JMLR: Workshop and Conference Proceedings 25:145–157, 2012 Asian Conference on Machine Learning

Max-Margin Ratio Machine

Suicheng Gu and **Yuhong Guo**
Department of Computer and Information Sciences
Temple University, Philadelphia, PA 19122, USA

Editor: Steven C.H. Hoi and Wray Buntine

Abstract

In this paper, we investigate the problem of exploiting global information to improve the performance of SVMs on large scale classification problems. We first present a unified general framework for the existing min-max machine methods in terms of within-class dispersions and between-class dispersions. By defining a new within-class dispersion measure, we then propose a novel max-margin ratio machine (MMRM) method that can be formulated as a linear programming problem with scalability for large data sets. Kernels can be easily incorporated into our method to address non-linear classification problems. Our empirical results show that the proposed MMRM approach achieves promising results on large data sets.

1. Introduction

Support vector machines (SVMs) are one of the most popular classification methods that have been widely used in machine learning and related fields. The standard SVMs were derived from purely geometric principles (Vapnik, 1995), where one attempts to solve for a consistent linear discriminant that maximizes the minimum Euclidean distance between any data points and the decision hyperplane. Although SVMs have demonstrated good performance in the literature, it has been noticed that the margins defined in SVMs are exclusively determined locally by a small set of data points called support vectors whereas all other data points are irrelevant to the decision hyperplane. The missing consideration for global statistical information in the data set could lead to the recognition of an inferior decision hyperplane over the training data in some cases (Huang et al., 2008).

Motivated by this important observation, a new large margin approach called maximin margin machines (M^4) was developed in (Huang et al., 2004a, 2008). The M^4 model takes both local information and global information into consideration by incorporating the within-class variance into the standard SVM formulation. It builds a connection between the standard SVM and a recently proposed minimax probability machine (MPM) (Lanckriet et al., 2002). The MPM model, focusing on global statistical information, maximizes the distance between the class means and minimizes the within-class variance. One extension to the MPM, minimum error minimax probability machine (MEMPM), has been proposed in (Huang et al., 2004b) which contains an explicit performance indicator. Another extension is the structured large margin machine (SLMM) proposed in (Yeung et al., 2007) which is sensitive to the data structures. Another earlier approach that is relevant in the context of exploring both within-class and between-class measures is linear discriminant analysis

© 2012 S. Gu & Y. Guo.

(LDA) (Duda and Hart, 1973), which minimizes the within-class dispersion and maximizes the between-class dispersion.

However, these techniques developed to improve (or have the potential to improve) the standard SVMs suffer from an evident drawback of lacking scalability. They all require to estimate covariance matrices from the data, which is very sensitive to specific data samples, e.g., the outliers, especially in small sample size problems. Moreover, the MPM is solved via a second order cone program (SOCP), and the M^4 and SLMM require to solve sequential SOCP programs. For high dimensional and large scale data sets, these approaches suffer from coherent limitations regarding to computational complexities.

More recently, a relative margin machine (RMM) method was introduced to overcome the sensitivity of SVMs over large data spread directions (Shivaswamy and Jebara, 2008, 2010). The RMM maximizes the margin relative to the spread of the data by bounding the projections of training examples in an area with radius less than a threshold value B. It was shown that a proper B can help improve the prediction accuracy. However, the model is very sensitive to the value of B and the B is not easy to tune. When B is too large, the constraint will be inactive and the solution will be the same as the SVM. When B is too small, all training samples are bounded by B and the large margin issue will not be appreciated enough.

In this paper, we first show many models mentioned above, such as M^4, MPM, LDA, SVMs and RMM, can be expressed in a unified general min-max machine framework in terms of within-class dispersion and between-class dispersion measures. We then propose a novel max-margin ratio machine (MMRM) within the min-max machine framework based on a new within-class dispersion definition. This MMRM can be reformulated into an equivalent linear programming problem and it possesses scalability over large data sets. Kernels can be incorporated into the MMRM to cope with non-linear classification problems. Our empirical results show that the proposed MMRM approach can achieve higher classification accuracies on large data sets comparing to the standard SVMs and the RMM method.

2. Min-Max Machine Framework

In this section, we present a min-max machine framework to address binary classification problems. First we need to define some notations used in this section and the whole paper. We use $X = \{\mathrm{x}_i\}_{i=1}^{n_x}$ to denote the positive samples, and $Y = \{\mathrm{y}_j\}_{j=1}^{n_y}$ to denote the negative samples, where $\mathrm{x}_i,\ \mathrm{y}_j \in \Re^m$ denote the positive and negative sample vectors, n_x, n_y denote the numbers of positive and negative samples respectively, and $n = n_x + n_y$ denotes the size of the training set. We use $(\bar{\mathrm{x}}, \Sigma_x)$ and $(\bar{\mathrm{y}}, \Sigma_y)$ to denote the mean vectors and variance matrices for each class respectively, such that $\bar{\mathrm{x}} = \frac{1}{n_x}\sum_{i=1}^{n_x} \mathrm{x}_i$, $\bar{\mathrm{y}} = \frac{1}{n_y}\sum_{j=1}^{n_y} \mathrm{y}_j$, $\Sigma_x = \sum_{i=1}^{n_x}(\mathrm{x}_i - \bar{\mathrm{x}})(\mathrm{x}_i - \bar{\mathrm{x}})^\top$, and $\Sigma_y = \sum_{j=1}^{n_y}(\mathrm{y}_j - \bar{\mathrm{y}})(\mathrm{y}_j - \bar{\mathrm{y}})^\top$.

We consider the problem of training a good linear classifier. The standard SVMs identify the optimal linear classifier $f(x) = sign(\mathrm{w}^\top x + b)$ as the one that maximizes the margin between the two classes, which is equivalent to maximizing the minimum projected between-class distance for linear separable data. However, it has been noticed that SVMs sometimes miss the optimal solution due to the fact that they ignored global statistical information (Huang et al., 2008). M^4, MPM and RMM methods are then proposed to tackle this drawback of SVMs by either considering both between-class and within-class

dispersions, or considering both the margin and the bounding for all training instances. Here we define a min-max machine framework to generalize all these models.

A **min-max machine** aims to determine a hyperplane $H(\mathrm{w}; b) = \{\mathrm{x}|\mathrm{w}^\top \mathrm{x} = b\}$, where $\mathrm{w} \in \Re^m$ and $b \in \Re$, to separate two classes of data points by minimizing the within-class dispersion and maximizing the between-class dispersion. More specifically, we can define the min-max machine as a general optimization problem

$$\min_{\mathrm{w}} \frac{d_w(\mathrm{w})}{d_b(\mathrm{w})} \tag{1}$$

where $d_w(\mathrm{w})$ and $d_b(\mathrm{w})$ denote the within-class dispersion and between-class dispersion respectively. By replacing $d_w(\mathrm{w})$ and $d_b(\mathrm{w})$ with different specific dispersion measures, one can obtain a set of variants of min-max machine models. Assuming linear separable data, we will show below that LDA, MPM, SVM, M^4 and RMM can be formulated within this min-max machine framework.

2.1. Linear Discriminant Analysis (LDA)

LDA is a well known discriminative method with an optimization goal of minimizing the within-class dispersion and maximizing the between-class dispersion, which is consistent with the min-max machine framework we proposed. LDA optimization is typically defined as $\max_{\mathrm{w}} \frac{\mathrm{w}^\top \Sigma_b \mathrm{w}}{\mathrm{w}^\top \Sigma_w \mathrm{w}}$, where $\Sigma_b = (\bar{\mathrm{x}} - \bar{\mathrm{y}})(\bar{\mathrm{x}} - \bar{\mathrm{y}})^\top$ and $\Sigma_w = \Sigma_x + \Sigma_y$. Thus the within-class dispersion for LDA is defined as the squared root of the sum of the projected within-class variances of the two classes

$$d_w^v(\mathrm{w}) = \sqrt{\mathrm{w}^\top \Sigma_x \mathrm{w} + \mathrm{w}^\top \Sigma_y \mathrm{w}} \tag{2}$$

and the between-class dispersion of LDA is defined as the projected distance between the two mean (center) vectors of the two classes

$$d_b^c(\mathrm{w}) = \sqrt{\mathrm{w}^\top (\bar{\mathrm{x}} - \bar{\mathrm{y}})(\bar{\mathrm{x}} - \bar{\mathrm{y}})^\top \mathrm{w}} = \mathrm{w}^\top (\bar{\mathrm{x}} - \bar{\mathrm{y}}) \tag{3}$$

2.2. Minimax Probability Machine (MPM)

The optimization problem for MPM is formulated as

$$\min_{\mathrm{w}} \sqrt{\mathrm{w}^\top \Sigma_x \mathrm{w}} + \sqrt{\mathrm{w}^\top \Sigma_y \mathrm{w}} \quad \text{s.t. } \mathrm{w}^\top (\bar{\mathrm{x}} - \bar{\mathrm{y}}) = 1 \tag{4}$$

which is solved using a second order cone program (SOCP) (Lanckriet et al., 2002). One can easily verify that this problem is equivalent to a min-max machine problem with the between-class dispersion $d_b^c(\mathrm{w})$ defined in (3) and the within-class dispersion defined as the sum of the standard deviation of the two classes

$$d_w^s(\mathrm{w}) = d_x^s(\mathrm{w}) + d_y^s(\mathrm{w}) = \sqrt{\mathrm{w}^\top \Sigma_x \mathrm{w}} + \sqrt{\mathrm{w}^\top \Sigma_y \mathrm{w}} \tag{5}$$

2.3. Support Vector Machines (SVMs)

The hyperplane of the standard SVM can be viewed as being determined by minimizing a simple within-class dispersion in terms of the norm of w (note it is obvious that a smaller norm of w will result in a smaller within-class dispersion.)

$$d_w^n = \left(\frac{1}{2}\mathrm{w}^\top \mathrm{w}\right)^{\frac{1}{2}} \tag{6}$$

and maximizing the between-class margin distance defined between the samples closest to the margin (the support vectors)

$$d_b^m(\mathrm{w}) = \mathrm{w}^\top(\mathrm{x}^* - \mathrm{y}^*), \tag{7}$$

where x^* and y^* are the margin samples in class X and Y respectively such that

$$\mathrm{x}^* = \arg\min_{\mathrm{x}_i} \mathrm{w}^\top \mathrm{x}_i \tag{8}$$

$$\mathrm{y}^* = \arg\max_{\mathrm{y}_j} \mathrm{w}^\top \mathrm{y}_j \tag{9}$$

Applying the within-class dispersion and the between-class dispersion defined in (6) and (7), the min-max machine problem in (1) would result to the standard linear separable SVM

$$\min_{\mathrm{w,b}} \frac{1}{2}\mathrm{w}^\top \mathrm{w} \qquad \text{subject to} \;\; \mathrm{w}^\top \mathrm{x}_i + b \geq 1, \quad \forall i; \quad -(\mathrm{w}^\top \mathrm{y}_j + b) \geq 1, \quad \forall j. \tag{10}$$

where the bias term is

$$b = b_1 = -\frac{\mathrm{w}^\top \mathrm{x}^* + \mathrm{w}^\top \mathrm{y}^*}{2}. \tag{11}$$

2.4. Maxi-Min Margin Machine (M^4)

The M^4 is proposed in an attempt to integrate SVM and MPM by considering both discriminative margin information and the global statistical information. It is formulated as an optimization problem below

$$\begin{aligned} \max_{\rho,\mathrm{w},b} \quad & \rho \\ \text{subject to} \quad & \mathrm{w}^\top \mathrm{x} + b \geq \rho\sqrt{\mathrm{w}^\top \Sigma_x \mathrm{w}}, \quad \forall i; \\ & -(\mathrm{w}^\top \mathrm{y} + b) \geq \rho\sqrt{\mathrm{w}^\top \Sigma_y \mathrm{w}}, \; \forall j \end{aligned} \tag{12}$$

which can be put into the framework of min-max machine by using the within-class dispersion, $d_w^s(\mathrm{w})$, defined in (5) and the between-class margin distance, $d_b^m(\mathrm{w})$, defined in (7). The bias term b for M^4 is computed via

$$b = b_2 = -\mathrm{w}^\top \mathrm{y}^* - \frac{d_y^s(\mathrm{w})}{d_w^s(\mathrm{w})}(\mathrm{w}^\top \mathrm{x}^* - \mathrm{w}^\top \mathrm{y}^*).$$

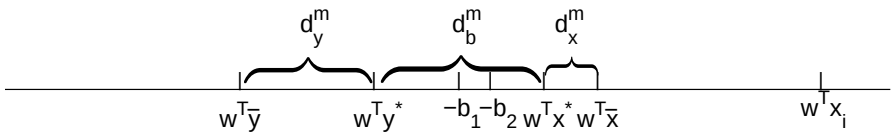

Figure 1: The mapping of various samples in the w space. $\bar{\mathrm{x}}, \bar{\mathrm{y}}$: mean of the positive and negative classes, $\mathrm{x}^*, \mathrm{y}^*$: margin samples (support vectors), b_1, b_2: bias. x_i: an outlier sample.

2.5. Relative Margin Machines (RMM)

The RMM maximizes the classification margin relative to the spread of the data. It is formulated as

$$\begin{aligned} \min_{\mathrm{w,b}} \quad & \frac{1}{2}\mathrm{w}^\top\mathrm{w} \\ \text{subject to} \quad & \mathrm{w}^\top\mathrm{x}_i + b \geq 1, \quad \mathrm{w}^\top\mathrm{x}_i + b \leq B, \quad \forall i; \\ & -(\mathrm{w}^\top\mathrm{y}_j + b) \geq 1, \quad -(\mathrm{w}^\top\mathrm{y}_j + b) \leq B, \quad \forall j. \end{aligned} \tag{13}$$

It is easy to verify that the above problem is equivalent to a min-max machine problem where the within-class dispersion is defined as the regularized projected diameter of the whole data set

$$d_w^d(\mathrm{w}) = \max_{i,j} \left(\mathrm{w}^\top\mathrm{x}_i - \mathrm{w}^\top\mathrm{y}_j\right) + \frac{D}{2}\mathrm{w}^\top\mathrm{w} \tag{14}$$

and the between-class margin distance, $d_b^m(\mathrm{w})$, is defined in (7). The D in (14) is a trade-off parameter.

The techniques we presented above suggest that a combination of different within-class dispersions and between-class dispersions (projected to the w space) within the min-max machine framework can lead to models with strength in different perspectives. Figure 1 provides an intuitive understanding about outlier samples, class mean vectors, marginal samples (support vectors) and the bias terms in the projected space.

3. Max-Margin Ratio Machine

Except support vector machines and relative margin machines, all the other three methods we reviewed within the min-max machine framework above require the estimation of covariance matrices, which could make the computation steps in these techniques inefficient for high dimensional or large data sets. Moreover, covariance matrices are typically not very robust to outlier samples. Due to these observations, in this section we propose a new within-class margin dispersion in the min-max machine framework, which leads to a novel classification method, *max-margin ratio machine*.

3.1. Max-Margin Ratio Machine Model

Consider the within-class dispersion $d_w^s(\mathrm{w})$ defined in (5) for MPM. It can be rewritten into the follow by expanding the covariance terms

$$d_w^s(\mathrm{w}) = d_x^s(\mathrm{w}) + d_y^s(\mathrm{w}) = \Big(\sum_{i=1}^{n_x}(\mathrm{w}^\top \mathrm{x}_i - \mathrm{w}^\top \bar{\mathrm{x}})^2\Big)^{\frac{1}{2}} + \Big(\sum_{j=1}^{n_y}(\mathrm{w}^\top \mathrm{y}_j - \mathrm{w}^\top \bar{\mathrm{y}})^2\Big)^{\frac{1}{2}} \tag{15}$$

Now we replace the Frobenius norm in (15) with $L^\infty-$norm, then a new within-class disperse measure can be obtained as below

$$\begin{aligned}&\lim_{d\to\infty} \Big(\sum_i |\mathrm{w}^\top \mathrm{x}_i - \mathrm{w}^\top \bar{\mathrm{x}}|^d\Big)^{\frac{1}{d}} + \Big(\sum_j |\mathrm{w}^\top \mathrm{y}_j - \mathrm{w}^\top \bar{\mathrm{y}}|^d\Big)^{\frac{1}{d}} \\ &= \max_i \ |\mathrm{w}^\top \bar{\mathrm{x}} - \mathrm{w}^\top \mathrm{x}_i| + \max_j \ |\mathrm{w}^\top \mathrm{y}_j - \mathrm{w}^\top \bar{\mathrm{y}}|\end{aligned}$$

where covariance terms are avoided. For linear separable data, samples can only appear on the correct side of the separation hyperplane, and the outliers will be far away from the separation hyperplane. Thus to reduce the influence of possible outliers, we can consider only samples x_i and y_j that satisfy $\mathrm{w}^\top \bar{\mathrm{x}} - \mathrm{w}^\top \mathrm{x}_i \geq 0$ and $\mathrm{w}^\top \mathrm{y}_j - \mathrm{w}^\top \bar{\mathrm{y}} \geq 0$. From Figure 1, we can see these samples are the ones that lie between each class center and the separation hyperplane. With this additional constraint, the absolute value sign in (16) can be dropped, and a novel within-class dispersion measure can be obtained

$$d_w^m(\mathrm{w}) = \max_i \ (\mathrm{w}^\top \bar{\mathrm{x}} - \mathrm{w}^\top \mathrm{x}_i) + \max_j \ (\mathrm{w}^\top \mathrm{y}_j - \mathrm{w}^\top \bar{\mathrm{y}}) = (\mathrm{w}^\top \bar{\mathrm{x}} - \mathrm{w}^\top \mathrm{x}^*) + (\mathrm{w}^\top \mathrm{y}^* - \mathrm{w}^\top \bar{\mathrm{y}}) \tag{16}$$

which is the sum of the distances between the margin samples, x^* and y^*, and the class mean vectors.

This new within-class dispersion does not require data covariance terms and hopefully the computational cost can be reduced. The outliers (see x_i in Figure 1) that are far away from the hyperplane, counted only in the estimation of the mean vectors, have very small influence on this dispersion measure. This new within-class dispersion measure also builds a connection between the between-class center distance in (3) and the between-class margin distances in (7), such that

$$d_w^m(\mathrm{w}) = d_b^c(\mathrm{w}) - d_b^m(\mathrm{w}). \tag{17}$$

Combining this new within-class dispersion (16) and the between-class margin distance $d_b^m(\mathrm{w})$ of (7) used in SVMs in a similar way as the M^4 within the min-max machine framework, we can obtain the following optimization problem

$$\max_{\rho,\mathrm{w},b} \quad \rho \tag{18}$$

$$\begin{aligned}\text{subject to} \quad & \mathrm{w}^\top \mathrm{x}_i + b \geq \rho \mathrm{w}^\top(\bar{\mathrm{x}} - \mathrm{x}_i), \quad \forall i; \\ & -(\mathrm{w}^\top \mathrm{y}_j + b) \geq \rho \mathrm{w}^\top(\mathrm{y}_j - \bar{\mathrm{y}}), \ \forall j\end{aligned} \tag{19}$$

where the ρ represents the ratio between the between-class dispersion $d_b^m(\mathrm{w})$, and the within-class dispersion $d_w^m(\mathrm{w})$, which is stated in the following theorem.

Theorem 1 *For linear separable data, the following equation stands between the optimal solution* (ρ^*, w^*) *for the optimization problem (18), the within-class dispersion* $d_w^m(w^*)$ *and the between-class margin distance* $d_b^m(w^*)$

$$\rho^* = \frac{d_b^m(w^*)}{d_w^m(w^*)} \tag{20}$$

Proof: According to the constraints of (18), when w is fixed, the samples $\{\mathrm{x}_i\}$ and $\{\mathrm{y}_j\}$ have no effect on ρ if $\mathrm{w}^\top(\bar{\mathrm{x}} - \mathrm{x}_i) \leq 0$ or $\mathrm{w}^\top(\mathrm{y}_j - \bar{\mathrm{y}}) \leq 0$. The optimal ρ corresponding to w is

$$\begin{aligned}
\rho &= \max_b \min_{i,j} \left\{ \frac{\mathrm{w}^\top \mathrm{x}_i + b}{\mathrm{w}^\top(\bar{\mathrm{x}} - \mathrm{x}_i)}, \frac{-(\mathrm{w}^\top \mathrm{y}_j + b)}{\mathrm{w}^\top(\mathrm{y}_j - \bar{\mathrm{y}})} \,\middle|\, \mathrm{w}^\top(\bar{\mathrm{x}} - \mathrm{x}_i) > 0, \mathrm{w}^\top(\mathrm{y}_j - \bar{\mathrm{y}}) > 0 \right\} \\
&= \max_b \min \left\{ \frac{\mathrm{w}^\top \mathrm{x}^* + b}{\mathrm{w}^\top(\bar{\mathrm{x}} - \mathrm{x}^*)}, \frac{-(\mathrm{w}^\top \mathrm{y}^* + b)}{\mathrm{w}^\top(\mathrm{y}^* - \bar{\mathrm{y}})} \right\} \\
&\leq \max_b \frac{\mathrm{w}^\top \mathrm{x}^* + b - (\mathrm{w}^\top \mathrm{y}^* + b)}{\mathrm{w}^\top(\bar{\mathrm{x}} - \mathrm{x}^*) + \mathrm{w}^\top(\mathrm{y}^* - \bar{\mathrm{y}})} = \frac{d_b^m(\mathrm{w})}{d_w^m(\mathrm{w})}
\end{aligned} \tag{21}$$

Note that, we use the fact that if $b, d > 0$, then $\min(\frac{a}{b}, \frac{c}{d}) \leq \frac{a+c}{b+d}$. The equality holds if and only if

$$\frac{\mathrm{w}^\top \mathrm{x}^* + b}{\mathrm{w}^\top(\bar{\mathrm{x}} - \mathrm{x}^*)} = \frac{-(\mathrm{w}^\top \mathrm{y}^* + b)}{\mathrm{w}^\top(\mathrm{y}^* - \bar{\mathrm{y}})}$$

which implies

$$b = -\mathrm{w}^\top \mathrm{y}^* - \frac{\mathrm{w}^\top(\mathrm{y}^* - \bar{\mathrm{y}})}{\mathrm{w}^\top(\bar{\mathrm{x}} - \mathrm{x}^*) + \mathrm{w}^\top(\mathrm{y}^* - \bar{\mathrm{y}})}(\mathrm{w}^\top \mathrm{x}^* - \mathrm{w}^\top \mathrm{y}^*). \tag{22}$$

□ According to this interpretation, we therefore name the model we proposed in (18) above as *max-margin ratio machine*. Note the optimization problem in (18) is not convex due to the existence of the bilinear term ρw in the constraints. Although sequential quadratic programming can be developed to solve this problem, it will not be an efficient solution.

3.2. An Equivalent Alternative Formulation

In order to obtain a simple optimization problem, we next try to integrate the proposed within-class dispersion $d_w^m(\mathrm{w})$ in (16) with the much simpler between-class dispersion $d_b^c(\mathrm{w})$ defined in (3). Within the min-max machine framework, this optimization problem can be literally formulated as

$$\begin{aligned}
\max_{\mathrm{w}} \quad & \mathrm{w}^\top(\bar{\mathrm{x}} - \bar{\mathrm{y}}) \\
\text{subject to} \quad & (\mathrm{w}^\top \bar{\mathrm{x}} - \mathrm{w}^\top \mathrm{x}_i) + (\mathrm{w}^\top \mathrm{y}_j - \mathrm{w}^\top \bar{\mathrm{y}}) \leq 1 \quad \forall i, \forall j.
\end{aligned} \tag{23}$$

More interestingly, we will show below that the two optimization problems (18) and (23) we formulated above yield the same optimal solution.

Lemma 2 *Assuming linear separable training data, the optimization problems* (18) *and* (23) *have the same optimal solution with respect to w.*

Proof: We know $d_w^m(\mathrm{w}) = d_b^c(\mathrm{w}) - d_b^m(\mathrm{w})$ (see (17)). If w^* is the optimal solution of (18), then

$$\rho(\mathrm{w}^*) = \frac{d_b^m(\mathrm{w}^*)}{d_w^m(\mathrm{w}^*)} = \max_{\mathrm{w}} \frac{d_b^m(\mathrm{w})}{d_w^m(\mathrm{w})}.$$

The optimization problem (23) is

$$\max_{\mathrm{w}} \frac{d_b^c(\mathrm{w})}{d_w^m(\mathrm{w})} = \max_{\mathrm{w}} \frac{d_b^m(\mathrm{w}) + d_w^m(\mathrm{w})}{d_w^m(\mathrm{w})} = \max_{\mathrm{w}} \frac{d_b^m(\mathrm{w})}{d_w^m(\mathrm{w})} + 1 = \rho(\mathrm{w}^*) + 1.$$

Therefore, w^* is also the optimal solution of (23). □

Lemma 2 suggests the optimization problems (18) and (23) are equivalent for linear separable data. Below we will further exploit the relationship between $d_w^m(\mathrm{w})$, $d_b^c(\mathrm{w})$ and $d_b^m(\mathrm{w})$ to construct an even simpler formulation. We consider the following equations

$$\arg\max_{\mathrm{w}} \frac{d_b^m(\mathrm{w})}{d_w^m(\mathrm{w})} = \arg\min_{\mathrm{w}} \frac{d_w^m(\mathrm{w})}{d_b^m(\mathrm{w})} + 1 = \arg\min_{\mathrm{w}} \frac{d_w^m(\mathrm{w}) + d_b^m(\mathrm{w})}{d_b^m(\mathrm{w})} = \arg\min_{\mathrm{w}} \frac{d_b^c(\mathrm{w})}{d_b^m(\mathrm{w})}.$$

This suggests a simple linear programming optimization problem as below

$$\min_{\mathrm{w},b} \ \mathrm{w}^\top(\bar{\mathrm{x}} - \bar{\mathrm{y}}) \quad \text{subject to} \quad \mathrm{w}^\top \mathrm{x}_i + b \geq 1, \quad \forall i; \quad -(\mathrm{w}^\top \mathrm{y}_j + b) \geq 1, \ \forall j. \tag{24}$$

Theorem 3 *If w is the optimal solution of* (24)*, then w is the optimal solution of* (18)*, and the optimal margin ration of* (18) *is* $\rho = \frac{2}{w^\top(\bar{x}-\bar{y})-2}$.

Proof: Assume w is an optimal solution of (24). Let x^* and y^* be margin samples of the two classes along the direction w, as defined in (8) and (9). Then it is straightforward to have $\mathrm{w}^T(\mathrm{x}^* - \mathrm{y}^*) = 2$. Let b_2 be defined as in (22), then $(\mathrm{w}, \rho = \frac{2}{\mathrm{w}^T(\bar{\mathrm{x}}-\bar{\mathrm{y}})-2}, b_2)$ is a feasible solution for (18) according to the proof in Theorem 1. If w is not an optimal solution for (18), there will exist another feasible solution for (18), (w_1, ρ_1), satisfying $\rho_1 > \frac{2}{\mathrm{w}^T(\bar{\mathrm{x}}-\bar{\mathrm{y}})-2}$. Let

$$\mathrm{x}^1 = \arg\min_{x_i} \mathrm{w}_1^T \mathrm{x}_i \tag{25}$$

$$\mathrm{y}^1 = \arg\max_{y_j} \mathrm{w}_1^T \mathrm{y}_j \tag{26}$$

Assume $\mathrm{w}_1^T(\mathrm{x}^1 - \mathrm{y}^1) = \frac{2}{s}$. Let $\mathrm{w}_2 = s\mathrm{w}_1$, then (w_2, ρ_1) is a feasible solution for (18) as well. Moreover, $\mathrm{w}_2^T(\mathrm{x}^1 - \mathrm{y}^1) = 2$, and w_2 is a feasible solution for (24) since there will be a b value such that

$$\mathrm{w}_2^T \mathrm{x}_i + b \geq \mathrm{w}_2^T \mathrm{x}_1 + b = 1, \forall i; \qquad -(\mathrm{w}_2^T \mathrm{y}_j + b) \geq -(\mathrm{w}_2^T \mathrm{y}_1 + b) = 1, \forall j.$$

Thus we have

$$\mathrm{w}_2^T(\bar{\mathrm{x}} - \bar{\mathrm{y}}) = \frac{2}{\rho_1} + 2 < \frac{2}{\rho} + 2 = \mathrm{w}^T(\bar{\mathrm{x}} - \bar{\mathrm{y}}) \tag{27}$$

This inequality means w is not an optimal solution for (24), which contradicts the assumption.□

The theorem above suggests that one can solve the max-margin ratio machine problem (18) and the problem (23) by solving a simple linear programming problem (24). Below we show that the convex linear programming (24) is a bounded optimization problem.

Lemma 4 *If w satisfies the constraints in (24), then $w^\top(\bar{x}-\bar{y}) \geq 2$.*

Proof: It is straightforward to get the conclusion by summing all the constraint inequalities in (24). □

3.3. Slack Variables for Non-separable Data

All derivations we conducted above are for linear separable data. For non-separable problems, we can add slack variables to cope with misclassification errors, which yields a soft margin model as follow

$$\begin{aligned} \min_{\mathrm{w},b,\xi} \quad & \mathrm{w}^\top(\bar{\mathrm{x}}-\bar{\mathrm{y}}) + C\Big(\sum_{i=1}^{n_x}\xi_i + \sum_{j=1}^{n_y}\eta_j\Big) \\ \text{subject to} \quad & \mathrm{w}^\top \mathrm{x}_i + b \geq 1-\xi_i,\ \xi_i \geq 0, \quad \forall i; \\ & -(\mathrm{w}^\top \mathrm{y}_j + b) \geq 1-\eta_j,\ \eta_j \geq 0,\ \forall j. \end{aligned} \tag{28}$$

From now on, we will refer to this model as max-margin ratio machine (MMRM). This is, again, a standard linear programming problem, which can be easily solved. Note that, we can let $C \geq \max(\frac{1}{n_x}, \frac{1}{n_y})$ to ensure the problem have bounded solution (according to the proof of Lemma 4).

3.4. Kernelization of MMRM

We have focused on addressing linear classification problems so far. However, it has been well known that many linear non-separable problems are actually separable in nonlinear high dimensional space. In order to deal with nonlinear classification problems, we exploit the standard kernelization technique to kernelize MMRM. This is done by introducing a nonlinear mapping function $\varphi : \Re^m \to \Re^f$ to map the original samples into a high dimensional feature space R^f, and rewriting the w parameter as

$$\mathrm{w} = \sum_{k=1}^{n_x}\alpha_k\varphi(\mathrm{x}_k) - \sum_{l=1}^{n_y}\beta_l\varphi(\mathrm{y}_l) \tag{29}$$

for nonnegative parameters $\{\alpha_k\}$ and $\{\beta_l\}$. The kernelized MMRM can then be obtained by replacing the w in (28) with (29) and using a Kernel function $K(\cdot,\cdot)$ to replace the inner

product of two high dimensional vectors, $\varphi(\cdot)^\top\varphi(\cdot)$,

$$\begin{aligned}\min_{\alpha,\beta,b,\xi}\quad & \sum_{k=1}^{n_x}\alpha_k\Big(\frac{1}{n_x}\sum_{i=1}^{n_x}K(x_k,x_i)-\frac{1}{n_y}\sum_{j=1}^{n_y}K(x_k,y_j)\Big) \\ & -\sum_{l=1}^{n_y}\beta_l\Big(\frac{1}{n_x}\sum_{i=1}^{n_x}K(y_l,x_i)-\frac{1}{n_y}\sum_{j=1}^{n_y}K(y_l,y_j)\Big)+C\Big(\sum_{i=1}^{n_x}\xi_i+\sum_{j=1}^{n_y}\eta_j\Big) \\ \text{subject to}\quad & \sum_{k=1}^{n_x}\alpha_k K(x_k,x_i)-\sum_{l=1}^{n_y}\beta_l K(y_l,x_i)+b\geq 1-\xi_i,\forall i; \\ & \sum_{l=1}^{n_y}\beta_l K(y_l,y_j)-\sum_{k=1}^{n_x}\alpha_k K(x_k,y_j)-b\geq 1-\xi_j,\forall j; \\ & \alpha_k\geq 0,\ \beta_l\geq 0,\quad \forall k,l;\qquad \xi_i\geq 0,\ \eta_j\geq 0,\quad \forall i,j.\end{aligned} \tag{30}$$

Many semidefinite kernel functions can be used here. Each different kernel function corresponds to a different feature mapping function. In the experiments of this paper, we in particular used RBF kernels.

By solving the above linear programming problem,the optimal nonlinear hyperplane (w, b) can be obtained implicitly. Given a new sample z, it can be classified by the following function

$$f(\mathrm{z})=\mathrm{w}^T\mathrm{z}+b=\sum_{k=1}^{n_x}\alpha_k K(\mathrm{z},\mathrm{x}_k)-\sum_{l=1}^{n_y}\beta_l K(\mathrm{z},\mathrm{y}_l)+b \tag{31}$$

Similarly as in SVMs, when $\alpha_k>0$ and $\beta_l>0$, the corresponding x_k and y_l are support vectors. The α and β are usually sparse. We only need to compute kernel values between the few support vectors and the test sample for the prediction.

4. Experiments

In this section, we report the experimental results on both binary data sets and multi-class data sets. The MMRM is compared with four other min-max approaches: SVM, RMM, MPM and M^4. Sedumi (Sturm, 1999) is employed to train MMRM, M^4 and MPM, and the libsvm (Fan et al., 2005) is employed to train SVM. We used the RMM code downloaded from Internet.[1] The RBF kernels used in the experiments are defined as $K(\mathrm{x},\mathrm{z})=\exp(-g\|\mathrm{x}-\mathrm{z}\|^2)$.

4.1. Results on UCI Data Sets

We first conducted experiments on three data sets, *Ionosphere*, *Pima* and *Sonar*. Each data set was randomly partitioned into 90% training and 10% test sets. We tested both linear classification models and kernelized classification models with RBF kernels. The parameter g for RBF kernel, the parameter B in RMM, and the trade-off parameter C in each comparison approach were tuned using cross validation. Classification accuracies and

1. http://www.cs.columbia.edu/~pks2103/RMM/

Table 1: Comparisons of classification accuracies (%) and standard deviations among MMRM, SVM, RMM, M^4 and MPM.

	Linear kernel				
data	MMRM	SVM	RMM	M^4	MPM
Ionosphere	**87.8(0.3)**	86.7(0.3)	85.2(0.3)	87.7(0.4)	85.2(0.4)
Pima	76.6(0.4)	**77.9(0.4)**	76.1(0.5)	77.7(0.4)	77.5(0.4)
Sonar	**76.0(1.2)**	72.4(1.2)	74.3(1.2)	74.6(1.2)	71.6(1.2)
	RBF kernel				
data	MMRM	SVM	RMM	M^4	MPM
Ionosphere	**94.4(0.3)**	94.0(0.2)	94.3(0.3)	94.2(0.3)	92.3(0.3)
Pima	76.9(0.4)	**78.0(0.5)**	76.2(0.5)	77.6(0.4)	76.2(0.5)
Sonar	**88.1(0.6)**	86.5(0.7)	86.9(0.7)	87.3(0.6)	84.9(0.7)

Table 2: Comparisons of classification errors of MMRM, RMM and SVM with RBF kernels.

	Data Info				Errors		
data sets	#train	#test	#features	#classes	MMRM	RMM	SVM
Isolet	6238	1559	617	26	**47**	50	52
Usps	7291	2007	256	10	**89**	96	98
Letters	16000	4000	16	26	**80**	88	88
Mnist	60000	10000	784	10	**124**	129	131

their standard deviations are reported in Table 1. The reported results are the averages over 50 random partitions for linear kernels and RBF kernels respectively. The proposed MMRM obtained highest accuracies on two of the three data sets. Moreover, the results suggest that kernelization can improve the classification performance of these approaches on non-separable data sets.

4.2. Results on Large Data Sets

Although our empirical results on regular UCI data sets are promising, we are more interested in investigating the performance of the proposed MMRM on large scale data sets, where previous extensions of SVMs, such as MPM and M^4, are computationally expensive. We then conducted experiments on four large scale data sets: *Letters, Isolet, Usps* and *Mnist*. Information about these data sets is given in Table 2. For Isolet, Usps and Letters, we used the original data without further preprocessing. Principal components analysis (PCA) was used to reduce the dimensionality of Mnist from 784 down to 80 to speed up training.

We compared the proposed MMRM approach with standard SVMs and RMMs on these four data sets while the MPM and M^4 are not tested due to their high computational complexity. Large data sets are often linear non-separable, hence we used RBF kernels. Moreover, these four data sets have multiple classes, and we thus need to address multi-class classification problems. In this study, we used the "one-against-one" strategy and trained $M(M-1)/2$ binary classifiers for each M-class problem. On each data set, we

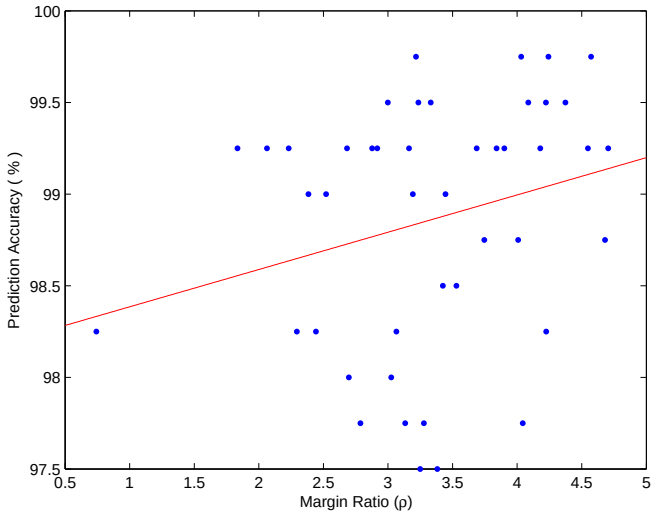

Figure 2: Relationship between prediction accuracy and margin ratio on USPS data set.

selected parameters g and C using SVMs via a 5-fold cross validation, then used the same g and C for both RMM and MMRM. The additional parameter B in RMM was also selected by cross validation.

Test errors of the three methods are reported in Table 2. The RMM outperformed the SVM in most cases. However, the price of this improvement is the addition of an extra parameter which is difficult to select and limits the generalization of the model. The proposed MMRM obtained the lower test errors on all four data sets than both SVMs and RMMs without adding any additional parameters. In most cases, the prediction errors yielded by MMRMs are 10% lower than that by SVMs. The MMRM is a linear programming problem while the SVM is a quadratic programming problem.

4.3. Margin Ratio vs. Prediction Accuracy

We also conducted experiments to investigate the relationship between the margin ratio (ρ) achieved by the MMRM and the prediction accuracy it produced. We used the Usps data set. We trained 45 MMRM models, where each model is trained for a pair of two classes. We plot the margin ratio on training set and the corresponding prediction accuracy on testing set of each pair in Figure 2. It suggests that the margin ratio of the model on training data is positively correlated with the prediction accuracy on testing data. We used a linear regression to model this correlation relationship: $\textit{prediction accuracy} = 0.20354\rho + 98.182$, which implies the prediction accuracy increases linearly as the margin ratio increases. The regression function is shown as a red line in Figure 2. This verifies our assumption that maximizing margin ratio can lead to a classification model with good generalization performance on testing data.

5. Conclusion

In this paper, a unified general framework for the existing min-max machine methods was presented in terms of within-class dispersions and between-class dispersions. Then a new within-class dispersion measure was introduced which leads to a novel max-margin ratio machine (MMRM) method. The MMRM can be formulated as a linear programming problem and has scalability for large data sets. Kernel techniques were used to derive the non-linear version of MMRM to cope with non-linear classification problems. The empirical results show that the proposed MMRM approach can achieve higher classification accuracies on large data sets comparing to the standard SVMs and relative margin machines (RMMs).

References

R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

R. Fan, P. Chen, and C. Lin. Working set selection using second order information for training support vector machines. *JMLR*, 6:1889–1918, 2005.

K. Huang, H. Yang, I. King, and M. Lyu. Learning large margin machines locally and globally. In *Proc. of ICML*, 2004a.

K. Huang, H. Yang, I. King, M. Lyu, and L. Chan. Minimum error minimax probability machine. *JMLR*, 5:1253–1286, 2004b.

K. Huang, H. Yang, I. King, and M. Lyu. Maxi-min margin machine: Learning large margin classifiers locally and globally. *IEEE Tran. on Neural Networks*, 19(2):260–272, 2008.

G. Lanckriet, L. Ghaoui, C. Bhattacharyya, and M. Jordan. A robust minimax approach to classification. *JMLR*, 3:555–582, 2002.

P. Shivaswamy and T. Jebara. Relative margin machines. In *Proc. of NIPS*, 2008.

P. Shivaswamy and T. Jebara. Maximum relative margin and data-dependent regularization. *JMLR*, 11:747–788, 2010.

J. Sturm. Using SeDuMi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, 11:625–653, 1999.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag New York, 1995.

D. Yeung, D. Wang, W. Ng, E. Tsang, and X. Wang. Structured large margin machines: sensitive to data distributions. *Machine Learning*, 68:171–200, 2007.