suggested in [120], there is a relationship in the sequential five days' values. We select $k = 2$, i.e., five days' values, to model the local volatility. Since when $\epsilon \geq 2.0$, there are no support vectors in the SVR. We just set the ϵ values from $0.0, 0.2, \ldots, 1.0$, to 2.0. The corresponding results are reported in Table 6.3. As observed, the LSVR model demonstrates a consistent superiority to the SVR algorithm, even though the paired parameters (C, β) are not tuned

for our LSVR model. Furthermore, a paired t-test [107], performed on the best results of both models in Table 6.3, shows that the LSVR model outperforms SVR with $\alpha = 10\%$ significance level for a one-tailed test.

6.8 Summary

In this chapter, we propose a Local Support Vector Regression model. Different from the standard Support Vector Regression model, our novel model offers a systematic and automatic scheme to locally and flexibly adapt the margin. Therefore, it can tolerate the noise adaptively. We demonstrate that the promising model can not only captures the local information of the data in approximating functions, but also can branch out similar models to the standard SVR. The experiments conducted on *sinc* data sets and three indices data from stock markets show that our model outperforms the standard SVR. One future work of this model is to investigate efficient methods to directly solve the original optimization of LSVR instead of solving a relaxed form. In addition, both theoretical and empirical comparisons between the true solution and the approximated relaxed solution quantitatively are also valuable research topics in the future.

Chapter 7

Conclusion and Future Work

In this chapter, a summary of this thesis is provided: We will review the whole journey of this thesis, which starts from two schools of learning thought in the literatures of machine learning and then motivate the resulting combined learning thought including Maxi-Min Margin Machine, Minimum Error Minimax Probability Machine and their extensions. Following that, we then present both future perspectives within the proposed models and beyond the developed approaches.

7.1 Review of the Journey

Two paradigms exist in the literatures of machine learning. One is the school of global learning approaches; the other is the school of local learning approaches. Global learning enjoys a long and distinguished history, which usually focuses on describing phenomena by estimating a distribution from data. Based on the estimated distribution, the global learning methods can then perform inferences, conduct marginalizations, and make predictions. Although containing many good features, e.g., a relatively simple optimization, and the flexibility in incorporating global information such as structure information and invariance etc, these learning approaches have to assume a specific type of distribution a prior. However, in general, the assumption itself may be invalid. On the other

hand, local learning methods do not estimate a distribution from data. Instead, they focus on extracting only the local information, which is directly related to the learning task, i.e., the classification in this thesis. Recent progress following this trend has demonstrated that local learning approaches, e.g., Support Vector Machine (SVM) outperform the global learning methods in many aspects. Despite of the success, local learning actually discards plenty of important global information on data, e.g., the structure information. Therefore, this restricts the performance of this types of learning schemes. Motivated from the investigations of these two types of learning approaches, I therefore suggest to propose a hybrid learning framework. Namely, we should learn from data globally and locally.

Following the hybrid learning thought, I thus develop a hybrid model named Maxi-Min Margin Machine (M^4), which successfully combines two largely different but complementary paradigms. This new model is demonstrated to contain both appealing features in global learning and local learning. It can capture the global structure information from data, while it can also provide a task-oriented scheme for the learning purpose and inherits the superior performance from local learning. This model is theoretically important in the sense that M^4 contains many important learning models as special cases including Support Vector Machines, Minimax Probability Machine (MPM), and Fisher Discriminant Analysis as special cases; the proposed model is also empirically promising in that it can be cast as a Sequential Second Order Cone Programming problem, yielding a polynomial time complexity.

The idea of learning from data locally and globally is also applicable in regression tasks. Directly motivated from the Maxi-Min Margin Machine, a new regression model named Local Support Vector Regression (LSVR) is proposed in this thesis. LSVR is demonstrated to provide a systematic and automatic scheme to locally and flexibly adapt the margin, which is globally fixed in the standard Support Vector Regression (SVR), a state-of-the-art regression

model. Therefore, it can tolerate the noise adaptively. The proposed LSVR is promising in the sense that it not only captures the local information of the data in approximating functions, but more importantly, it includes special cases, which enjoy a physical meaning very much similar to the standard SVR. Both theoretical and empirical investigations demonstrate the advantages of this new model.

Besides the above two important models, another important contribution of this thesis is that we also develop a novel global learning model called Minimum Error Minimax Probability Machine (MEMPM). Although still within the framework of global learning, this model does not need to assume any specific distribution beforehand and represents a distribution-free Bayes optimal classifier in a worst-case scenario. This thus makes the model distinguished from the traditional global learning models, especially the traditional Bayes optimal classifier. One promising feature of MEMPM is that it can derive an explicit accuracy bound under a mild condition, leading to a good generalization performance for future data.

The fourth contribution of this thesis is that I develop the Biased Minimax Probability Machine (BMPM) model. Even though it is a special case of MEMPM, I highlight this model because BMPM provides the first systematic and rigorous approach for a kind of important learning tasks, namely, the biased learning or imbalanced learning. Different from traditional imbalanced (biased) learning methods, BMPM can quantitatively and explicitly incorporate a bias for one class and consequently emphasizes the more important class. A series of experiments demonstrate that BMPM is very promising in imbalanced learning and medical diagnosis.

7.2 Future Work

The models developed in this thesis bridge the gap between local learning and global learning. This brings a new viewpoint for both existing local models and global models. Following the viewpoint of learning from data both globally and locally, there seem to be a lot of immediate directions both inside and beyond the proposed models in this thesis.

7.2.1 Inside the Proposed Models

There are certainly a lot of work for improving the proposed models in this thesis.

First, all the models proposed in this thesis, including Minimum Error Minimax Probability Machine, Maxi-Min Margin Machine, and Local Support Vector Machine, involve in solving either a single Second Order Cone Programming or a Sequential Second Order Cone Programming problems. Although many optimization programs have demonstrated their good performance and mathematic tractability in solving this kind of problems, they are designed for general purposes and may not adequately exploit the specific properties in our models. Therefore, it is highly possible and valuable to develop some special optimization algorithms for speeding up their training. In particular, Maxi-Min Margin Machine and Local Support Vector Regression enjoy the feature of sparsity. By taking advantages of this property, researchers have developed fast optimization algorithms for Support Vector Machine. It is therefore very interesting to investigate whether similar procedures can be applied here. This interesting topic deserves more attentions and remains to be an open problem.

Second, an immediate problem for Minimum Error Minimax Probability Machine is the possible presence of local optimum in the practical optimization procedures. While empirical evidence shows that the global optimum can be attained in most of cases, the local optimum may occur when two types of

data are not well-separated. Conventional simulated annealing [19, 76] or deterministic annealing methods [40, 41] are certainly possible ways to attack this problem, however a formal approach, that is either a regularization augment or an algorithmic approximation, may prove more appropriate.

Third, as shown in this thesis, all the proposed models apply the kernelization trick to extend their applications into nonlinear tasks. However, it is well-known that some global information, e.g., the structure information, may not be well kept when the data are mapped from the original space to the feature space. This may restrict the power of learning from data both globally and locally. Motivated from this view, it is thus highly valuable to develop techniques to retain the global information of data when performing the projection from the original space to the feature space. This can also be considered as a task on how to choose a suitable kernel, which currently attracts much interest in the machine learning community [4, 83].

Another important future direction for the proposed classification models, i.e., Minimum Error Minimax Probability Machine and Maxi-Min Margin Machine, is how to extend the current binary classifications into multi-way classifications. Although One Vs. All and One Vs. One [2, 128] approaches present the main tools for conducting the upgrading, one always prefers to a more systematic and more rigorous approach.

7.2.2 Beyond the Proposed Models

Although several important models have been motivated and developed from the viewpoint of learning from data both globally and locally, beyond these models, there are plenty of work deserving future investigations.

One natural question is whether other famous local models or global models can be extended by engaging the viewpoint of learning from data globally and locally. For example, Neural Networks, a large family of popular learning

models, also focus on modelling data in a local fashion. It is therefore very interesting to investigate whether global information can be also incorporated into these kind of learning processes.

It is noted that the learning discussed in this thesis is restricted within the framework of either classification or regression tasks. Both tasks belong to the so-called supervised learning [7, 33, 157]. However, the other largely different learning paradigm, unsupervised learning [36, 43, 142] is not considered. Therefore, exploring possible applications of hybrid learning in this field presents a straightforward and immediate ongoing topic.

Appendix A

Proof of Lemma 2

Proof: In Marshall and Olkin Theory, if we define $\mathcal{S} = \{\mathbf{a}^T\mathbf{y} \geq b\}$, the theorem is changed to:

$$\sup_{\mathbf{y}\sim\{\overline{\mathbf{y}},\Sigma_{\mathbf{y}}\}} \mathbf{Pr}\{\mathbf{a}^T\mathbf{y} \geq b\} = \frac{1}{1+d^2}, \quad \text{with} \quad d^2 = \inf_{\mathbf{a}^T\mathbf{y}\geq\mathbf{b}} (\mathbf{y}-\overline{\mathbf{y}})^T\Sigma_{\mathbf{y}}^{-1}(\mathbf{y}-\overline{\mathbf{y}}) \ .$$

We next show that d can be obtained with as follows:

$$\begin{aligned} d^2 &= \inf_{\mathbf{a}^T\mathbf{y}\geq\mathbf{b}} (\mathbf{y}-\overline{\mathbf{y}})^T\Sigma_{\mathbf{y}}^{-1}(\mathbf{y}-\overline{\mathbf{y}}) \\ &= \frac{\max\left(b-\mathbf{a}^T\overline{\mathbf{y}},\mathbf{0}\right)^2}{\mathbf{a}^T\Sigma_{\mathbf{y}}\mathbf{a}}. \end{aligned}$$

This can be proved by using the Lagrangian multiplier method as follows

1). If $\mathbf{a}^T\overline{\mathbf{y}} \leq b$

Denoting $\mathbf{p}^T = \mathbf{a}^T\Sigma_{\mathbf{y}}^{1/2}$, $\mathbf{g} = \Sigma_{\mathbf{y}}^{-1/2}(\mathbf{y}-\overline{\mathbf{y}})$, and $q = b - \mathbf{a}^T\overline{\mathbf{y}}$, one can write $d^2 = \inf_{\mathbf{p}^T\mathbf{w}\geq q} \mathbf{g}^T\mathbf{g}$. One can obtain $\mathbf{g}$ by introducing Lagrangian multiplier:

$$\{\mathbf{g},\lambda\} = \arg\min_{\mathbf{g}} \arg\max_{\lambda} \{\mathbf{g}^T\mathbf{g} + \lambda(q-\mathbf{p}^T\mathbf{g})\},$$

where the multiplier $\lambda \geq 0$. Therefore, one can get the following equalities:

$$\mathbf{g} = \frac{\lambda \mathbf{p}}{2}, \quad q = \mathbf{p}^T \mathbf{g} \tag{A.1}$$

Since $\mathbf{a}^T \overline{\mathbf{y}} \leq b$, one can easily obtain $q \geq 0$. One can further obtain:

$$\lambda = \frac{2q}{\mathbf{p}^T \mathbf{p}}, \quad \mathbf{g} = \frac{d\mathbf{p}}{\mathbf{p}^T \mathbf{p}}.$$

Finally, this leads to the following equation:

$$\begin{aligned} d^2 &= \inf_{\mathbf{a}^T \mathbf{y} \geq \mathbf{b}} (\mathbf{y} - \overline{\mathbf{y}})^T \Sigma_{\mathbf{y}}^{-1} (\mathbf{y} - \overline{\mathbf{y}}) \\ &= \frac{(b - \mathbf{a}^T \overline{\mathbf{y}})^2}{\mathbf{a}^T \Sigma_{\mathbf{y}} \mathbf{a}}. \end{aligned}$$

2). If $\mathbf{a}^T \overline{\mathbf{y}} \geq b$

In this case, we can only have $\mathbf{y} = \overline{\mathbf{y}}$. Therefore, $d = 0$.

By integrating the above, we thus complete the proof of this theorem ■

Appendix B

A Simple Manual for the Package of MEMPM Version 1.0

B.1 Starting MEMPM Version 1.0

- Unzip MEMPM-1.0.zip

- Run *mempm_demo* in Matlab to see the demo of MEMPM.
  The *mempm_demo* function is designed step by step interactively. You may press any key to continue it. The m-file *mempm_demo.m* is designed to demonstrate on how to use the MEMPM functions, it is commented and explained so that you can use the code conveniently.

- Run *robust_demo* in Matlab to see the demo of robost MEMPM.
  The *robust_demo.m* function demonstrates how to use the robust version of MEMPM and MPM (with $\nu_{\mathbf{x}} \neq \nu_{\mathbf{y}}$) functions. It is commented and explained so that you can use the code conveniently with your own data.

B.2 An Overview of the Files

- build_MEMPM_lin_bi_QI.m

  This function is designed to train the Minimum Error Minimax Probability Machine (the linear version) for binary classifications by using Sequential Biased Minimax Probability Machine method. The line search is performed by the Quadratic Interpolation method and the Biased Minimax Probability Machine is performed by the Rosen Gradient projection method.

- solve_FP_RG.m

  This function is designed to solve a Fractional Programming problem by the Rosen Gradient projection method.

- build_MEMPM_k_bi_QI.m

  This function is designed to train the Minimum Error Minimax Probability Machine (the kernelized version) for binary classifications using the Sequential Biased Minimax Probability Machine method, the line search is performed by the Quadratic Interpolation method.

- solve_FP_PM.m

  This function is designed to solve a Fractional Programming problem by the Parametric Method.

- build_BMPM_k_bi_PM.m

  This function is designed to train a Biased Minimax Probability Machine (the kernelized version) for binary classifications by using the Parametric Method to solve the Fractional Programming problem.

- build_robMEMPM_lin_bi_QI.m

  This function is designed to train a robust Minimum Error Minimax

Probability Machine (the linear version) for binary classifications by using sequential Biased Minimax Probability Machine method, the line search is performed by the Quadratic Interpolation method.

- solv_robF_RG.m

  This function is designed to solve a Fractional Programming problem with robust parameters by the Rosen Gradient projection method.

- build_robMEMPM_k_bi_QI.m

  This function is designed to train a robust Minimum Error Minimax Probability Machine (the kernelized version) for binary classifications by using sequential Biased Minimax Probability Machine method. The line search is performed by the Quadratic Interpolation method.

- solve_robFP_PM.m

  This function is designed to solve a Fractional Programming problem with robust parameters by the Parametric Method.

- build_robBMPM_k_bi_PM.m

  This function is designed to train a robust Biased Minimax Probability Machine (BMPM, kernelized version) for binary classification by using the Parametric Method to solve the Fractional Programming problem.

- build_robMPMn_lin_bi_PM.m

  This function is designed to train a robust linear Minimax Probability Machine for binary classifications with $\nu_{\mathbf{x}} \neq \nu_{\mathbf{y}}$, it is implemented by solving a Fractional Programming problem using parametric method.

- build_robMPM_k_bi_PM.m

  This function is designed to train a robust Minimax Probability Machine (the kernelized version) for binary classifications with $\nu_{\mathbf{x}} \neq \nu_{\mathbf{y}}$. It is implemented by solving a Fractional Programming problem using parametric method.

- eval_lin_bi.m

  This function is designed to evaluate all kinds of probability machines (linear version), including Minimax Probability Machine , Biased Minimax Probability Machine , Minimum Error Minimax Probability Machine, for binary classifications.

- eval_k_bi.m

  This function is designed to evaluate all probability machines (kernelized version), including Minimax Probability Machine, Biased Minimax Probability Machine, Minimum Error Minimax Probability Machine, for binary classifications.

- mempm.mat:

  This data file contains the data for the demonstration in .mat format

- mempm_demo.m:

  This function is designed to demonstrate on how to use the MEMPM toolbox.

- robust_demo.m:

  This function is designed to demonstrate on how to use the robust version of MEMPM and MPM with unequal parameters.

Appendix C

Publication List

Here is the publication list during my Ph.D. study:

1. **Kaizhu Huang**, Haiqin Yang, Irwin King, Michael R. Lyu, and Laiwan Chan. Minimum Error Minimax Probability Machine. Journal of Machine Learning Research, 2004. Accepted with minor revision.

2. **Kaizhu Huang**, Haiqin Yang, Irwin King, Michael R. Lyu. M^4: Learning Large Margin Machines Locally and Globally. Submitted to Journal of Machine Learning, 2004.

3. **Kaizhu Huang**, Irwin King, and Michael R. Lyu, Haiqin Yang. Improving Chow-Liu tree performance based on association rules. In Neural Information Processing: Research and Development. Series: Studies in Fuzziness and Soft Computing, PP94-112, Vol. 152. Rajapakse, Jagath C.; Wang, Lipo (Eds.). Springer-Verlag. 2004.

4. Haiqin Yang, Irwin King, Laiwan Chan and **Kaizhu Huang**. Financial Time Series Prediction Using Non-fixed and Asymmetrical Margin Setting with Momentum in Support Vector Regression. In Neural Information Processing: Research and Development. Series: Studies in Fuzziness and Soft Computing, PP334-150, Vol. 152. Rajapakse, Jagath C.; Wang, Lipo (Eds.). Springer-Verlag. 2004.

5. **Kaizhu Huang**, Haiqin Yang, Irwin King, and Michael R. Lyu, Imbalanced learning with Biased Minimax Probability Machine. Submitted to IEEE Trans. On System, Man, Cybernetics, Part B, 2004.

6. **Kaizhu Huang**, Haiqin Yang, Irwin King, and Michael R. Lyu, Maximizing Sensitivity in Medical Diagnosis Using Biased Minimax Probability Machine. Submitted to Decision Support Systems, 2004.

7. **Kaizhu Huang**, Haiqin Yang, Irwin King, Michael R. Lyu. Learning Large Margin Machines Locally and Globally. In Proceedings international Conference on Machine Learning (ICML' 2004), Banff, Canada, 2004.

8. **Kaizhu Huang**, Haiqin Yang, Irwin King, Michael R. Lyu. Learning Classifiers from Imbalanced Data Based on Biased Minimax Probability Machine. In Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR' 2004), Washington, DC, June 27 - July 2, 2004.

9. Chu-Hong Hoi, Chi-Hang Chan, **Kaizhu Huang**, Michael R Lyu and Irwin King. Biased Support Vector Machine for Relevance Feedback in Image Retrieval. In Proceedings of International Joint Conference on Neural Networks (IJCNN'2004), Budapest, Hungary, 25-29 July, 2004.

10. **Kaizhu Huang**, Haiqin Yang, Irwin King, Michael R. Lyu, and Laiwan Chan. Biased Minimax Probability Machine for Medical Diagnosis. The 8th International Symposium on Artificial Intelligence and Mathematics (AMAI'2004), Fort Lauderdale, Florida, January 4-6, 2004.

11. Haiqin Yang, **Kaizhu Huang**, Irwin King, and Michael R. Lyu. Varying the Tube: A Local Support Vector Regression Model. Submitted.

12. Haiqin Yang, **Kaizhu Huang**, Laiwan Chan, Irwin King, and Michael R. Lyu. Outliers Treatment in Support Vector Regression for Financial Time Series Prediction. Accepted by the International Conference on Neural Information Processing (ICONIP'2004).

13. **Kaizhu Huang**, Irwin King, and Michael R. Lyu. Finite Mixture Model of Bounded Semi-Naive Bayesian Networks Classifiers. ICANN'-2003, LNCS PP115-122, Long paper, Publisher: Springer-Verlag Heidelberg, Volume 2714 / 2003, January 2003.

14. **Kaizhu Huang**, Irwin King, and Michael R. Lyu. Discriminative Training of Bayesian Chow-Liu Tree Multinet Classifiers. In Proceedings of the International Joint Conference on Neural Networks (IJCNN'2003), Volume: 1 , July 20 - 24, 2003 page(s): 484 -488.

15. **Kaizhu Huang**, Irwin King, and Michael R. Lyu. Learning maximum likelihood semi-naive bayesian network classifier. In 2002 IEEE International Conference on Systems, Man and Cybernetics, Volume: 3, 6-9 Oct. 2002 Page(s): 6 pp. vol.3, Hamammet, Tunisia, October 6-9, 2002.

16. **Kaizhu Huang**, Irwin King, and Michael R. Lyu. Constructing a large node Chow-Liu Tree based on frequent itemsets. In Lipo Wang, Jagath C. Rajapakse, Kunihiko Fukushima, Soo-Young Lee, and Xi Yao, editors, Proceedings to the International Conference on Neural Information Processing (ICONIP'2002), page 498 -502, Orchid Country Club, Singapore, November 18-22, 2002. Nanyang Technological University.

Bibliography

[1] D. Aha, D. Kibler, and M. Albert. Instance-based learning algorithms. 6:37–66, 1991.

[2] E. L. Allwein, R. E. Schapire, , and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

[3] R. Anand, G. K. Mehrotram, K. C. Mohan, and S. Ranka. An improved alogrithm for neural network classification of imbalance training sets. *IEEE Transactions on Neural Networks*, 4(6):962–969, 1993.

[4] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of International Conference on Machine Learning (ICML-2004)*, 2004.

[5] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer. Estimating hidden markov model parameters so as to maximize speech recognition accuracy. IEEE *Transactions on Speech and Audio Processing*, 1:77–82, 1993.

[6] C. B. Barber, D. P. Dobkin, and H. Huhanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.

[7] P. L. Bartlett. Learning theory and generalization for neural networks and other supervised learning techniques. In *Neural Information Processing Systems Tutorial*, 1998.

[8] L. E. Baum and J. A. Egon. An inequality with applications to statistical estimation for proba-bilistic functions of a markov process and to a model for ecology. *Bull. Amer. Meteorol. Soc.*, 73:360C363, 1967.

[9] M. S. Bazaraa. *Nonlinear Programming: Theory and Algorithms*. New York: Wiley, 2nd edition, 1993.

[10] F. Beaufays, M. Wintraub, and Y. Konig. Discriminative mixture weight estimation for large gaussian mixture models. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 337–340, 1999.

[11] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massashusetts, 2nd edition, 1999.

[12] C. L. Blake and C. J. Merz. Repository of machine learning databases, University of California, Irvine, http://www.ics.uci.edu/∼mlearn/mlrepository.html, 1998.

[13] H. Bozdogan. *Statistical data mining and knowledge discovery*. Boca Raton, Fla. : Chapman & Hall/CRC, 2004..

[14] A. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithm. *Pattern Recognition*, 30(7):1145–1159, 1997.

[15] M. Brand. Structure discovery via entropy minimization. In *Neural Information Processing System 11*, 1998.

[16] L. Breiman. Arcing classifiers. Technical Report 460, Statistics Department, University of California, 1997.

[17] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[18] C. Cardie and N. Howe. Improving minority class prediction using case specific feature weights. In *Proceedings of the Fourteen International Conference on Machine Learning (ICML-1997)*, pages 57–65. San Francisco, CA: Morgan Kaufmann, 1997.

[19] V. Cerny. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *JJ. Opt. Theory Appl.*, 45(1):41–51, 1985.

[20] C.-C. Chang and C.-J Lin. LIBSVM: a Library for Support Vector Machines, 2001.

[21] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer. Smote:sythetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[22] S. Chen. *Basis Pursuit*. PhD thesis, Department of Statistics, Standform University, 1995.

[23] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis puisuit. Technique Report 479, Department of Statistics, Standform University, 1995.

[24] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Information Theory*, 14:462–467, 1968.

[25] Y. S. Chow and H. Teicher. *Probability theory: Independence, interchageability, martingales*. Springer-Verlag, New York, 3 edition, 1997.

[26] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best-basis selection. *IEEE Transactions on Information Theory*, (38):713–718, 1992.

[27] T.M. Cover and P.E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, IT-13(1):21–27, 1967.

[28] B. D. Craven. *Mathematical Programming and Control Theory*. Chapman and Hall, London, 1978.

[29] B. D. Craven. *Fractional Programming, Sigama Series in Applied Mathematics 4*. Heldermann Verlag, Berlin, 1988.

[30] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines(and Other Kernel-based Learning Methods)*. Cambridge University Press, Cambridge, U.K.; New York, 2000.

[31] I. Daubechies. Ten lectures on wavelets. In *CBMS-NSF Regional Conferences Series in Applied Mathematics*, SIAM Philadelphia, PA, 1992.

[32] G. Deco and D. Obradovic. *An information-theoretic approach to neural computing*. Springer-Verlag, 1996.

[33] T. G. Dietterich. Machine learning research: Four current directions. *AI Magazine*, 18(4):97–136, 1997.

[34] K. Dorfman, D. Berbaum and C. Metz. Receiver operating characteristic rating analysis:generalization to the population of readers and patients with the jackknife method. *Investigative Radiology*, 27:723–731, 1992.

[35] D. Dori and W. Liu. Sparse pixel vectorization: An algorithm and its performance evaluation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21:202–215, 1999.

[36] James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In *International Conference on Machine Learning*, pages 194–202, 1995.

[37] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. N. Vapnik. Support Vector Regression Machines. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 155–161. The MIT Press, 1997.

[38] R. Duda and P. Hart. In *Pattern classification and scene analysis*. John Wiley & Sons, 1973.

[39] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2000.

[40] G. Dueck. New optimization heuristics :the great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 104:86–92, 1993.

[41] G. Dueck and T. Scheurer. Threshold accepting: A general purpose optimization algorithm. *Journal of Computational Physics*, 90:161–175, 1990.

[42] L. Fausett. *Fundamentals of Neural Networks.* New York: Prentice Hall., 1994.

[43] M Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *Transaction on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

[44] O. Firschein and T. Strat. *RADIUS:Image understanding for imagery intelligence*. San Francisco, CA:Morgan Kaufmann, 1996.

[45] D. A. Forsyth and J. Ponce. *Computer vision : a modern approach*. Upper Saddle River, N.J. : Prentice Hall, 2003.

[46] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–161, 1997.

[47] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, 2nd edition, 1990.

[48] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. London : Chapman & Hall, 1996.

[49] F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, 10(6):1455–1480, 1998.

[50] M. G. Gonzalez, R. C. ; Thomason. *Syntactic pattern recognition : an introduction*. Reading, Mass. : Addison-Wesley Pub. Co., Advanced Book Program, 1978.

[51] P. Grzegorzewski, O. Hryniewicz, and M. Gil. *Soft methods in probability, statistics and data analysis*. Physica-Verlag, Heidelberg ; New York, 2002.

[52] J. W. Grzymala-Busse, L. K. Goodwin, and X. Zhang. Increasing sensitivity of preterm birth by changing rule strengths. *Pattern Recognition Letters*, (24):903–910, 2003.

[53] S. Gunn. Support vector machines for classification and regression. Technical Report NC2-TR-1998-030, Faculty of Engineering and Applied Science, Department of Electronics and Computer Science, University of Southampton, May 1998.

[54] D. Hackman, C. Meek, and G. Cooper. A tutorial on learning bayesian networks. In *Tech Report MSR-TR-95-06*. Microsoft Research, 1995.

[55] G. F. Harpur and R. W. Prager. Development of low entropy coding in a recurrent network. *Networs*, 7:277–284, 1996.

[56] T. Hastie and R. Tibshirani. Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society(B)*, 58:155–176, 1996.

[57] S. Haykin. *Neural Networks: A Comprehensive Foundation.* New York: Macmillan Publishing., 1994.

[58] R. Herbrich and T. Graepel. Large scale Bayes point machines. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

[59] K. Huang, I. King, L. Chan, and H. Yang. Improving Chow-Liu tree performance based on association rules. In J. C. Rajapakse and L. Wang, editors, *Neural Information Processing: Research and Development*, volume 152 of *Studies in Fuzziness and Soft Computing*, pages 94–112. Springer-Verlag, 2004.

[60] K. Huang, I. King, and M. R. Lyu. Finite mixture model of bound semi-naive bayesian network for classification. *Technique report, Department of Computer Science and Enginerring, the Chinese University of Hong Kong*, 2002.

[61] K. Huang, I. King, and M. R. Lyu. Learning maximum likelihood semi-naive bayesian network classifier. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC2002), Hammamet, Tunisia*, page TA1F3, 2002.

[62] K. Huang, I. King, and M. R. Lyu. Discriminative training of bayesian chow-liu tree multinet classifiers. In *Proceedings of International*

Joint Conference on Neural Network (IJCNN-2003), Oregon, Portland, U.S.A., volume 1, pages 484–488, 2003.

[63] K. Huang, I. King, and M. R. Lyu. Finite mixture model of bound semi-naive bayesian network classifier. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN-2003), Lecture Notes in Artificial Intelligence, Long paper*, volume 2714, pages 115–122. Springer-Verlag Heidelberg, 2003.

[64] K. Huang, H. Yang, I. King, and M. R. Lyu. Learning large margin classifiers locally and globally. In *The Twenty-First International Conference on Machine Learning (ICML-2004), Banff, Alberta, Canada, accepted*, 2004.

[65] K. Huang, H. Yang, I. King, M. R. Lyu, and L. Chan. Biased minimax probability machine for medical diagonsis. In *the Eighth International Symposium on Artificial Intelligence and Mathematics*, 2003.

[66] K. Huang, H. Yang, I. King, M. R. Lyu, and L. Chan. Maxi-min margin machine: Learning large margin classifiers globally and locally. *Journal of Machine Learning Research, submitted*, 2004.

[67] K. Huang, H. Yang, I. King, M. R. Lyu, and L. Chan. Minimum error minimax probability machine. *Journal of Machine Learning Research, under revision*, 2004.

[68] T. Ibaraki. Solving mathematical programming problems with fractional objective functions. In S. Schaible and W. T. Ziemba., editors, *Generalized concavity in optimization and economics*, pages 441–472. Academic Press, New York, 1981.

[69] V. V. Ivannov. On linear problems which are not well-posed. *Soviet Math. Docl.*, 3(4):981–983, 1962.

[70] T. Jebara. *Discriminative, Generative and Imitative Learning.* PhD thesis, Massachusetts Institute of Technology, 2002.

[71] M. I. Jordan. Why the logistic function? A tutorial discussion on probabilities and neural networks. Technical Report 9503, MIT Computational Cognitive Science Report, 1995.

[72] M. I. Jordan. *Learning in Graphical models.* Kluwer Academic Publishers, 1998.

[73] G.T. Toussaint J.W. Jaromczyk. Relative neighborhood graphs and their relatives. *Proceedings IEEE*, 80(9):1502–1517, 1992.

[74] R. E. Kass, B. P. Carlin, A. Gelman, and R. M. Neal. Markov chain monte carlo in practice: A roundtable discussion. *The American Statistician*, 52:93–100, 1998.

[75] D. Keysers, F. J. Och, and H. Ney. Maximum entropy and gaussian models for image object recognition. In *Proceedings of the 24th DAGM Symposium, Volume LNCS 2449 of Lecture Notes in Computer Science*, pages 498–506. Springer-Verlag, 2002.

[76] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[77] R. Kohavi. A study of cross validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the fourtheenth International Joint Conference on Artificial Intelligence (IJCAI-1995)*, pages 338–345. San Francisco, CA:Morgan Kaufmann, 1995.

[78] R. Kohavi, B. Becker, and D Sommerfield. Improving simple bayes. In *Technique report.* Data Mining and Visualization Group, Silicon Graphics Inc., Mountain View, CA., 1997.

[79] S. Kruk and H. Wolkowicz. General nonlinear programming. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, pages 563–575. Kluwer Academic Publishers, Boston, 2000.

[80] M. Kubat, R. Holte, and S. Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, (30):195–215, 1998.

[81] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings of the Fourteen International Conference on Machine Learning (ICML-1997)*, pages 179–186. San Francisco, CA: Morgan Kaufmann, 1997.

[82] N. M. Laird, A. P. Dempster, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Society*, B39:1–38, 1977.

[83] G. R. G. Lanckriet, N. Cristianini, L. El Ghaoui, P. L. Bartlett, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 2004.

[84] G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan. Minimax probability machine. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

[85] G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3:555–582, 2002.

[86] G. R. G. Lanckriet, L. E. Ghaoui, and M. I. Jordan. Robust novelty detection with single-class mpm. In *Advances in Neural Information Processing Systerm (NIPS)*, 2002.

[87] P. Langley. Induction of recursive bayesian classifiers. *In Proceedings of the 1993 European Conference on Machine learning*, pages 153–164, 1993.

[88] P. Langley, W. Iba, and K. Thompson. An analysis of bayesian classifiers. *In Proceedings of National Conference on Artificial Intelligence*, pages 223–228, 1992.

[89] B. Lerner and N. D. Lawrence. A comparison of state-of-the-art classification techniques with application to cytogenetics. *Neural Computing and Applications*, 10(1):39–47, 2001.

[90] D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the Eleventh International Conference on Machine Learning (ICML-1994)*, pages 148–156. San Francisco, CA: Morgan Kaufmann, 1994.

[91] C. Lin and R. Nevatia. Building detection and description from a single intensity image. *Computer Vision and Image Understanding*, 72:101–121, 1998.

[92] C. Ling and C. Li. Data mining for direct marketing:problems and solutions. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-1998)*, pages 73–79. Menlo Park, CA:AAAI Press, 1998.

[93] W. Liu and D. Dori. A protocol for performance evaluation of line detection algorithms. *Machine Vision and Application*, 9:240–250, 1997.

[94] M. Lobo, L. Vandenberghe, , S. Boyd, and H. Lebret. Applications of second order cone programming. *Linear Algebra and its Applications*, 284:193–228, 1998.

[95] A. Luntz and V. Brailovsky. On estimation of characters obtained in statistical procedure of recognition (in Russian). *Technicheskaya Kibernetica*, (6), 1969.

[96] S. Mallat and Z. Zhang. Matching pursuit in a time-frequency dictionary. *IEEE Transactions on Signal Processing*, (41):3397–3415, 1993.

[97] M. A. Maloof. On machine learning, ROC analysis, and statistical tests of significance. In *Proceedings of the Sixteenth International Conference on Pattern Recognition*, pages 204–207, Los Alamitos, CA, 2002. IEEE Press.

[98] M. A. Maloof. Learning when data sets are imbanlanced and when costs are unequal and unknown. In *Proceedings of International Conference on Machine Learning (ICML-2003)*, 2003.

[99] M. A. Maloof, P. Langley, T. O. Binford, R. Nevatia, and S. Sage. Improved rooftop detection in aerial images with machine learning. *Machine Learning*, 53:157–191, 2003.

[100] Olvi L. Mangasarian. *Nonlinear programming*. Philadelphia : Society for Industrial and Applied Mathematics, 1994.

[101] A. W. Marshall and I. Olkin. Multivariate chebyshev inequalities. *Annals of Mathematical Statistics*, 31(4):1001–1014, 1960.

[102] D. Mcclish. Analyzing a portion of the roc curve. *Medical Decision Making*, (9):190–195, 1989.

[103] G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker Inc., New York, 1988.

[104] Pankaj Mehra and Benjamin W. Wah. *Artificial neural networks : concepts and theory.* Los Alamitos, California: IEEE Computer Society Press, 1992.

[105] X. L. Meng and D. B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2), 1993.

[106] T. Minka. *A family of algorithms for approximate Inference.* PhD thesis, Massachusetts Institute of Technology, 2001.

[107] Douglas C. Montgomery and George C. Runger. *Applied statistics and probability for engineers.* Wiley, New York, 2nd edition, 1999.

[108] Hervé Moulin. *Cooperative microeconomics : a game-theoretic introduction.* Princeton University Press, Princeton, N.J., 1995.

[109] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schö1kopf. An introduction to Kernel-Based Learning Algorithms. *IEEE Transactions on Neural Networks*, 12:181–201, 2001.

[110] R. M. Neal. Probabilistic inference using markov chain monte carlo methods. *Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto*, 1993.

[111] R. M. Neal. Suppressing random walks in markov chain monte carlo using ordered overrelaxation. *M. I. Jordan (editor) Learning in Graphical Models*, Dordrecht: Kluwer Academic Publishers:205–225, 1998.

[112] Y. Nesterov and A. Nemirovsky. Interior point polynomial methods in convex programming: Theory and applications. *SIAM, Philadelphia, PA.*, 1994.

[113] A. S. Nugroho, S. Kuroyanagi, and A. Iwata. A solution for imbalanced training sets problem by combnet and its application on fog forecasting. *IEICE TRANS. INF. & SYST.*, E85-D(7), 2002.

[114] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, (381):607–609, 1996.

[115] E. Osuna, R. Freund, and F. Girosi. Support Vector Machines: Training and Applications. Technical Report AIM-1602, MIT, 1997.

[116] D. Patterson. *Artificial Neural Networks.* Singapore: Prentice Hall., 1996.

[117] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: networks of plausible inference*. Morgan Kaufmann, CA, 1988.

[118] R. L. Pinto and R. M. Neal. Improving markov chain monte carlo estimators by coupling to an approximating chain. *Technical Report No. 0101, Dept. of Statistics, University of Toronto*, 2001.

[119] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. *Technical Report MSR-TR-98-14*, 1998.

[120] Bernd Pompe. Mutual information and relevant variables for predictions. In Abdol S. Soofi and Liangyue Cao, editors, *Modelling and forecasting financial data: techniques of nonlinear dynamics*, pages 61–92. Kluwer Academic Publishers, Boston, Mass., 2002.

[121] I. Popescu and D. Bertsimas. Optimal inequalities in probability theory: A convex optimization approach. Technical Report TM62, INSEAD, 2001.

[122] F. Provost. Learning from imbanlanced data sets. In *Proceedings of The Seventeenth National Conference on Artificial Intelligence (AAAI 2000)*, 2000.

[123] F. Provost and T. Fawcett. Analysis and visulization of classifier performance:comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 43–48. Menlo Park, CA:AAAI Press, 1997.

[124] A. Pruessner. Conic programming in GAMS. In *Optimization Software - The State of the Art*. INFORMS Atlanta, http://www.gamsworld.org/cone/links.htm, 2003.

[125] J. R. Quinlan. *C4.5 : programs for machine learning*. San Mateo, California:Morgan Kaufmann Publishers, 1993.

[126] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257C286, 1989.

[127] C. Rathinavelu and L. Deng. The trended HMM with discriminative training for phonetic classification. In *Proceedings of ICSLP*, 1996.

[128] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

[129] B.D. Ripley. *Pattern Recognition and Neural Networks*. Press Syndicate of the University of Cambridge, 1996.

[130] R. Rujam. Preceptron learning by playing billiards. *Neural Computation*, 9:99–122, 1997.

[131] S. J. Russell and P. Norvig. *Artificial intelligence : a modern approach*. Englewood Cliffs, N.J. : Prentice Hall, 1995.

[132] Jaakkola Tommo S. and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems (NIPS)*, 1998.

[133] S. Schaible. Fractional programming. *Zeitschrift für Operational Research*, Serie A 27(1):39–54, 1977.

[134] S. Schaible. *Fractional programming*. Nonconvex Optimization and its Applications. Kluwer Academic Publishers, Dordrecht-Boston-London, 1995.

[135] P. Schmidt and A. Witte. *Predicting recidivism using survival models*. New York, NY:Spring-Verlag, 1988.

[136] B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson. Shrinking the Tube: A New Support Vector Regression Algorithm. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11, pages 330 – 336, Cambridge, MA, 1999. MIT Press.

[137] B. Schölkopf, C. Burges, and A. Smola, editors. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, Massachusetts, 1999.

[138] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[139] B. Schölkopf and A. Smola, editors. *Learning with kernels: support vector machines, regularization, optimization and beyond*. MIT Press, Cambridge, Massachusetts, 2002.

[140] A. Smola and B. Schölkopf. A tutorial on support vector regression. Technical Report NC2-TR-1998-030, NeuroCOLT2, 1998.

[141] A. J. Smola, P. L. Bartlett, B. Scholkopf, and D. Schuurmans. *Advances in Large Margin Classifiers*. The MIT Press, 2000.

[142] H. Steck and T. Jaakkola. Unsupervised active learning in large domains. In *In Proceedings of the Eighteenth Annual Conference on Uncertainty in Artificial Intelligence*, 2002.

[143] A. Stolcke and S. Omohundro. Hidden markov model induction by bayesian model merging. In *NIPS 5*, pages 11–18, 1993.

[144] J. F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11:625–653, 1999.

[145] J. F. Sturm. Central region method. In J. B. G. Frenk, C. Roos, T. Terlaky, and S. Zhang, editors, *High Performance Optimization*, pages 157–194. Kluwer Academic Publishers, 2000.

[146] J. Swets. Measureing the accuracy of diagnostic systems. *Science*, (240):1285–1293, 1988.

[147] J. Swets and R. Pickett. *Evaluation of diagnoistic systems: Methods from signal detection theory*. New York, NY: Springer-Verlag, 1982.

[148] B. Thiesson, C. Meek, and D. Heckman. Learning mixtures of bayesian networks. In *Technique Report, MSR-TR-97-30*. Microsoft Research, 1998.

[149] A. N Tikhonov. On solving ill-posed problem and method of regularization. *Doklady Akademii Nauk USSR*, 153:501–504, 1963.

[150] M. Tipping. The relevance vector machien. In *Advances in Neural Information Processing Systems 12 (NIPS)*, 1999.

[151] S. Tong and D. Koller. Restricted bayes optimal classifiers. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI), Austin, Texas, August 2000*, pages 658–664, 2000.

[152] P. K. Trivedi. Estimation of a distributed lag model under quadratic loss. *Econometrica*, 46(5):1181–1192, 1978.

[153] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

[154] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 2nd edition, 1999.

[155] V. V. Vasin. Relationship of several variational methods for approximate solutions of ill-posed problems. *Math. Notes*, 7:161–166, 1970.

[156] J. Weizenbaum. Eliza- a computer program for the study of natural language communication between man and machine. *Communications of the Association for Computing Machinery*, 1966.

[157] Hannes. Wettig, Peter Grunwald, and Teemu Roos. Supervised naive bayes parameters. In P. Ala-siuru and S. Kasko., editors, *The Art of Natural and Artificial:Proceedings of the* 10*th Finnish Artificial Intelligence Conference*, pages 72–83, 2002.

[158] P. Woodland and D. Povey. Large scale discriminative training for speech recognition. In *Proceedings of ASR 2000*, 2000.

[159] Kevin Woods, W.Philip Kegelmeyer Jr., and Kevin Bowyer. Combination of multiple classifiers using local accuracy estimates. *IEEE Tansactions on Pattern Analysis and Machine Intelligence*, 19(4):405–410, April 1997.

[160] H. Yang, K. Huang, I. King, M. R. Lyu, and L. Chan. Matlab toolbox for biased minimax probability machine (bmpm1.0).

http://www.cse.cuhk.edu.hk/∼miplab/mempm_toolbox/index.htm, 2004.

[161] H. Yang, K. Huang, I. King, M. R. Lyu, and L. Chan. Matlab toolbox for minimum error minimax probability machine (mempm1.0). http://www.cse.cuhk.edu.hk/∼miplab/mempm_toolbox/index.htm, 2004.

[162] H. Yang, I. King, L. Chan, and K. Huang. Financial Time Series Prediction Using Non-fixed and Asymmetrical Margin Setting with Momentum in Support Vector Regression. In J. C. Rajapakse and L. Wang, editors, *Neural Information Processing: Research and Development*, volume 152 of *Studies in Fuzziness and Soft Computing*, pages 334–350. Springer-Verlag, 2004.

[163] J. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propogation. In *Neural Information Processing Systems 13*, 2000.

[164] W. Zhang and I. King. A study of the relationship between support vector machine and gabriel graph. In *In Proceedings of IEEE World Congress on Computational Intelligence—International Joint Conference on Neural Networks*, 2002.

[165] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm support vector machines. In *Advances in Neural Information Processing Systems (NIPS 16)*, 2003.