

# Max-Margin Multiple-Instance Learning via Semidefinite Programming

Yuhong Guo

Department of Computer & Information Sciences  
Temple University, Philadelphia, PA 19122, USA  
yuhong@temple.edu

**Abstract.** In this paper, we present a novel semidefinite programming approach for multiple-instance learning. We first formulate the multiple-instance learning as a combinatorial maximum margin optimization problem with additional instance selection constraints within the framework of support vector machines. Although solving this primal problem requires non-convex programming, we nevertheless can then derive an equivalent dual formulation that can be relaxed into a novel convex semidefinite programming (SDP). The relaxed SDP has  $\mathcal{O}(T)$  free parameters where  $T$  is the number of instances, and can be solved using a standard interior-point method. Empirical study shows promising performance of the proposed SDP in comparison with the support vector machine approaches with heuristic optimization procedures.

## 1 Introduction

Multiple-instance learning was introduced by Dietterich et al. [1] to solve a generalized supervised classification problem where the data set is composed of many *bags* such that each of them contains many instances and the class labels are associated with the bags, instead of individual instances. A bag is labeled as a *positive* bag if it contains at least one positive instance; otherwise it is labeled as a negative bag. Different from the standard supervised learning where all training instances are with known labels, the labels for individual instances in a positive bag is unknown in a multiple-instance learning problem, which makes the multiple-instance learning a much more challenging problem than the standard supervised classification.

Multiple-instance learning problems arise naturally from many application domains. One prominent example is the problem of drug activity prediction [1], where each molecule has a bag of different conformations, and the molecule qualified to make a drug has at least one conformation that could tightly bind to the target protein molecules. A second application is in content-based image retrieval or classification [2–4], where each image can be viewed as a bag of local subimages and one image is relevant with respect to one particular category if it has at least one relevant subimage. Another application is the problem of text categorization [5], where each document contains multiple passages over different

topics and a document is considered relevant regarding to one particular topic when it has one or more passages on this topic.

Motivated by these application challenges, multiple-instance learning has become one active research area in machine learning. A number of multiple-instance learning approaches have been developed in the literature, including special purpose algorithms using axis-parallel rectangular hypothesis [1], diverse density [3, 6], kernel methods [7], support vector machines [5, 8], ensemble methods [9], boosting methods [10], non-i.i.d. style methods [11] and etc.

In this paper, we propose to extend one popular classification method, maximum margin classification, to address the multiple-instance learning problem. Two maximum margin multiple-instance learning methods, *mi-SVM* and *MI-SVM*, based on support vector machines have been proposed in [5], *mi-SVM* for instance-level classification and *MI-SVM* for bag-level classification. The *mi-SVM* explicitly treats the instance labels in positive bags as unobserved hidden variables subject to constraints defined by their bag labels. In comparison, the *MI-SVM* aims to maximize the bag margin, which is defined as the margin of the most positive instance in case of positive bags, or the margin of the least negative instance in case of negative bags. However, due to the combinatorial nature of the formulated maximum margin problems, iterative heuristic procedures were used to conduct optimization in [5], which naturally suffer from the problem of local optima. Here we propose to formulate the multiple-instance learning as a combinatorial optimization problem of maximizing the classification margin with additional instance selection constraints. Like the *mi-SVM*, our approach can be categorized as an instance-level method. However, instead of figuring out the labels for all instances in positive bags, our approach selects only positive instances to use and ignore the negative ones in positive bags. The primal maximum margin formulation we developed is still non-convex, but we nevertheless can derive its equivalent dual formulation which can be relaxed into a convex semidefinite programming (SDP) problem by exploiting the Schur complement lemma. The relaxed SDP has  $\mathcal{O}(T)$  free parameters where  $T$  is the number of instances, and can be solved using a standard interior-point method. Our empirical study shows promising performance of the proposed SDP in comparison with the support vector machine approaches with heuristic optimization procedures.

The remainder of this paper is organized as follows. After establishing the preliminaries and notations in Section 2, we present our maximum margin formulation for multiple-instance learning in Section 3. In Section 4, we derive an equivalent dual formulation and show it can be finally relaxed to yield a convex semidefinite programming problem which allows a global solution to be computed. Experimental results are reported in Section 5. We finally conclude the paper in Section 6.

## 2 Preliminaries

Since our approach is based on support vector machines (SVMs), we will first establish the background knowledge of SVMs as well as establish the notation

we will use. Assume we are given labeled training instances  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)$ , where each instance is assigned to one of two classes  $y_i \in \{-1, +1\}$ . The goal of a SVM is to find the linear discriminant function  $f_{\mathbf{w}, b}(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$  that achieves maximum margin (*separation*) between the two classes in  $\phi(\mathbf{x})$  space. Note here  $\phi(\mathbf{x})$  denotes the general feature vector produced from the original feature vector  $\mathbf{x}$ , and it is introduced to cope with nonlinear classification. The standard primal soft margin SVM is formulated as follow

$$\begin{aligned} \min_{\mathbf{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \boldsymbol{\xi}^\top \mathbf{e} \\ \text{s.t.} \quad & y_i (\mathbf{w}^\top \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \forall_{i=1}^T \\ & \boldsymbol{\xi} \geq 0 \end{aligned} \tag{1}$$

where slack variables  $\boldsymbol{\xi}$  are introduced to cope with noisy instances and the non-separability of the training data;  $C$  is a parameter that controls the tradeoff between the separation margin and the misclassification error;  $b$  is a parameter to control the bias; and  $\mathbf{e}$  denotes the vector of all 1 entries. For the simplicity reason, we will use the same  $\mathbf{e}$  notation to denote any vectors with all 1 entries later in the paper. The length of the  $\mathbf{e}$  vector for each of its appearance can be determined from the context. By introducing Lagrangian multipliers and following the standard procedure, an equivalent dual formulation for SVM in (1) can be obtained

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}^\top \mathbf{e} - \frac{1}{2} \boldsymbol{\alpha}^\top (K \circ \mathbf{y} \mathbf{y}^\top) \boldsymbol{\alpha} \\ \text{s.t.} \quad & 0 \leq \boldsymbol{\alpha} \leq C, \boldsymbol{\alpha}^\top \mathbf{y} = 0 \end{aligned} \tag{2}$$

where  $\boldsymbol{\alpha}$  denotes the vector of dual variables;  $K$  denotes the  $T \times T$  kernel matrix formed from the inner products of feature vectors  $\Phi = [\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_T)]$  such that  $K = \Phi^\top \Phi$ ;  $\mathbf{y}$  denotes the label vector, such that  $\mathbf{y} = [y_1, \dots, y_T]^\top$ ;  $A \circ B$  denotes componentwise matrix multiplication.

In the following section, we will extend the standard SVM to address the problem of multiple-instance learning.

### 3 Max-Margin Multiple-Instance Learning

Different from the standard supervised learning scenario where a label is assigned to each training instance, in multiple-instance learning, a label is assigned to a bag of instances. A bag is labeled as a positive bag if it contains at least one positive instance; otherwise it is labeled as a negative bag, which means all instances in negative bags are negative instances. Thus the difficulty for extending any standard supervised learning methods to address the multiple-instance learning problem lies in that the labels for instances in positive bags are unknown. Moreover, different from the standard semi-supervised learning scenario, we need to guarantee that at least one instance from each positive bag gets a positive label.

The *mi-SVM* proposed in [5] views the instance labels in positive bags as hidden variables, and maximizes a soft margin criterion jointly over discriminant model parameters and possible label assignments while taking an extra checking step to enforce that at least one instance gets a positive label for each positive bag. However, there are usually a lot of ambiguities with regard to the label assignments over the hidden variables. On the other hand, we can obtain many confirmed negative instances from negative bags. Based on these observations, we propose to select only a set of positive instances from positive bags to use together with the negative instances from negative bags for multiple-instance learning. Our intuition is to incorporate only the most useful information into the model learning process while avoiding the unnecessary ambiguities. Specifically, we propose to formulate the multiple-instance learning as a combinatorial optimization problem that maximizes a soft SVM margin criterion jointly over both the model parameters and the instance selection variables. The instance selection variables are used to choose the most informative positive instances from the positive bags such that when the selected positive instances are incorporated into the proposed maximum margin model, the soft margin criterion can be maximumly optimized. Below we will present this joint optimization model in detail. Moreover, we will show later that this optimization model with instance selection variables can lead to a simple SDP formulation.

Assume we are given a multiple-instance training set with  $N$  bags of instances  $\{\mathbf{B}_1, \dots, \mathbf{B}_N\}$ , where the first  $N_p$  bags are positive bags, following by  $N_n$  negative bags such that  $N_p + N_n = N$ . Assume each bag  $\mathbf{B}_i$  contains  $t_i$  instances such that  $\sum_i^{N_p} t_i = T_p$ ,  $\sum_{i=N_p+1}^N t_i = T_n$  and  $T_p + T_n = T$ . Our maximum margin multiple-instance learning can be formulated as follows

$$\begin{aligned} \min_{\boldsymbol{\eta}} \min_{\mathbf{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \boldsymbol{\xi}^\top [\boldsymbol{\eta}; \mathbf{e}] \\ \text{s.t.} \quad & y_i (\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \forall_{i=1}^T \\ & \boldsymbol{\xi} \geq 0 \\ & \boldsymbol{\eta} \in \{0, 1\}^{T_p \times 1} \\ & \mathbf{A} \boldsymbol{\eta} \geq \mathbf{e} \end{aligned} \tag{3}$$

where  $\boldsymbol{\eta}$  denotes the vector of instance selection binary variables;  $y_i = 1$  for  $i = 1, \dots, T_p$  and  $y_i = -1$  for  $i = 1 + T_p, \dots, T$ ;  $\mathbf{A}$  is a  $N_p \times T_p$  binary matrix such that

$$\mathbf{A} = \begin{bmatrix} \text{ones}(1, t_1), & \text{zeros}(1, t_2), & \dots, & \text{zeros}(1, t_{N_p}) \\ \text{zeros}(1, t_1), & \text{ones}(1, t_2), & \dots, & \text{zeros}(1, t_{N_p}) \\ \vdots & \vdots & \vdots & \vdots \\ \text{zeros}(1, t_1), & \text{zeros}(1, t_2), & \dots, & \text{ones}(1, t_{N_p}) \end{bmatrix}$$

and all the other notations are same as introduced before. Note that the constraint  $\mathbf{A} \boldsymbol{\eta} \geq \mathbf{e}$  is used to guarantee that at least one positive instance from each positive bag will be selected. Given fixed  $\boldsymbol{\eta}$ , the optimization problem (3) will

become a standard SVM optimization problem over a training set formed by the negative instances from negative bags and the positive instances selected using  $\boldsymbol{\eta}$  from positive bags.

The minimization problem (3) we formulated is a NP-hard combinatorial optimization problem. In order to obtain an efficient convex optimization, we first need to derive its dual formulation.

**Proposition 1.** *For fixed  $\boldsymbol{\eta}$ , the inner minimization problem in (3), that is*

$$\begin{aligned} \min_{\mathbf{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \boldsymbol{\xi}^\top [\boldsymbol{\eta}; \mathbf{e}] \\ \text{s.t.} \quad & y_i (\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b) \geq 1 - \xi_i, \forall_{i=1}^T \\ & \boldsymbol{\xi} \geq 0 \end{aligned} \quad (4)$$

is equivalent to the following dual maximization problem

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}^\top \mathbf{e} - \frac{1}{2} \boldsymbol{\alpha}^\top (K \circ \mathbf{y}\mathbf{y}^\top) \boldsymbol{\alpha} \\ \text{s.t.} \quad & 0 \leq \boldsymbol{\alpha} \leq C[\boldsymbol{\eta}; \mathbf{e}] \\ & \boldsymbol{\alpha}^\top \mathbf{y} = 0 \end{aligned} \quad (5)$$

*Proof.* The proof is simple. Note the minimization problem in (4) is a slightly modified version of the standard SVM optimization in (1). The only difference lies in that the  $\boldsymbol{\xi}$  in the objective function of (4) is weighted by a vector  $[\boldsymbol{\eta}; \mathbf{e}]$ . Thus following the standard procedure for deriving a dual formulation of SVMs, an equivalent dual formulation (5) can be obtained. ■

Exploiting Proposition 1, the minimization problem (3) can be rewritten into the following equivalent min-max optimization problem by simply replacing the inner minimization of (3) with its equivalent dual formulation (5)

$$\begin{aligned} \min_{\boldsymbol{\eta}} \max_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}^\top \mathbf{e} - \frac{1}{2} \boldsymbol{\alpha}^\top (K \circ \mathbf{y}\mathbf{y}^\top) \boldsymbol{\alpha} \\ \text{s.t.} \quad & 0 \leq \boldsymbol{\alpha} \leq C[\boldsymbol{\eta}; \mathbf{e}] \\ & \boldsymbol{\alpha}^\top \mathbf{y} = 0 \\ & \boldsymbol{\eta} \in \{0, 1\}^{T_p \times 1} \\ & \mathbf{A}\boldsymbol{\eta} \geq \mathbf{e} \end{aligned} \quad (6)$$

Although the dual optimization problem in (6) does not provide a convex solution immediately, it provides a foundation for further reformulation.

## 4 Semidefinite Programming

In this section, we will reformulate the min-max optimization problem (6) obtained in the previous section to finally get a convex semidefinite programming problem which can provide a global solution without local optima.

**Theorem 1.** *The combinatorial min-max optimization problem in (6) is equivalent to the following minimization problem*

$$\begin{aligned}
& \min_{\boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \epsilon, \delta} \delta & (7) \\
& \text{s.t.} \quad \begin{pmatrix} K \circ \mathbf{y}\mathbf{y}^\top & (\mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\lambda} + \epsilon\mathbf{y}) \\ (\mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\lambda} + \epsilon\mathbf{y})^\top & 2\delta - 2C\boldsymbol{\lambda}^\top[\boldsymbol{\eta}; \mathbf{e}] \end{pmatrix} \succeq 0 \\
& \quad \boldsymbol{\mu} \geq 0 \\
& \quad \boldsymbol{\lambda} \geq 0 \\
& \quad \boldsymbol{\eta} \in \{0, 1\}^{T_p \times 1} \\
& \quad \mathbf{A}\boldsymbol{\eta} \geq \mathbf{e}
\end{aligned}$$

*Proof.* The min-max optimization problem (6) can be equivalently rewritten as

$$\begin{aligned}
& \min_{\boldsymbol{\eta}} \delta & (8) \\
& \text{s.t.} \quad \delta \geq \max_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^\top \mathbf{e} - \frac{1}{2} \boldsymbol{\alpha}^\top (K \circ \mathbf{y}\mathbf{y}^\top) \boldsymbol{\alpha} \\
& \quad 0 \leq \boldsymbol{\alpha} \leq C[\boldsymbol{\eta}; \mathbf{e}] \\
& \quad \boldsymbol{\alpha}^\top \mathbf{y} = 0 \\
& \quad \boldsymbol{\eta} \in \{0, 1\}^{T_p \times 1} \\
& \quad \mathbf{A}\boldsymbol{\eta} \geq \mathbf{e}
\end{aligned}$$

Below we will express the constraint  $\delta \geq \max_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^\top \mathbf{e} - \frac{1}{2} \boldsymbol{\alpha}^\top (K \circ \mathbf{y}\mathbf{y}^\top) \boldsymbol{\alpha}$  as a linear matrix inequality for given  $\boldsymbol{\eta}$ .

First define the Lagrangian of the maximization problem (5) by

$$\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \epsilon, \delta) = \boldsymbol{\alpha}^\top \mathbf{e} - \frac{1}{2} \boldsymbol{\alpha}^\top (K \circ \mathbf{y}\mathbf{y}^\top) \boldsymbol{\alpha} + \boldsymbol{\mu}^\top \boldsymbol{\alpha} + \boldsymbol{\lambda}^\top (C[\boldsymbol{\eta}; \mathbf{e}] - \boldsymbol{\alpha}) + \epsilon \boldsymbol{\alpha}^\top \mathbf{y}$$

where  $\boldsymbol{\mu} \geq 0$ ,  $\boldsymbol{\lambda} \geq 0$  and  $\epsilon \in \mathbb{R}$ . By duality [12], we have

$$\max_{\boldsymbol{\alpha}} \min_{\boldsymbol{\mu}, \boldsymbol{\lambda}, \epsilon, \delta} \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \epsilon, \delta) = \min_{\boldsymbol{\mu}, \boldsymbol{\lambda}, \epsilon, \delta} \max_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \epsilon, \delta).$$

Then the inner maximization over  $\boldsymbol{\alpha}$ ,  $\max_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \epsilon, \delta)$ , can be easily solved by determining a critical point, since  $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \epsilon, \delta)$  is concave in  $\boldsymbol{\alpha}$ . By setting  $\partial \mathcal{L} / \partial \boldsymbol{\alpha} = 0$ , we obtain  $\boldsymbol{\alpha} = (K \circ \mathbf{y}\mathbf{y}^\top)^{-1} (\mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\lambda} + \epsilon\mathbf{y})$ . Substituting this into  $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \epsilon, \delta)$ , we can form the following dual problem of (5)

$$\begin{aligned}
& \min_{\boldsymbol{\mu}, \boldsymbol{\lambda}, \epsilon, \delta} C\boldsymbol{\lambda}^\top[\boldsymbol{\eta}; \mathbf{e}] + \frac{1}{2} (\mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\lambda} + \epsilon\mathbf{y})^\top (K \circ \mathbf{y}\mathbf{y}^\top)^{-1} (\mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\lambda} + \epsilon\mathbf{y}) & (9) \\
& \text{s.t.} \quad \boldsymbol{\mu} \geq 0 \\
& \quad \boldsymbol{\lambda} \geq 0
\end{aligned}$$

This implies that for any  $\delta$ , the constraint  $\delta \geq \max_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^\top \mathbf{e} - \frac{1}{2} \boldsymbol{\alpha}^\top (K \circ \mathbf{y}\mathbf{y}^\top) \boldsymbol{\alpha}$  holds if and only if there exist  $\boldsymbol{\mu} \geq 0$ ,  $\boldsymbol{\lambda} \geq 0$  and  $\epsilon$  such that

$$\delta \geq C\boldsymbol{\lambda}^\top[\boldsymbol{\eta}; \mathbf{e}] + \frac{1}{2} (\mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\lambda} + \epsilon\mathbf{y})^\top (K \circ \mathbf{y}\mathbf{y}^\top)^{-1} (\mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\lambda} + \epsilon\mathbf{y})$$

or equivalently using the Schur complement lemma [12] such that

$$\begin{pmatrix} K \circ \mathbf{y}\mathbf{y}^\top & (\mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\lambda} + \epsilon\mathbf{y}) \\ (\mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\lambda} + \epsilon\mathbf{y})^\top & 2\delta - 2C\boldsymbol{\lambda}^\top[\boldsymbol{\eta}; \mathbf{e}] \end{pmatrix} \succeq 0 \quad (10)$$

Substituting this into (8) yields (7). ■

However, the minimization problem (7) is still not convex for two reasons. First, there is a bilinear term  $\boldsymbol{\lambda}^\top[\boldsymbol{\eta}; \mathbf{e}]$  in the matrix inequality constraint (10), which makes the constraint non-convex. Second, the existence of binary constraints over variables  $\boldsymbol{\eta}$  makes the overall optimization problem a combinatorial optimization. We thus need to solve these two issues to obtain an efficient convex optimization problem.

For the problem of bilinear term, we notice that  $\boldsymbol{\lambda}^\top[\boldsymbol{\eta}; \mathbf{e}] = \boldsymbol{\lambda}_{1:T_p}^\top \boldsymbol{\eta} + \boldsymbol{\lambda}_{T_p+1:T}^\top \mathbf{e}$ , and  $2\boldsymbol{\lambda}_{1:T_p}^\top \boldsymbol{\eta} = (\boldsymbol{\lambda}_{1:T_p} + \boldsymbol{\eta})^\top (\boldsymbol{\lambda}_{1:T_p} + \boldsymbol{\eta}) - \boldsymbol{\lambda}_{1:T_p}^\top \boldsymbol{\lambda}_{1:T_p} - \boldsymbol{\eta}^\top \boldsymbol{\eta}$ . Note that since  $\boldsymbol{\eta}$  is a vector of binary variables, thus  $\boldsymbol{\eta}^\top \boldsymbol{\eta} = \boldsymbol{\eta}^\top \mathbf{e}$ . Now we introduce two new variables  $g$  and  $h$  such that  $g = (\boldsymbol{\lambda}_{1:T_p} + \boldsymbol{\eta})^\top (\boldsymbol{\lambda}_{1:T_p} + \boldsymbol{\eta})$  and  $h = \boldsymbol{\lambda}_{1:T_p}^\top \boldsymbol{\lambda}_{1:T_p}$ . For simplicity reason, we also let  $\mathbf{u}$  denote the constant vector  $[\mathbf{0}; \mathbf{e}]$ , where  $\mathbf{0}$  is a vector of all 0 entries, such that  $\boldsymbol{\lambda}^\top \mathbf{u} = \boldsymbol{\lambda}^\top [\mathbf{0}; \mathbf{e}] = \boldsymbol{\lambda}_{T_p+1:T}^\top \mathbf{e}$ . Therefore

$$2\boldsymbol{\lambda}^\top[\boldsymbol{\eta}; \mathbf{e}] = g - h - \boldsymbol{\eta}^\top \mathbf{e} + 2\boldsymbol{\lambda}^\top \mathbf{u}.$$

Substituting this back to the matrix inequality constraint in (7), we obtain an equivalent optimization problem

$$\begin{aligned} \min_{\boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \epsilon, \delta, g, h} \quad & \delta & (11) \\ \text{s.t.} \quad & \begin{pmatrix} K \circ \mathbf{y}\mathbf{y}^\top & (\mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\lambda} + \epsilon\mathbf{y}) \\ (\mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\lambda} + \epsilon\mathbf{y})^\top & 2\delta - C(g - h - \boldsymbol{\eta}^\top \mathbf{e} + 2\boldsymbol{\lambda}^\top \mathbf{u}) \end{pmatrix} \succeq 0 \\ & g = (\boldsymbol{\lambda}_{1:T_p} + \boldsymbol{\eta})^\top (\boldsymbol{\lambda}_{1:T_p} + \boldsymbol{\eta}) \\ & h = \boldsymbol{\lambda}_{1:T_p}^\top \boldsymbol{\lambda}_{1:T_p} \\ & \boldsymbol{\mu} \geq 0 \\ & \boldsymbol{\lambda} \geq 0 \\ & \boldsymbol{\eta} \in \{0, 1\}^{T_p \times 1} \\ & \mathbf{A}\boldsymbol{\eta} \geq \mathbf{e} \end{aligned}$$

Now we have successfully got rid of the bilinear term from the matrix inequality constraint. However, two new quadratic equality constraints have been introduced. In order to obtain a convex optimization, we need to relax the two quadratic equality constraints into inequality constraints

$$\begin{aligned} g - (\boldsymbol{\lambda}_{1:T_p} + \boldsymbol{\eta})^\top (\boldsymbol{\lambda}_{1:T_p} + \boldsymbol{\eta}) & \geq 0 \\ h - \boldsymbol{\lambda}_{1:T_p}^\top \boldsymbol{\lambda}_{1:T_p} & \geq 0 \end{aligned}$$

Using quadratic inequality constraints to replace corresponding equality constraints is a typical relaxation technique used in the literature, e.g. [13], towards

obtaining convex semidefinite approximations. Here the inequality constraints above can then be rewritten equivalently into convex linear matrix inequality constraints according to the Schur complement lemma [12]

$$\begin{pmatrix} I & (\boldsymbol{\lambda}_{1:T_p} + \boldsymbol{\eta}) \\ (\boldsymbol{\lambda}_{1:T_p} + \boldsymbol{\eta})^\top & g \end{pmatrix} \succeq 0 \quad (12)$$

$$\begin{pmatrix} I & \boldsymbol{\lambda}_{1:T_p} \\ \boldsymbol{\lambda}_{1:T_p}^\top & h \end{pmatrix} \succeq 0 \quad (13)$$

Finally replacing the two equality quadratic constraints in (11) with the relaxed constraints (12) and (13) and relaxing the integer constraints over  $\boldsymbol{\eta}$  into continuous constraints  $0 \leq \boldsymbol{\eta} \leq 1$ , we obtain a relaxed optimization problem

$$\begin{aligned} \min_{\boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \epsilon, \delta, g, h} \quad & \delta & (14) \\ \text{s.t.} \quad & \begin{pmatrix} K \circ \mathbf{y}\mathbf{y}^\top & (\mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\lambda} + \epsilon\mathbf{y}) \\ (\mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\lambda} + \epsilon\mathbf{y})^\top & 2\delta - C(g - h - \boldsymbol{\eta}^\top \mathbf{e} - 2\boldsymbol{\lambda}^\top \mathbf{u}) \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} I & (\boldsymbol{\lambda}_{1:T_p} + \boldsymbol{\eta}) \\ (\boldsymbol{\lambda}_{1:T_p} + \boldsymbol{\eta})^\top & g \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} I & \boldsymbol{\lambda}_{1:T_p} \\ \boldsymbol{\lambda}_{1:T_p}^\top & h \end{pmatrix} \succeq 0 \\ & \boldsymbol{\mu} \geq 0, \quad \boldsymbol{\lambda} \geq 0 \\ & 0 \leq \boldsymbol{\eta} \leq 1, \quad \mathbf{A}\boldsymbol{\eta} \geq \mathbf{e} \end{aligned}$$

The problem in (14) is a convex optimization problem, more specifically, a semidefinite programming problem. It has  $\mathcal{O}(T)$  free parameter in the SDP cone and  $\mathcal{O}(T)$  linear inequality constraints, that involves a worst-case computational complexity of  $\mathcal{O}(T^{4.5})$ . It is much more efficient than the SDP problems formulated for semi-supervised support vector machines such as in [14] which has  $\mathcal{O}(T^2)$  free parameter in the SDP cone. Our SDP problem can be efficiently solved by using an interior-point method [13] implemented in some optimization packages, such as SeDuMi [15]. In our experiments, we used the Yalmip interface [16] together with the optimization engine of SeDuMi to solve this semidefinite programming problem.

After the training process, we obtain continuous optimal  $\boldsymbol{\eta}^*$  values. We then use a heuristic rounding procedure to recover the discrete binary values  $\hat{\boldsymbol{\eta}}$  by enforcing the constraints  $\mathbf{A}\hat{\boldsymbol{\eta}} \geq \mathbf{e}$  while minimizing  $\boldsymbol{\lambda}_{1:T_p}^{*\top} \hat{\boldsymbol{\eta}}$ . (See the objective function of (9).) After recovering the discrete  $\hat{\boldsymbol{\eta}}$  values, the target maximum margin discriminant function can be learned by solving the optimization problem (4) or its dual (5).

## 5 Experimental Results

We have conducted experiments on various data sets to evaluate the proposed semidefinite programming approach, comparing with the two maximum margin

approaches, mi-SVM and MI-SVM, proposed in [5]. In our experiments, in order to reduce the ambiguity of the problem, we used equality constraints  $\mathbf{A}\boldsymbol{\eta} = \mathbf{e}$  instead of the inequality constraints for the proposed SDP. This implies that we only select one most promising positive instance from each positive bag. The  $C$  parameters used for each approach are selected based on results obtained using one random training/test split of the data.

### 5.1 Musk Data Sets

We first conducted experiments using the benchmark Musk data sets for multiple-instance learning, Musk1 and Musk2. The Musk data sets are produced for the task of drug activity prediction, and have been described in detail in [1]. The two data sets, Musk1 and Musk2, consist of instances describing different conformations of various molecules. A bag is defined as a set of all conformations for one molecule. A positive bag has at least one instance, that is one conformation of the molecule, that can bind well to a target protein.

We conducted experiments by randomly selecting 4/5 of the bags in Musk1 (1/8 in Musk2) as training data and keeping the remaining as test data. The experiments were repeated 10 times and the average test accuracies are reported in Table 1. One can see that the proposed SDP approach returns the best result among the three methods on data set Musk2. However, on Musk1, mi-SVM gives the best accuracy value and the bag-level method, MI-SVM, gives the weakest result. This might indict that on Musk1 data set it is helpful to incorporate more instances to build the classification model.

**Table 1.** Classification accuracy results on the Musk data sets(%)

Data Set	#Bags	#Data	SDP	MI-SVM	mi-SVM
Musk1	92	476	69.5	69.0	<b>71.6</b>
Musk2	102	6598	<b>61.3</b>	58.9	59.7

### 5.2 Corel Image Data Sets

We have also conducted experiments on the corel image data sets used in [5]. Here an image is viewed as a bag, which consists of a set of instances, that is segments, characterized by color, texture and shape descriptors. We used the three data sets constructed in [5]: Elephant, Fox and Tiger. The problem is to determine whether a given animal is present in an image. For each data set, we conducted experiments by randomly sampling 3/5 of the bags as training data and keeping the remaining as test data. The test accuracy results reported in Table 2 are averages over 10 repeated runs. In this case, the proposed SDP approach outperforms the other two methods on both Fox and Tiger data sets. However, MI-SVM presents a better test accuracy than SDP on the Elephant data set.

**Table 2.** Classification accuracy results on the Corel data sets(%)

Data Set	#Bags	#Data	SDP	MI-SVM	mi-SVM
Elephant	200	1391	74.8	<b>76.7</b>	70.8
Fox	200	1320	<b>56.8</b>	52.3	55.0
Tiger	200	1220	<b>73.6</b>	71.9	69.4

### 5.3 Text Categorization

Finally, we conducted experiments for text categorization using the text data sets generated from the publicly available TREC9 data set in [5]. In a multiple-instance learning setting, each document of the data set corresponds to a bag, where the instances in the bag are overlapping passages splitted from the document, consisting of 50 words in length. For each data set, we randomly sampled 1/3 of the data as training set and kept the remaining as test set. We repeated this process 10 times and the average results are reported in Table 3. Evidently here the SDP approach presents a more consistent advantage over the other two SVM methods. On the 7 data sets, the SDP has only been slightly overperformed by mi-SVM on TREC3. These results suggest that better performance can be gained by pursuing convex global optimization.

**Table 3.** Classification accuracy results on the TREC9 text sets(%)

Data Set	#Bags	#Data	SDP	MI-SVM	mi-SVM
TREC1	400	3224	<b>92.7</b>	92.5	85.8
TREC2	400	3344	<b>75.1</b>	74.4	63.4
TREC3	400	3246	74.3	73.2	<b>74.6</b>
TREC4	400	3391	<b>77.7</b>	76.9	72.8
TREC7	400	3367	<b>72.5</b>	70.9	63.8
TREC9	400	3300	<b>59.9</b>	55.0	59.0
TREC10	400	3453	<b>74.4</b>	73.8	67.8

## 6 Conclusion and Future Work

We have presented a novel maximum margin semidefinite programming approach for multiple-instance learning. Comparing to two other maximum margin approaches based on heuristic procedures that suffer from local optima, our convex approach can be solved using a global optimization method. Unlike the semi-supervised SDP method proposed in the literature which has  $\mathcal{O}(T^2)$  parameters, our SDP has only  $\mathcal{O}(T)$  parameters and can be solved more efficiently. The empirical results reported in the experimental section suggest that the proposed SDP can yield more promising results than the two locally optimized nonconvex maximum margin methods.

Although the SDP approach proposed in this paper selects only positive instances from positive bags to use, it is still an instance-level method. Extending it to get a bag-level approach is an interesting future work we are considering.

## References

1. Dietterich, T., Lathrop, R., Lozano-Perez, T.: Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence Journal* **89** (1997)
2. Carson, C., Thomas, M., Belongie, S., Hellerstein, J., Malik, J.: Blobworld: A system for region-based image indexing and retrieval. In: *Proceedings of the 3rd International Conference on Visual Information and Information Systems*. (1999)
3. Maron, O., Ratan, A.: Multiple-instance learning for natural scene classification. In: *Proceedings of the International Conference on Machine Learning*. (1998)
4. Zhang, Q., Goldman, S., Yu, W., Fritts, J.: Content-based image retrieval using multiple-instance learning. In: *Proceedings of the International Conference on Machine Learning*. (2002)
5. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: *Advances in Neural Information Processing Systems*. (2002)
6. Zhang, Q., Goldman, S.: EM-dd: An improved multiple-instance learning technique. In: *Advances in Neural Information Processing Systems*. (2001)
7. Gartner, T., Flach, P., Kowalczyk, A., Smola, A.: Multi-instance kernels. In: *Proceedings of the International Conference on Machine Learning*. (2002)
8. Mangasarian, O., Wild, E.: Multiple instance classification via successive linear programming. *Journal of Optimization Theory and Applications* **137** (2008)
9. Zhou, Z., Zhang, M.: Ensembles of multi-instance learners. In: *Proceedings of the European Conference on Machine Learning*. (2003)
10. Andrews, S., Hofmann, T.: Multiple instance learning via disjunctive programming boosting. In: *Advances in Neural Information Processing Systems*. (2003)
11. Zhou, Z., Sun, Y., Li, Y.: Multiple-instance learning by training instances as non-i.i.d. samples. In: *Proceedings of the International Conference on Machine Learning*. (2009)
12. Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L., Jordan, M.: Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research* **5** (2004)
13. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge U. Press (2004)
14. Xu, L., Neufeld, J., Larson, B., Schuurmans, D.: Maximum margin clustering. In: *Advances in Neural Information Processing Systems*. (2004)
15. Sturm, J.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software* **11** (1999)
16. Lofberg, J.: YALMIP: a toolbox for modeling and optimization in MATLAB. *Proceedings of the CACSD Conference* (2004)