Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

A Maximum K-Min Approach for Classification

Mingzhi Dong[†], Liang Yin[†], Weihong Deng[†], Li Shang[‡], Jun Guo[†], Honggang Zhang[†]
[†]Beijing University of Posts and Telecommunications
[‡]Intel Labs China
mingzhidong@gmail.com, {yin,whdeng}@bupt.edu.cn, li.shang@intel.com, {guojun,zhhg}@bupt.edu.cn

Abstract

In this paper, a general Maximum K-Min approach for classification is proposed. With the physical meaning of optimizing the classification confidence of the K worst instances, Maximum K-Min Gain/Minimum K-Max Loss (MKM) criterion is introduced. To make the original optimization problem with combinational number of constraints computationally tractable, the optimization techniques are adopted and a general compact representation lemma for MKM Criterion is summarized. Based on the lemma, a Nonlinear Maximum K-Min (NMKM) classifier and a Semi-supervised Maximum K-Min (SMKM) classifier are presented for traditional classification task and semi-supervised classification task respectively. Based on the experiment results of publicly available datasets, our Maximum K-Min methods have achieved competitive performance when comparing against Hinge Loss classifiers.

Introduction

In the realm of classification, maximin approach, which pays strong attention to the worst situation, is widely adopted and it is regarded as one of the most elegant ideas. Hard-margin Support Vector Machine (SVM) (Vapnik 2000; Cristianini and Shawe-Taylor 2000) is the most renowned maximin classifier and it enjoys the intuition of margin maximization. Nevertheless, maximin methods based on the worst instance may be sensitive to noisy points/outliers near the boundary, as shown in Figure 1(a). Therefore, slack variables are introduced and Soft-margin SVM is proposed (Vapnik 2000; Cristianini and Shawe-Taylor 2000). By tuning the hyper-parameter/hyperparameters, a balance between the margin and the Hinge Loss can be obtained. Satisfied classification performance has been reported in a large number of applications and numerous modified algorithms have been proposed for specified tasks, such as S3VM for semi-supervised classification (Bennett, Demiriz, and others 1999), MI-SVM for multi-instance classification (Andrews, Tsochantaridis, and Hofmann 2002) and so on. But during the training process, the methods based on Hinge Loss can not control the number of worst instances to be considered exactly, in many applications, we may prefer to set a parameter K and focuse on maximizing the gain obtained by the K worst-classified instances while ignoring the remaining ones, as exemplified in Figure 1(c)(d).

Therefore, after the previous work on a special case of naive linear classifier (Dong et al. 2012), in this paper, we propose a general Maximum K-Min approach for classification. With the physical meaning of optimizing the classification confidence of the K worst instances, Maximum K-Min Gain/Minimum K-Max Loss (MKM) criterion is firstly introduced. Then, the general compact representation lemma is summarized, which makes the original optimization problem with combinational number of constraints computationally tractable. To verify the performance of Maximum K-Min criterion, a Nonlinear Maximum K-Min (NMKM) classifier and a Semi-supervised Maximum K-Min (SMKM) classifier are presented for traditional classification task and semi-supervised classification task respectively. Based on the experiment results of publicly available datasets, our Maximum K-Min methods have achieved competitive performance when comparing against Hinge Loss classifiers.

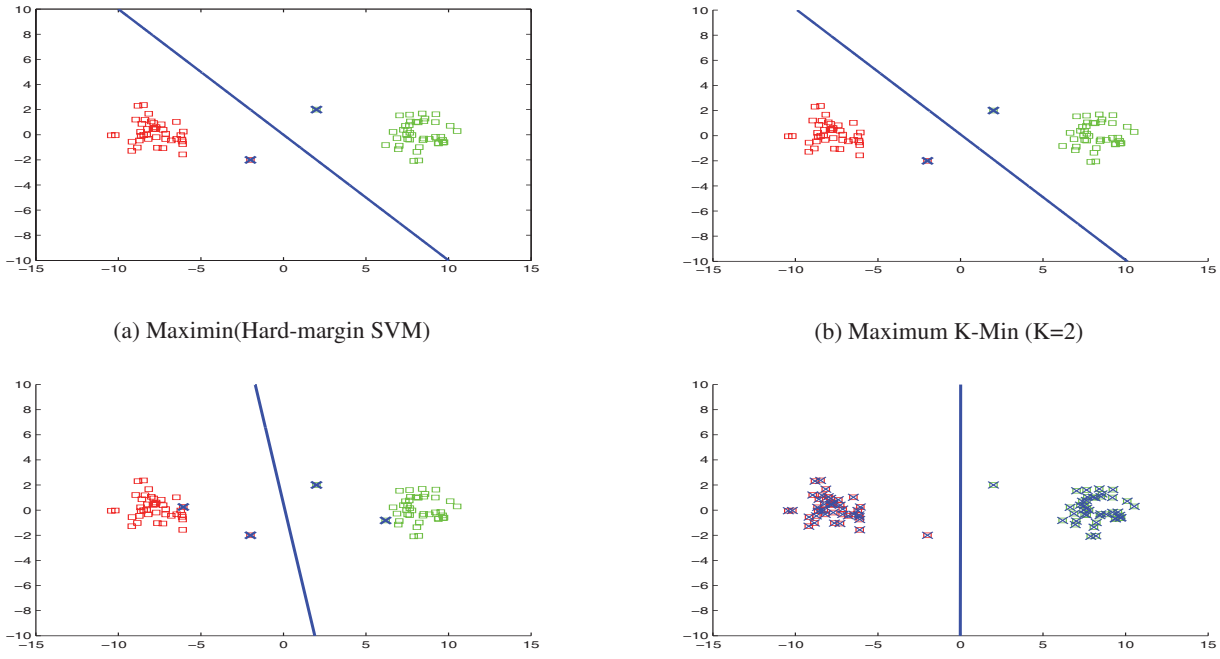
In summary, the contributions of this paper are listed as follows

- MKM criterion is introduced, which can control the parameter of K directly and servers as an alternative of Hinge Loss;
- The general compact representation lemma of MKM criterion is summarized;
- NMKM and SMKM are presented for traditional classification and semi-supervised classification respectively;
- The performance of MKM criterion is verified via experiments.

This paper is organized as follows. Section 2 discusses SVM in the view of maximin. Then Section 3 describes the MKM criterion and the tractable representation. Section 4 proposes two MKM classifiers of NMKM and SMKM. Section 5 shows the experiment results and Section 6 draws the conclusion.

SVM in the View of Maximin

Firstly, we will discuss the maximin formula of SVM in this section.



(a) Maximin(Hard-margin SVM)

(b) Maximum K-Min (K=2)

(c) Maximum K-Min (K=4)

(d) Maximum K-Min (K=N)

Figure 1: A comparison of Maximin Approach (Hard-Margin SVM) and linear Maximum K-Min Approach(Support Vectors/Worst K Instances are marked $\times$). When selected as the support vectors, the outliers near the decision boundary have much influence in Maximin approach, as shown in (a). In contrast, with proper chosen K, Maximum K-Min approach will be more robust, as shown in (c)(d).

Maximin Formula of Hard-Margin SVM

In traditional binary classification, the task is to estimate the parameters $\boldsymbol{w}$ of a classification function f according to a set of training instances whose categories are known. Under linear assumption, the classification function can be expressed as $f(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{w}^T\boldsymbol{x} + b$. Then, the category t of a new instance $\boldsymbol{x}$ will be determined as follows

$$\begin{array}{ll} f(\boldsymbol{x}, \boldsymbol{w}) \geq 0 & \longrightarrow t = 1; \\ f(\boldsymbol{x}, \boldsymbol{w}) < 0 & \longrightarrow t = -1. \end{array} \tag{1}$$

In linear separable case, there will exist $\boldsymbol{w}$ such that

$$g_n = t_n f_n = t_n f(\boldsymbol{x}_n, \boldsymbol{w}) \geq 0, \quad n = 1, \ldots, N \tag{2}$$

where $f_n = f(\boldsymbol{x}_n, \boldsymbol{w})$, $\boldsymbol{x}_n$ indicates the nth training instances, t_n indicates the corresponding label, N indicates the number of training instances.

Hard-margin SVM defines margin as the smallest distance between the decision boundary and any of the training instances. A set of parameters which maximizes the margin will be obtained during the training process. The original objective function of linear Hard margin SVM can be expressed as

$$\max_{\boldsymbol{w},b}\{\min_n \frac{t_n f(\boldsymbol{x}_n, \boldsymbol{w})}{\|\boldsymbol{w}\|}\}, \quad n = 1, \ldots, N. \tag{3}$$

Therefore, it is obvious that Hard-margin SVM can be interpreted as a special case of maximin (Bishop 2006), which tries to find a hyperplane best classifying the worst instance/instances.

To solve Formula 3, by assigning a lower bound to the numerator and minimizing the denominator, the following Quadratic Programming (QP) formula of SVM (Vapnik 2000; Cristianini and Shawe-Taylor 2000) can be obtained

$$\begin{array}{ll} \min\limits_{\boldsymbol{w},b} & \boldsymbol{w}^T\boldsymbol{w}; \\ s.t. & t_n(\boldsymbol{w}^T\boldsymbol{x}_n + b) \geq 1; \\ & n = 1, \ldots, N. \end{array} \tag{4}$$

Different from the above formula, we can also assign an upper bound to the denominator and maximize the numerator. Then the original objective function can be reformulated into the following Quadratically Constrained Linear Programming (QCLP) formula

$$\begin{array}{ll} \max\limits_{\boldsymbol{w},b} & \min\limits_n\{t_n(\boldsymbol{w}^T\boldsymbol{x}_n + b); \\ s.t. & \boldsymbol{w}^T\boldsymbol{w} \leq 1; \\ & n = 1, \ldots, N. \end{array} \tag{5}$$

In the following derivation of our methods, a similar reformulation strategy will be adopted.

Soft-Margin SVM with Hinge Loss

It is well known that hard-margin SVM cannot deal with non-linearly separable problems efficiently. In Soft-margin SVM (Vapnik 2000; Cristianini and Shawe-Taylor 2000), by

introducing the slack variables $\xi_n > 0$ and the constraints, an objective function making a balance between the Hinge Loss and the margin is given

$$\begin{array}{cc} \min\limits_{\boldsymbol{w},b,\boldsymbol{\xi}} & C\sum\limits_{n=1}^{N}\xi_n + \frac{1}{2}\|\boldsymbol{w}\|^2; \\ s.t. & t_n(\boldsymbol{w}^T\boldsymbol{x}_n + b) + \xi_n \geq 1; \\ & \xi_n \geq 0, \quad n = 1, \ldots, N. \end{array} \tag{6}$$

With kernel $k(\boldsymbol{x}_n, \boldsymbol{x}_m)$, the corresponding Lagrange dual optimization problem can be obtained

$$\begin{array}{cc} \max\limits_{\boldsymbol{a}} & \sum\limits_{n=1}^{N} a_n - \frac{1}{2}\sum\limits_{n=1}^{N}\sum\limits_{m=1}^{N} a_n a_m t_n t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m); \\ s.t. & \sum\limits_{n=1}^{N} a_n t_n = 0; \\ & 0 \leq a_n \leq C, \quad n = 1, \ldots, N. \end{array} \tag{7}$$

$f(\boldsymbol{x})$ can be expressed in terms of Lagrange multipliers a_n and the kernel function

$$f(\boldsymbol{x}) = \sum_{n=1}^{N} a_n t_n k(\boldsymbol{x}, \boldsymbol{x}_n) + b. \tag{8}$$

MKM Criterion and Tractable Representation

As discussed above, hard-margin SVM solves maximin problem. Soft-margin SVM enhances the robustness of maximin via the introduction of slack variables and it draws a balance between the margin and the Hinge Loss by adjusting hyperparameter according to the training instances. In the following section, we will introduce MKM Criterion and adopt optimization techniques to obtain a computational tractable representation of MKM Criterion.

MKM Criterion

Maximum K-Min Gain can be readily solved via Minimum K-Max Loss. Since Minimum K-Max are more frequently discussed in the realm of optimization, in the following sections, MKM Criterion will be discussed and solved via minimizing K-Max Loss.

According to Formula 2, $g_n = t_n f_n = t_n f(\boldsymbol{x}_n, \boldsymbol{w})$ represents the classification confidence of the nth training instance $\boldsymbol{x}_n$; $g_n \geq 0$ indicates the correct classification of $\boldsymbol{x}_n$ and the larger g_n is, the more confident the classifier outputs the classification result. Therefore, we can define a vector $\boldsymbol{g} = (g_1, \ldots, g_N)^T \in R^N$ to represent a kind of classification gain of all training instances. Then, $\boldsymbol{l} = -\boldsymbol{g} = (-g_1, \ldots, -g_N)^T = (-t_1 f_1, \ldots, -t_N f_N)^T \in R^N$ can be introduced to represent a kind of classification loss. After that, by sorting $\boldsymbol{l}_n$ in descent order, we can obtain $\boldsymbol{\theta} = (\theta_{[1]}, \ldots, \theta_{[N]})^T \in R^N$ where $\theta_{[n]}$ denotes the nth largest element of $\boldsymbol{l}$, i.e. the ith largest loss of the classifier. Based on the above discussion, K-Max Loss of a classifier can be defined as $\Theta_K = \sum_{n=1}^{K}\theta_{[n]}$, with $1 \leq K \leq N$ and it measures the sum loss of the worst K training instances during classification. Therefore, the minimization of K-Max Loss with respect to the parameters can result in a classifier which performs best with regard to the worst K training instances.

Definition 1: MKM Criterion The parameters $\boldsymbol{w}$ of a classifier is obtained via the minimization of K-Max Loss Θ_K (Dong et al. 2012)

$$\min_{\boldsymbol{w}} \quad \Theta_K = \sum_{n=1}^{K}\theta_{[n]}. \quad \blacksquare \tag{9}$$

Therefore, according to the above definition, the MKM can be expressed as the following optimization problem

$$\begin{array}{cc} \min\limits_{\boldsymbol{w}} & s; \\ s.t. & l_{n_1} + \cdots + l_{n_K} \leq s; \\ where & 1 \leq n_1 \leq \cdots \leq n_K \leq N. \end{array} \tag{10}$$

i.e. minimize the maximum of all possible sums of K different elements of $\boldsymbol{y}$. If $f(\boldsymbol{x}, \boldsymbol{w})$ is defined to be convex, $l_n = -t_n f_n, n = 1, \ldots, N$ is convex.

Tractable Representation

However, it's prohibitive to solve an optimization problem with C_N^K inequality constrains. We need to summarize a compact formula with the optimization techniques (Boyd and Vandenberghe 2004). Firstly, we introduce the following lemma.

Lemma 1 For a fixed integer K, the sum of K largest elements of a vector $\boldsymbol{l}$, is equivalent to the optimal value of a linear programming problem as follows,

$$\begin{array}{cc} \max\limits_{\boldsymbol{z}} & \boldsymbol{l}^T\boldsymbol{z}; \\ s.t. & 0 \leq z_n \leq 1; \\ & \sum_n z_n = K; \\ where & n = 1, \ldots, N; \\ & \boldsymbol{z} = (z_1, \ldots, z_N)^T. \quad \blacksquare \end{array} \tag{11}$$

Proof According to the definition, $\theta_{[1]} \geq \ldots \theta_{[n]} \geq \ldots \theta_{[N]}$ denote the elements of $\boldsymbol{l}$ sorted in decreasing order.
With a fixed integer K, the optimal value of $\max \boldsymbol{l}^T\boldsymbol{z}$ under the constraints is obviously $\theta_{[1]} + \ldots \theta_{[K]}$, which is the same as the sum of K largest elements of $\boldsymbol{y}$, i.e. $\Theta_K(\boldsymbol{y}) = \theta_{[1]} + \ldots \theta_{[K]}$. $\blacksquare$

According to Lemma 1, we can obtain an equivalent representation of Θ_K with $2N + 1$ constraints. Nevertheless, $\boldsymbol{l}$ is multiplied by $\boldsymbol{z}$, this can be solved by deriving the dual of Formula 11

$$\begin{array}{cc} \min\limits_{s,\boldsymbol{u}} & Ks + \sum_n u_n; \\ s.t. & s + u_n \geq l_n; \\ & u_n \geq 0; \\ where & n = 1, \ldots, N; \\ & \boldsymbol{u} = (u_1, \ldots, u_N)^T. \end{array} \tag{12}$$

According to the strong duality theory, the optimal value of Formula 12 is equivalent to the optimal value of Formula 11. To make the above conclusion more general, we introduce the following Equivalent Representation Lemma.

Lemma 2 Θ_K can be equivalently represented as follows (1)In the objective Function:

$$\min \quad \Theta_K. \tag{13}$$

is equivalent to

$$\begin{array}{cc} \min\limits_{s,\boldsymbol{u}} & Ks+\sum\limits_{n} u_n; \\ s.t. & s+u_n \geq l_n; \\ & u_n \geq 0; \\ where & n=1,\ldots,N; \\ & \boldsymbol{u}=(u_1,\ldots,u_N)^T. \end{array} \tag{14}$$

(2)In the constraints:

$$\boldsymbol{l} \text{ satisfies } \Theta_K \leq \alpha \tag{15}$$

is equivalent to

$$\begin{array}{c} \text{There exists } s,\boldsymbol{u} \text{ which satisfy} \\ Ks+\sum\limits_{n} u_n \leq \alpha; \\ s+u_n \geq l_n; \\ u_n \geq 0; \\ n=1,\ldots,N; \\ \boldsymbol{u}=(u_1,\ldots,u_N)^T. \end{array} \tag{16}$$

Maximum K-Min Classifiers

In this section, we will adopt Lemma 2 to design Maximum K-Min Classifiers.

MKM Classifier for Traditional Classification

First, we will design a Nonlinear Maximum K-Min Classifier (NMKM) for traditional binary classification task.

Linear Representation Assumption To utilize kernel in NMKM approach, we make the following linear representation assumption directly

$$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{a},b)=\sum_{i=1}^{N} a_i s(\boldsymbol{x},\boldsymbol{x}_i)t_i+b \tag{17}$$

where $s(\boldsymbol{x},\boldsymbol{x}_i)$ denotes certain kind of similarity measurement between $\boldsymbol{x}$ and $\boldsymbol{x}_i$. Kernel function is certainly a good choice. $\boldsymbol{a}=\{a_1,\ldots,a_N\}$ denotes the weight of training instances in the task of classification. Thus, under this assumption, the local evidence is weighted more strongly than distant evidence and different points also show different importance during classification. The prediction of $\boldsymbol{x}$ is made by taking weighted linear combinations of the training set target values, where the weight is the product of a_i and $s(\boldsymbol{x},\boldsymbol{x}_i)$. During the training process of NMKM, the similarity between $\boldsymbol{x}_n$ and itself will not be considered. Therefore, the term of $s(\boldsymbol{x}_n,\boldsymbol{x}_n)$ is deleted and we have the following f_n

$$f_n=\sum_{i=1,\ldots,n-1,n+1,\ldots,N} a_i s(\boldsymbol{x}_n,\boldsymbol{x}_i)t_i+b. \tag{18}$$

Different from Equation 8, there's no constraint of a_i in our model assumption. Thus during the learning process, we may obtain a_i with negative value, which indicates that x_i may be mistakenly labeled. For a dataset carefully labeled without mistakes, the constrains of $a_i \geq 0$ can be added as prior knowledge manually.

KNN (Cover and Hart 1967) in binary classification can be regarded as a special case of the above assumption with $a_i=1, i=1,\ldots,N$ and $s(\boldsymbol{x},\boldsymbol{x}_i)$ of the following formula

$$s(\boldsymbol{x},\boldsymbol{x}_i)=\begin{cases} 1, & \text{if } \boldsymbol{x}_i \text{ is the k-nearest neighbor of } \boldsymbol{x}. \\ 0, & \text{else.} \end{cases}$$

A test point $\boldsymbol{x}$ can be classified by KNN according to the sign of $f(\boldsymbol{x})$.

In Bayesian Linear Regression (Bishop 2006), the mean of the predicated variable can be obtained according to the following formula

$$\boldsymbol{m}(\boldsymbol{x})=\sum_{i=1}^{N} k(\boldsymbol{x},\boldsymbol{x}_i)t_i.$$

This form can also be considered as a special case of assumption Equation 17, where $s(\boldsymbol{x},\boldsymbol{x}')=k(\boldsymbol{x},\boldsymbol{x}')=\beta\phi(\boldsymbol{x})^T\boldsymbol{S}_N\phi(\boldsymbol{x}')$ is known as the equivalent kernel.

Nonlinear Maximum K-Min Classifier Under the assumption of Formula 17, the original optimization Formula of NMKM is proposed as follows

$$\begin{array}{cc} \min\limits_{\boldsymbol{a},b,\boldsymbol{\beta}} & \sum\limits_{K-largest} \boldsymbol{\beta}; \\ s.t. & \left\{ \sum\limits_{i=1,\ldots,n-1,n+1,\ldots,N} a_i s(\boldsymbol{x}_n,\boldsymbol{x}_i)t_i+b \right\} t_n \leq \beta_n; \\ & \boldsymbol{a}^T\boldsymbol{a} \leq 1; \\ where & n=1,\ldots,N; \\ & \boldsymbol{a}=(a_1,\ldots,a_N)^T; \\ & \boldsymbol{\beta}=(\beta_1,\ldots,\beta_N)^T. \end{array} \tag{19}$$

In the above optimization problem, $\sum\limits_{K-largest} \boldsymbol{\beta}$ indicates the K-largest elements of vector $\boldsymbol{\beta}$. The regularization of parameters $\boldsymbol{a}$ is performed via a similar way as Formula 5. We can also add a l_1 or l_2 regularization term in the objective function, but in this way we have to deal with one more hyperparameter.

Tractable Formula of NMKM According to Lemma 2, we can obtain a tractable representation for NMKM as follows

$$\begin{array}{cc} \min\limits_{\boldsymbol{a},b,\boldsymbol{u},s} & Ks+\sum_{n=1}^{N} u_n; \\ s.t. & \left\{ \sum\limits_{i=1,\ldots,n-1,n+1,\ldots,N} a_i s(\boldsymbol{x}_n,\boldsymbol{x}_i)t_i+b \right\} t_n \leq s+u_n; \\ & \boldsymbol{a}^T\boldsymbol{a} \leq 1; \\ & u_i \geq 0 \\ where & n=1,\ldots,N; \\ & \boldsymbol{a}=(a_1,\ldots,a_N)^T; \\ & \boldsymbol{u}=(u_1,\ldots,u_N)^T. \end{array} \tag{20}$$

Thus the original optimization problem with C_N^K+1 constraints is reformulated into a convex problem with $2N+1$ constraints. As a Quadratically Constrained Linear Programming (QCLP) problem, standard convex optimization methods, such as interior-point methods (Boyd and Vandenberghe 2004) (Wright 1997), can also be adopted to solve the above problem efficiently and a global maximum solution is guaranteed.

MKM Classifier for Semi-supervised Classification

Semi-supervised Classification and S3VM In practical classification problems, since the labeling process is always expensive and laborious, semi-supervised Learning is proposed and tries to improve the classification performance with unlabeled training sets (Chapelle et al. 2006). Therefore, semi-supervised classifiers make use of both labeled and unlabeled data for training and fall between unsupervised classifier (without any labeled training data) and supervised classifier (with completely labeled training data).

As illustrated in (Bennett, Demiriz, and others 1999), Semi-Supervised SVM(S3VM) is a natural extension of SVM in semi-supervised problems and the objective function of S3VM can be written as

$$\begin{array}{cc} \min\limits_{\boldsymbol{w},b,\boldsymbol{\eta},\boldsymbol{\xi},\boldsymbol{z},\boldsymbol{d}} & C_1\sum\limits_{i=1}^{L}\eta_i + C_2\sum\limits_{j=L+1}^{L+U}(\xi_i+z_j) + \boldsymbol{w}^T\boldsymbol{w}; \\ s.t. & y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)+\eta_i \geq 1; \\ & \boldsymbol{w}^T\boldsymbol{x}_j+b+\xi_j+M(1-d_j)\geq 1; \\ & -(\boldsymbol{w}^T\boldsymbol{x}_j+b)+z_j+Md_j\geq 1; \\ & \eta_i\geq 0;\quad \xi_j\geq 0;\quad z_j\geq 0; \\ where & M>0 \text{ is a sufficiently large constant}; \\ & i=1,\ldots,L;\quad j=L+1,\ldots,L+U; \\ & \boldsymbol{\eta}=(\eta_1,\ldots,\eta_L)^T; \\ & \boldsymbol{\xi}=(\xi_{L+1},\ldots,\xi_{L+U})^T; \\ & \boldsymbol{z}=(z_{L+1},\ldots,z_{L+U})^T; \\ & \boldsymbol{d}=(d_{L+1},\ldots,d_{L+U})^T; \\ & d_j\in\{0,1\}. \end{array} \tag{21}$$

In the above formula, L indicates the size of labeled training set; U indicates the size of unlabeled training set; d_j is a binary variable which indicates the predicated category of the unlabeled training instances; $\sum_{i=1}^{L}\eta_i$ measures the Hinge Loss of the labeled training instances and $\sum_{j=L+1}^{L+U}(\xi_i+z_j)$ measures the Hinge Loss of the unlabeled training instances. Since the optimization problem has both binary variables and continuous variables, it is a Mixed-Integer Quadratic Programming (MIQP) problem and the globally optimal solution can be solved via commercial solvers, such as Gurobi (Optimization 2012) and Cplex (CPLEX 2009).

Semi-supervised Maximum K-Min Classifier We will adopt MKM Criterion to measure the loss of unlabeled training set and we can obtain the following formula

$$\begin{array}{cc} \min\limits_{\boldsymbol{w},b,\boldsymbol{\eta},\boldsymbol{d},\boldsymbol{\alpha},\boldsymbol{\beta}} & \sum\limits_{i=1}^{L}\eta_i + C\sum\limits_{K-largest}\{\boldsymbol{\alpha};\boldsymbol{\beta}\}; \\ s.t. & y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)+\eta_i\geq 1; \\ & \boldsymbol{w}^T\boldsymbol{x}_j+b+M(1-d_j)\leq \alpha_j; \\ & -(\boldsymbol{w}^T\boldsymbol{x}_j+b)+Md_j\leq \beta_j; \\ & \boldsymbol{w}^T\boldsymbol{w}\leq 1;\eta_i\geq 0; \\ where & M>0 \text{ is a sufficiently large constant}; \\ & i=1,\ldots,L;\quad j=L+1,\ldots,L+U; \\ & \boldsymbol{\eta}=(\eta_1,\ldots,\eta_L)^T; \\ & \boldsymbol{\alpha}=(\alpha_{L+1},\ldots,\alpha_{L+U})^T; \\ & \boldsymbol{\beta}=(\beta_{L+1},\ldots,\beta_{L+U})^T; \\ & \boldsymbol{d}=(d_{L+1},\ldots,d_{L+U})^T; \\ & d_j\in\{0,1\}. \end{array} \tag{22}$$

In the above formula, $\sum_{K-largest}\{\boldsymbol{\alpha};\boldsymbol{\beta}\}$ indicates the K-largest elements of a set $\boldsymbol{\gamma}$ which contains all elements of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, i.e. $\boldsymbol{\gamma}=(\alpha_{L+1},\ldots,\alpha_{L+U},\beta_{L+1},\ldots,\beta_{L+U})^T$.

Tractable Formula of SMKM According to Lemma 2, we can obtain a tractable representation for SMKM as follows

$$\begin{array}{cc} \min\limits_{\boldsymbol{w},b,\boldsymbol{\eta},\boldsymbol{d},\boldsymbol{u},s} & \sum\limits_{i=1}^{L}\eta_i + C(Ks+\sum_{n=1}^{2U}u_n); \\ s.t. & y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)+\eta_i\geq 1; \\ & \boldsymbol{w}^T\boldsymbol{x}_j+b+M(1-d_j)-s-u_t\leq 0; \\ & -(\boldsymbol{w}^T\boldsymbol{x}_j+b)+Md_j-s-u_{t+U}\leq 0; \\ & \boldsymbol{w}^T\boldsymbol{w}\leq 1;\boldsymbol{\eta}\geq 0;\boldsymbol{u}\geq 0; \\ where & M>0 \text{ is a sufficiently large constant}; \\ & i=1,\ldots,L;\quad t=1,\ldots,U; \\ & j=L+1,\ldots,L+U; \\ & \boldsymbol{\eta}=(\eta_1,\ldots,\eta_L)^T; \\ & \boldsymbol{u}=(u_1,\ldots,u_{2U})^T; \\ & \boldsymbol{d}=(d_{L+1},\ldots,d_{L+U})^T; \\ & d_j\in\{0,1\}. \end{array} \tag{23}$$

The above formula is a Mixed-Integer Quadratically Constrained Programming (MIQCP) problem and the globally optimal solution can be obtained via optimization solvers, such as Gurobi (Optimization 2012) and Cplex (CPLEX 2009).

Experiment

Traditional Classification Experiment

In the experiment of traditional classification , NMKM with Radical Basis Function (RBF) kernel matrix is compared to SVM with RBF kernel and L2-regularized Logistic Regression (LR). NMKM is implemented using cvx toolbox[1] (CVX Research ; Grant and Boyd 2008) in matlab environment with the solver of SeDuMi (Sturm 1999). LR is implemented using libliner toolbox[2] and kernel SVM is implemented using libsvm toolbox[3] (Chang and Lin 2011).

Ten publicly available binary classification datasets are adopted in the experiment. The detailed features are shown in Table 1. The parameters of NMKM and SVM are chosen via 10 fold cross-validation during the training stage. For NMKM, 21 values for parameter K ranging from 2 to N are tested, where N indicates the number of training instances. 31 values for parameters C ranging from 2^{-20} to 2^{20} are tested for SVM. The step size of C in the log searching domain of $[-20,-10)$ or $(10,20]$ is 2 and $[-10,10]$ is 1.

As shown in Table 2, among 10 different datasets, NMKM achieves best accuracy in 6 datasets. SVM performs best in 2 datesets and LR shows best performance in 2 datasets. Among all datasets which SVM or LR performes better than NMKM, NMKM performs slightly weaker than SVM or LR. While for datasets which NMKM gains better performance, the accuracy of NMKM may be much higher than MSVM and LR. The accuracy gap of SVM in dataset

[1] http://cvxr.com/

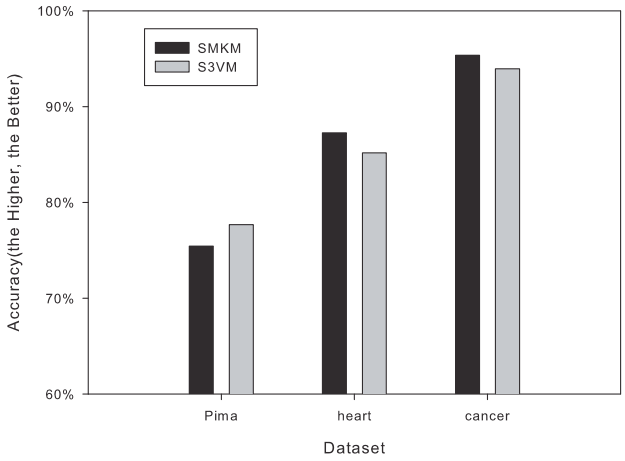
[2] www.csie.ntu.edu.tw/ cjlin/liblinear

[3] www.csie.ntu.edu.tw/ cjlin/libsvm

Table 1: Detailed Features of the Binary Classification Datasets in NMKM Experiment

datasets	sources	features	instances(training)	instances(testing)	instances(total)
australian	Statlog / Australian	14	414	276	690
breast-cancer	UCI / Wisconsin Breast Cancer	10	410	273	683
colon-cancer	paper (Alon et al. 1999)	62	37	25	62
diabetes	UCI / Pima Indians Diabetes	8	460	308	768
fourclass	paper (Ho and Kleinberg 1996)	2	517	345	862
german.numer	Statlog / German	24	600	400	1000
heart	Statlog / Heart	13	162	108	270
ionosphere	UCI / Ionosphere	34	210	141	351
liver-disorders	UCI / Liver-disorders	6	207	138	345
splice	Delve / splice	60	1,000	2,175	3,175

Figure 2: Experiment Results of SMKM



liver and LR in dataset *fourclass* from NMKM is 8.86% and 27.85% respectively. Thus we can conclude, in traditional classification experiment, compared with SVM and LR, NMKM has obtained competitive classification performance.

Semi-supervised Classification Experiment

In the experiment of semi-supervised classification , SMKM is compared with S3VM. Both SMKM and S3VM are implemented using cvx toolbox (CVX Research ; Grant and Boyd 2008) in matlab environment with the solver of Gurobi (Optimization 2012). Three publicly available UCI datasets of 'Cancer', 'Heart' and 'Pima' are selected for comparison. All datasets are randomly splitted into labeled training set (50 instances), unlabeled training set (50 instances) and the testing set (all other instances). The hyperparameters of NMKM (C, K) and SVM (C_1, C_2) are chosen via 10 fold cross-validation during the training stage. 11 values for C, C_1, C_2 ranging from 2^{-5} to 2^5 are tested. 10 values for K ranging from 1 to 20 are tested. As shown in Figure 2, SMKM performs better in the datasets of 'Cancer' and 'Heart', while S3VM performs better in the dataset of 'Pima'. Therefore, SMKM has obtained competitive performance when comparing against S3VM in semi-supervised experiment.

Table 2: Experiment Results of NMKM (Accuracy, the higher the better)

Algorithm	DataSet				
	australian	breast	colon	diabetes	fourclass
$NMKM$	**85.07%**	**96.35%**	84.40%	**74.84%**	**99.88%**
SVM	76.67%	93.54%	**87.20%**	67.37%	99.65%
LR	84.02%	95.07%	84.40%	66.62%	72.03%
	german	heart	ionosphere	liver	splice
$NMKM$	76.11%	80.65%	**94.39%**	**71.42%**	82.25%
SVM	65.75%	73.98%	88.33%	62.5%6	**83.50%**
LR	**76.38%**	**81.67%**	82.20%	67.61%	79.38%

[i] NMKM and SVM are implemented with RBF kernel;
[ii] The best result for each dataset is shown in boldface.

Conclusion

In this paper, a general Maximum K-Min approach for classification is proposed. By reformulating the original objective function into a compact representation, the optimization of MKM Criterion becomes tractable. To verify the performance of MKM methods, a Nonlinear Maximum K-Min (NMKM) classifier and a Semi-supervised Maximum K-Min (SMKM) classifier are presented for traditional classification task and semi-supervised classification task respectively. As shown in the experiments, the classification performance of Maximum K-Min classifiers is competitive with Hinge Loss classifiers.

Acknowledge

We would like to thank the anonymous reviewers for their insight and constructive comments, which helped improve the presentation of this paper. Meanwhile, this work was partially supported by National Natural Science Foundation of China under Grant No.61005025, 61002051, 61273217, 61175011 and 61171193, the 111 project under Grant No.B08004 and the Fundamental Research Funds for the Central Universities.

References

Alon, U.; Barkai, N.; Notterman, D. A.; Gish, K. W.; Ybarra, S.; Mack, D.; and Levine, A. J. 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of The National Academy of Sciences* 96:6745–6750.

Andrews, S.; Tsochantaridis, I.; and Hofmann, T. 2002. Support vector machines for multiple-instance learning. *Advances in neural information processing systems* 15:561–568.

Bennett, K.; Demiriz, A.; et al. 1999. Semi-supervised support vector machines. *Advances in Neural Information processing systems* 368–374.

Bishop, C. 2006. *Pattern recognition and machine learning*. Springer.

Boyd, S., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge Univ Press.

Chang, C.-C., and Lin, C.-J. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2(3):27.

Chapelle, O.; Schölkopf, B.; Zien, A.; et al. 2006. *Semi-supervised learning*, volume 2. MIT press Cambridge.

Cover, T., and Hart, P. 1967. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on* 13(1):21–27.

CPLEX, I. I. 2009. V12. 1: Users manual for cplex. *International Business Machines Corporation* 46(53):157.

Cristianini, N., and Shawe-Taylor, J. 2000. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

CVX Research, I. CVX: Matlab software for disciplined convex programming, version 2.0 beta.

Dong, M.; Yin, L.; Deng, W.; Wang, Q.; Yuan, C.; Guo, J.; Shang, L.; and Ma, L. 2012. A linear max k-min classifier. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2967–2971. IEEE.

Grant, M., and Boyd, S. 2008. Graph implementations for nonsmooth convex programs. In Blondel, V.; Boyd, S.; and Kimura, H., eds., *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences. Springer-Verlag Limited. 95–110.

Ho, T. K., and Kleinberg, E. M. 1996. Building projectable classifiers of arbitrary complexity. In *International Conference on Pattern Recognition*, volume 2.

Optimization, G. 2012. Gurobi optimizer reference manual. *URL: http://www. gurobi. com*.

Sturm, J. F. 1999. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software* 11(1-4):625–653.

Vapnik, V. 2000. *The nature of statistical learning theory*. Springer-Verlag New York Inc.

Wright, S. 1997. *Primal-dual interior-point methods*, volume 54. Society for Industrial Mathematics.