Applied Numerical Mathematics 60 (2010) 358–369

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

Multiple instance learning via margin maximization

O. Erhun Kundakcioglu [a], Onur Seref [b], Panos M. Pardalos [a,c,d,e,*]

[a] Department of Industrial and Systems Engineering, University of Florida, 303 Weil Hall, PO Box 116595, Gainesville, FL 32611, USA
[b] Department of Business Information Technology, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA
[c] Center for Applied Optimization, University of Florida, 401 Weil Hall, Gainesville, FL 32611, USA
[d] J. Crayton Pruitt Family Department of Biomedical Engineering, University of Florida, 130 BME Building, PO Box 116131, Gainesville, FL 32611, USA
[e] McKnight Brain Institute, University of Florida, Gainesville, FL 32611, USA

ARTICLE INFO

Article history:
Received 25 February 2009
Accepted 27 May 2009
Available online 2 June 2009

Keywords:
Multiple instance learning
Support vector machines
Branch and bound
Pattern classification
Object recognition
Drug activity prediction

ABSTRACT

In this paper, we consider the classification problem within the multiple instance learning (MIL) context. Training data is composed of labeled bags of instances. Despite the large number of margin maximization based classification methods, there are only a few methods that consider the margin for MIL problems in the literature. We first formulate a combinatorial margin maximization problem for multiple instance classification and prove that it is $\mathcal{NP}$-hard. We present a way to apply the kernel trick in this formulation for classifying nonlinear multiple instance data. We also propose a branch and bound algorithm and present computational results on publicly available benchmark data sets. Our approach outperforms a leading commercial solver in terms of the best integer solution and optimality gap in the majority of image annotation and molecular activity prediction test cases.

Published by Elsevier B.V. on behalf of IMACS.

1. Introduction

Multiple instance learning (MIL) is a supervised machine learning problem. In an MIL problem, instances are considered to be contained in bags and actual instance labels are not available. A bag is classified as a *positive bag* if one or more instances in that bag are positive, otherwise it is classified as a *negative bag*. MIL concept is first introduced in a drug activity prediction problem. In this problem, a molecule has the desired drug effect if at least one of its conformations binds to the target, and no effect is observed otherwise. The goal for the learning algorithm is to identify which conformations (instances) are binding (actual positive) using labeled molecule (bag) information. To generalize this concept, an MIL algorithm takes a training set of labeled bags as input and finds a hypothesis that correctly classifies the bags in the training set, and also predicts the labels of bags whose labels are unknown. MIL has numerous successful implementations in a number of application areas such as drug design [15,10], hard drive failure prediction [18], text categorization [4], and content-based image retrieval [5,20,7,8].

There is an array of methods proposed for the MIL problem, most of which are hybrids of other well-known methods. A combination of lazy learning and Hausdorff distance is used for the MIL problem in [25] with two extensions of k-nearest neighbor (k-NN) algorithm and applications on the drug discovery benchmark data. EM-DD technique, which combines expectation maximization (EM) with the diverse density (DD) algorithm, is proposed in [27]. EM-DD is relatively insensitive to the number of features and scales up well to large bag sizes. In [11], extensions of k-NN, citation-kNN, and DD algorithm are proposed with applications to boolean and real valued data.

* Corresponding author at: Department of Industrial and Systems Engineering, University of Florida, 303 Weil Hall, PO Box 116595, Gainesville, FL 32611, USA.
E-mail address: pardalos@ufl.edu (P.M. Pardalos).

0168-9274/$30.00 Published by Elsevier B.V. on behalf of IMACS.
doi:10.1016/j.apnum.2009.05.013

Margin maximization is the fundamental concept in support vector machine (SVM) classifiers, which is shown to minimize the bound on the generalization error [23]. An increasing number of methods that involve SVMs have been proposed to solve MIL problems. A generalization of SVM for MIL is introduced in [1]. This method is based on a heuristic that iteratively changes the labels of instances in positive bags and uses standard SVM formulation, until a local optimal solution is found. A novel automatic image annotation system that integrates an MIL-based SVM formulation together with a global-feature-based SVM is proposed in [20]. For region-based image categorization, a combination of DD and SVM is used in [7]. In this method, a DD function is used to create instance prototypes that represent the instances which are more likely to belong to a bag with a specific label. Instance prototypes are classified using a standard SVM formulation. In [6], an instance similarity measure is used to map bags to a feature space. This method lifts the requirement for the existence of at least one positive instance to label a positive bag and uses a 1-norm SVM to eliminate redundant and irrelevant features. A formulation with linear objective and bilinear constraints is proposed to solve multiple instance classification problems in [17]. Bilinear constraints are handled by an alternating method that uses successive fast linear programs that converge to a local solution in a few iterations. The linear classifiers found by this method are substantially sparse.

Recently, a fast training algorithm, MIL-boost, is proposed to detect objects in images [24]. This method combines a cascade detector method optimized for MIL within a boost framework. A Bayesian MIL method is introduced in [21], which automatically identifies relevant features and uses inductive transfer to learn multiple classifiers. In [12], a method that uses a convex hull representation of multiple instances is shown to perform significantly faster and better on unbalanced data with few positive bags and very large number of negative bags. The convex hull framework applies to most hyperplane based MIL methods.

This paper mainly focuses on the maximal margin classifiers for MIL. Our goal is to find a hyperplane that maximizes the margin between a selection of instances from each positive bag and all of the instances from negative bags. The formulation proposed for the selection of actual positive instances renders this problem to be $\mathcal{NP}$-hard. A generalization of this formulation is proposed in [22], where the selection concept applies to both positive and negative instances. This selective learning method is used to classify neural time-series data. Another similar formulation is introduced within a new supervised learning problem that involves aggregate outputs for training [19]. Our main contribution in this study is to introduce the margin maximization formulation and its dual for multiple instance classification, discuss the complexity of the problem and propose a branch and bound algorithm to solve the problem.

The remainder of this paper is organized as follows: Section 2 presents the mathematical formulation with some insights regarding the kernel trick and demonstrates $\mathcal{NP}$-hardness of margin maximization for multiple instance data. Section 3 gives the implementation details of our solution approach and Section 4 presents the computational results. In Section 5, we provide concluding remarks and directions for future work on this class of problems.

2. Margin maximization for multiple instance data

In this section, we review the fundamental SVM classifiers and extend the margin maximization formulation for multiple instance data.

Let $\mathbf{X}$ be a set of pattern vectors $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, n$, with class labels $y_i \in \{1, -1\}$. The problem of classifying these pattern vectors is finding a function $f(\cdot)$ which correctly assigns a class label for a given pattern vector $\mathbf{x}$. Assume that we want to separate the pattern vectors in positive and negative classes by a hyperplane (ψ, b) where $\psi \in \mathbb{R}^d$ and $b \in \mathbb{R}$. Let the distance between the hyperplane and the closest pattern vector $\mathbf{x}^*$ be $\gamma = |\langle \psi, \mathbf{x}^* \rangle + b|$, which is called the *functional margin*. Our goal is to maximize the *geometric margin*, which is the functional margin of a normalized weight vector, on both sides of the hyperplane. Alternatively, the functional margin can be fixed to 1 on both sides[1] and the norm of the weight vector can be minimized to obtain

$$\min_{\psi, b, \xi} \ \frac{1}{2}\|\psi\|^2 + \frac{C}{2}\sum_{i=1}^{n} \xi_i^2 \tag{1a}$$

$$\text{s.t.} \quad y_i\big(\langle \psi, \mathbf{x}_i \rangle + b\big) \geqslant 1 - \xi_i, \quad i = 1, \ldots, n. \tag{1b}$$

In this formulation, ξ is the slack variable for misclassified pattern vectors and C is the penalty term in the objective function for such vectors. The role of scalar C is to control the trade-off between margin violation and regularization. This formulation is called "2-norm soft margin" since the 2-norm of the margin slack vector is penalized in the objective function. The phrase "soft margin" implies that misclassifications are allowed.

This formulation can be used for nonlinear classification by taking its Lagrangian dual and implementing kernel methods [23,9]. The Lagrangian dual problem of (1) can be written as

$$\max_{\alpha} \ \sum_{i=1}^{n} \alpha_i - \frac{1}{2C}\sum_{i=1}^{n} \alpha_i^2 - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{2a}$$

[1] Hyperplanes with functional margin 1 are sometimes referred to as canonical hyperplanes.

$$\text{s.t.} \quad \sum_{i=1}^{n} y_i \alpha_i = 0, \tag{2b}$$

$$\alpha_i \geqslant 0, \quad i = 1, \ldots, n, \tag{2c}$$

where α_i are the dual variables associated with constraints (1b).

The main significance of the dual formulation is that nonlinear maps can be used to embed the pattern vectors in a higher dimensional space in such a way that a hyperplane can separate the mapped pattern vectors in the embedded space. This embedding is done via the *kernel trick*. The mapping is defined over dot product Hilbert spaces. This transformation is done by replacing the dot product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$, with a nonlinear kernel $K(\mathbf{x}_i, \mathbf{x}_j)$. One of the most commonly used kernels is the Gaussian kernel, which is given as

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{ \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma} \right\}, \tag{3}$$

where parameter σ is usually referred to as the *bandwidth*.

SVMs are based on the theory of linear classifiers, more precisely the idea of the maximum margin hyperplane. Next, we introduce a generalization of the above formulation for the multiple instance classification problem.

2.1. Problem formulation for classification of multiple instance data

The formal definition of MIL setting in the context of classification is as follows: Given a set of patterns $\mathbf{x}_1, \ldots, \mathbf{x}_n$ that are grouped into bags $\mathbf{X}_1, \ldots, \mathbf{X}_m$ with $\mathbf{X}_j = \{\mathbf{x}_i \colon i \in I_j\}$, $I_j \subseteq \{1, \ldots, n\}$, and $\bigcup_j I_j = \{1, \ldots, n\}$; each bag $\mathbf{X}_j$ is associated with a label $y_j \in \{1, -1\}$. These labels are interpreted in the following way: "*If a bag has a negative label, then all patterns in that bag inherit the negative label. On the other hand, if a bag has a positive label, then at least one pattern in that bag is a positive example of the underlying concept.*"

Based on this definition, the maximum margin formulation can be generalized as the following Mixed 0–1 Quadratic Programming problem.

$$\min_{\psi, b, \xi, \eta} \quad \frac{1}{2}\|\psi\|^2 + \frac{C}{2}\sum_{i=1}^{n} \xi_i^2 \tag{4a}$$

$$\text{s.t.} \quad \langle \psi, \mathbf{x}_i \rangle + b \geqslant 1 - \xi_i - M(1 - \eta_i), \quad i \in I^+, \tag{4b}$$

$$-\langle \psi, \mathbf{x}_i \rangle - b \geqslant 1 - \xi_i, \quad i \in I^-, \tag{4c}$$

$$\sum_{i \in I_j} \eta_i \geqslant 1, \quad j \in J^+, \tag{4d}$$

$$\eta_i \in \{0, 1\}, \quad i \in I^+. \tag{4e}$$

In this formulation, $I^+ = \{i \colon i \in I_j \wedge y_j = 1\}$, $I^- = \{i \colon i \in I_j \wedge y_j = -1\}$, and $J^+ = \{j \colon y_j = 1\}$. Note that, M is a sufficiently large number that ensures that the corresponding constraint is active if and only if $\eta_i = 1$. η_i is a binary variable that is 1 if i-th instance is one of the actual positive examples of its bag.

Next, we show the application of *kernel trick* for nonlinear multiple instance classification. In order to apply the kernel trick, the dot products of the input patterns are needed. We rewrite formulation (4) as

$$\min_{\substack{\eta \\ \sum_{i \in I_j} \eta_i \geqslant 1 \\ \eta_i \in \{0,1\}}} \quad \min_{\psi, b, \xi} \quad \frac{1}{2}\|\psi\|^2 + \frac{C}{2}\sum_{i=1}^{n} \xi_i^2 \tag{5a}$$

$$\text{s.t.} \quad \langle \psi, \mathbf{x}_i \rangle + b \geqslant 1 - \xi_i - M(1 - \eta_i), \quad i \in I^+, \tag{5b}$$

$$-\langle \psi, \mathbf{x}_i \rangle - b \geqslant 1 - \xi_i, \quad i \in I^-. \tag{5c}$$

In this formulation, the outer minimization sets the binary variables, and the inner minimization solves regular 2-norm soft margin problem based on these binary values. Therefore we can write the Lagrangian function for the inner minimization as

$$\begin{aligned} L(\psi, b, \xi, \alpha) = {} & \frac{1}{2}\|\psi\|^2 + \frac{C}{2}\sum_{i=1}^{n} \xi_i^2 - \sum_{i \in I^-} \alpha_i \left[-\langle \psi, \mathbf{x}_i \rangle - b - 1 + \xi_i \right] \\ & - \sum_{i \in I^+} \alpha_i \left[\langle \psi, \mathbf{x}_i \rangle + b - 1 + \xi_i + M(1 - \eta_i) \right]. \end{aligned} \tag{6}$$

Differentiating L with respect to the primal variables ψ, b, and ξ, and using stationarity, we obtain

$$\frac{\partial L}{\partial \psi} = \psi - \sum_{i=1}^{n} y_i \alpha_i \mathbf{x}_i = 0, \tag{7a}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} y_i \alpha_i = 0, \tag{7b}$$

$$\frac{\partial L}{\partial \xi_i} = C\xi_i - \alpha_i = 0. \tag{7c}$$

We can substitute the expressions in (7) back in the Lagrangian function to obtain the dual formulation, which will give a maximization problem inside the minimization problem [16]. Instead, we substitute the conditions (7) inside (5) directly:

$$\min_{\substack{\eta \\ \sum_{i\in I_j}\eta_i \geqslant 1 \\ \eta_i \in \{0,1\}}} \min_{\alpha,b} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \frac{1}{2C}\sum_{i=1}^{n} \alpha_i^2 \tag{8a}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} y_j \alpha_j \langle \mathbf{x}_j, \mathbf{x}_i \rangle + b \geqslant 1 - \frac{\alpha_i}{C} - M(1-\eta_i), \quad i \in I^+, \tag{8b}$$

$$-\sum_{j=1}^{n} y_j \alpha_j \langle \mathbf{x}_j, \mathbf{x}_i \rangle - b \geqslant 1 - \frac{\alpha_i}{C}, \quad i \in I^-, \tag{8c}$$

$$\sum_{i=1}^{n} y_i \alpha_i = 0, \tag{8d}$$

$$\alpha_i \geqslant 0, \quad i = 1, \ldots, n. \tag{8e}$$

We finalize the discussion by applying the *kernel trick* on (8) and the resulting formulation is

$$\min_{\alpha,b,\eta} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{2C}\sum_{i=1}^{n} \alpha_i^2, \tag{9a}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b \geqslant 1 - \frac{\alpha_i}{C} - M(1-\eta_i), \quad i \in I^+, \tag{9b}$$

$$-\sum_{j=1}^{n} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) - b \geqslant 1 - \frac{\alpha_i}{C}, \quad i \in I^-, \tag{9c}$$

$$\sum_{i=1}^{n} y_i \alpha_i = 0, \tag{9d}$$

$$\sum_{i\in I_j} \eta_i \geqslant 1, \quad j \in J^+, \tag{9e}$$

$$\eta_i \in \{0,1\}, \quad i \in I^+, \tag{9f}$$

$$\alpha_i \geqslant 0, \quad i = 1, \ldots, n. \tag{9g}$$

Note that, from Karush–Kuhn–Tucker complementarity conditions, the constraints in the primal problem are binding for those with corresponding dual variable $\alpha_i^* > 0$. It should also be noted that $\eta_i^* = 0$ implies that $\alpha_i^* = 0$ since the corresponding constraint cannot be binding. Knowing α^*, we can derive b^* using any $\alpha_i^* > 0$ and (9b) or (9c).

Next we present the complexity results on margin maximization for multiple instance data.

2.2. Complexity of the problem

MIL setting is similar to the concept of *selective learning* introduced in [22]. Selective learning is originally developed to efficiently solve a time series alignment problem in neural data. However, the problem definition in selective learning is slightly different; the patterns are chosen from each positive and negative set in such a way that the margin between the selected positive and negative pattern vectors is maximized. Selective learning, which is a generalization of MIL,[2] is proved

[2] Multiple instance learning is a special case of selective learning where all negative bags are of size 1 (i.e., no selection is performed over negative bags).

to be $\mathcal{NP}$-hard [22]. However, this is not enough to prove the complexity of MIL. To the best of our knowledge, there is no formal proof on the complexity of classifying multiple instance data and this section intends to fill this gap.

It is clear that for sufficiently high penalty C, formulation (4) provides a separating hyperplane where $\xi_i = 0$, $i = 1, \ldots, n$, if data is linearly separable. Therefore, the decision version of the optimization problem in (4) is defined as follows:

Multiple Instance Learning Decision (MILD) problem. Given a set of d-dimensional patterns $\mathbf{x}_1, \ldots, \mathbf{x}_n$ that are grouped into bags $\mathbf{X}_1, \ldots, \mathbf{X}_m$ with $\mathbf{X}_j = \{\mathbf{x}_i\colon\ i \in I_j\}$, $I_j \subseteq \{1, \ldots, n\}$, and $\bigcup_j I_j = \{1, \ldots, n\}$; each bag $\mathbf{X}_j$ is associated with a label $y_j \in \{1, -1\}$. Is there a selection of at least one instance from each positive labeled bag such that all vectors with negative labels can be separated from selected positive instances with no misclassification by a hyperplane (ψ, b) that satisfies $\frac{1}{2}\|\psi\|^2 \leqslant n$?

Theorem 2.1. *MILD is $\mathcal{NP}$-complete for bags of size at least 2.*

Proof. We show that MILD is $\mathcal{NP}$-complete for bags of size at least 2 by a reduction from the classical PARTITION problem.

The classical PARTITION problem is described as follows: Given a set of positive integers $S = \{s_1, s_2, \ldots, s_n\}$, does there exist a subset $S' \subseteq S$ such that

$$\sum_{i\colon s_i \in S'} s_i = \sum_{i\colon s_i \in S \setminus S'} s_i = \frac{1}{2}\sum_{i=1}^{n} s_i? \tag{10}$$

This problem is known to be $\mathcal{NP}$-complete [13]. Next, we consider the following variant of the PARTITION problem. Given a set of n positive integers $S = \{s_1, s_2, \ldots, s_n\}$, does there exist a vector $\psi \in \{-1, +1\}^d$, such that

$$\sum_{i=1}^{n} s_i \psi_i = 0? \tag{11}$$

Suppose we are given an instance of the PARTITION problem. We will add n dummy features and set the dimension of the space $d = 2n$ and construct an instance of the MILD problem as follows:

Let $\mathbf{e}_i$ be a d-dimensional vector whose components are zero except component i, which is equal to 1.

(i) Add the pattern $(s_1, s_2, \ldots, s_n, 1, 0, \ldots, 0)^T$ with positive label.
(ii) Add the pattern $(s_1, s_2, \ldots, s_n, -1, 0, \ldots, 0)^T$ with negative label.
(iii) Add patterns $\mathbf{e}_{n+1}, \mathbf{e}_{n+2}, \ldots, \mathbf{e}_{2n}$ with positive labels.
(iv) Add patterns $-\mathbf{e}_{n+1}, -\mathbf{e}_{n+2}, \ldots, -\mathbf{e}_{2n}$ with negative labels.
(v) Add n bags with positive labels where bag i consists of patterns $\mathbf{e}_i$ and $-\mathbf{e}_i$ for $i = 1, \ldots, n$.

After this reduction, the corresponding inequalities in (4) become

$$\sum_{i=1}^{n} s_i \psi_i + \psi_{n+1} + b \geqslant 1, \tag{12a}$$

$$-\sum_{i=1}^{n} s_i \psi_i + \psi_{n+1} - b \geqslant 1, \tag{12b}$$

$$\psi_i + b \geqslant 1, \quad i = n+1, \ldots, 2n, \tag{12c}$$

$$\psi_i - b \geqslant 1, \quad i = n+1, \ldots, 2n, \tag{12d}$$

$$(\psi_i + b \geqslant 1) \quad \text{OR} \quad (-\psi_i + b \geqslant 1), \quad i = 1, \ldots, n. \tag{12e}$$

Note that, C is a sufficiently large number and a hyperplane that has the maximum interclass margin with $\xi_i = 0$, $i = 1, \ldots, n$, is desired.

Let us assert that $b = 0$ and prove the constraints in (12) ensure a YES answer for MILD if and only if PARTITION has a YES answer.

It is apparent from (12c) and (12d) that $\psi_i = 1$, $i = n+1, \ldots, 2n$, and from (12e) that $\psi_i \in \{-1, +1\}$, $i = 1, \ldots, n$, since the goal is to minimize $\|\psi\|^2$ and satisfy $\frac{1}{2}\|\psi\|^2 \leqslant n$. Using this fact with (12a), (12b), the answer for MILD is YES if and only if $\sum_{i=1}^{n} s_i \psi_i = 0$ (i.e., PARTITION has a YES answer).

Next, we prove by contradiction that $b = 0$ in the maximum margin solution. Note that, when $b = 0$, the solution described above is feasible with $\psi_i \in \{-1, 1\}$, $i = 1, \ldots, n$, and $\psi_i = 1$, $i = n+1, \ldots, 2n$, provided that PARTITION has a YES answer. This separation gives an objective function of n. Assume that there is a better solution with $b = \delta \neq 0$. Then (12c), (12d) force $\psi_i \geqslant 1 + |\delta|$, $i = n+1, \ldots, 2n$, and (12e) forces $|\psi_i| \geqslant 1 - |\delta|$, $i = 1, \ldots, n$. Even if (12a), (12b) are ignored, the objective function value is at least $n + n|\delta|^2$ which is strictly more than n, thus a worse solution and a contradiction.

The presented reduction is polynomial. Hence MILD is $\mathcal{NP}$-complete for bags of size at least 2. □

Corollary 2.2. *Maximum margin formulation for MIL (i.e., formulation (4)) is $\mathcal{NP}$-hard for bags of size at least 2.*

Next, we prove a stronger complexity result for a special case of the problem.

Theorem 2.3. *MILD is strongly $\mathcal{NP}$-complete for bags of size at least 3.*

Proof. We show that MILD is strongly $\mathcal{NP}$-complete for bags of size at least 3 by a reduction from the classical 3SAT problem.

The classical 3SAT problem is described as follows: Given a collection $C = \{c_1, c_2, \ldots, c_m\}$ of clauses on a finite set U of variables such that $|c_i| = 3$ for $1 \leqslant i \leqslant m$, is there a truth assignment for U that satisfies all the clauses in C?

If u is a variable in U, then u and $\bar{u}$ are *literals* over U. This problem is known to be strongly $\mathcal{NP}$-complete [13].

Suppose we are given an instance of the 3SAT problem. We will set the dimension of the space $d = 2n$ and construct an instance of the MILD problem as follows:

Note that, $\mathbf{e}_i$ is a d-dimensional vector whose components are zeros except for component i, which is equal to 1.

(i) Add m bags with positive labels for each clause that consists of vectors $\mathbf{e}_i$ for literals u_i and $-\mathbf{e}_i$ for literals $\bar{u}_i$ in the corresponding clause.
(ii) Add patterns $\mathbf{e}_{n+1}, \mathbf{e}_{n+2}, \ldots, \mathbf{e}_{2n}$ with positive labels.
(iii) Add patterns $-\mathbf{e}_{n+1}, -\mathbf{e}_{n+2}, \ldots, -\mathbf{e}_{2n}$ with negative labels.
(iv) Add n bags with positive labels where bag i consists of patterns $\mathbf{e}_i$ and $-\mathbf{e}_i$ for $i = 1, \ldots, n$.

After this reduction, the corresponding inequalities in (4) become

$$(\gamma_{il}\psi_i + b \geqslant 1) \quad \text{OR} \quad (\gamma_{jl}\psi_j + b \geqslant 1) \quad \text{OR} \quad (\gamma_{kl}\psi_k + b \geqslant 1), \quad l = 1, \ldots, m, \tag{13a}$$

$$\psi_i + b \geqslant 1, \quad i = n+1, \ldots, 2n, \tag{13b}$$

$$\psi_i - b \geqslant 1, \quad i = n+1, \ldots, 2n, \tag{13c}$$

$$(\psi_i + b \geqslant 1) \quad \text{OR} \quad (-\psi_i + b \geqslant 1), \quad i = 1, \ldots, n, \tag{13d}$$

where γ_{il} is 1 if u_i appears in clause c_l, and -1 if $\bar{u}_i$ appears in clause c_l.

Note that, C is a sufficiently large number and a hyperplane that has the maximum interclass margin with $\xi_i = 0$, $i = 1, \ldots, n$, is desired.

Let us assert that $b = 0$ and prove the constraints in (13) ensure a YES answer for MILD if and only if 3SAT has a YES answer.

It is obvious from (13a) that ψ_i are either greater than 1 or less than -1 and the objective of minimizing $\|\psi\|^2$ ensures ψ_i are set to either 1 or -1, respectively. It is easy to see that the answer for 3SAT is YES if and only if, $\psi_i = 1$ for variables that are set to TRUE and $\psi_i = -1$ for those that are FALSE.

Next, we prove by contradiction that $b = 0$ in the maximum margin solution. Assume that there is a better solution with $b = \delta \neq 0$. Then (13b), (13c) force $\psi_i \geqslant 1 + |\delta|$, $i = n+1, \ldots, 2n$, and (13d) forces $|\psi_i| \geqslant 1 - |\delta|$, $i = 1, \ldots, n$. The objective function value is at least $n + n|\delta|^2$ which is strictly more than n, thus a worse solution and a contradiction.

The presented reduction is polynomial. Hence MILD is strongly $\mathcal{NP}$-complete for bags of size at least 3. □

Corollary 2.4. *Maximum margin formulation for MIL (i.e., formulation (4)) is strongly $\mathcal{NP}$-hard for bags of size at least 3.*

Next, we describe our proposed branch and bound scheme that scales up better than a leading commercial solver.

3. A branch and bound algorithm for MIL

A typical way to solve a combinatorial problem is via an enumeration tree where the leaves of the tree correspond to feasible solutions that should be examined in complete enumeration. *Branch and bound algorithm*, which uses an intelligent decomposition of the main problem and bound information on an enumeration tree, can help us solve problems that are impossible to solve using complete enumeration [26].

In a *minimization* problem with binary variables, the problem is decomposed into two problems at each node of the enumeration tree. These decompositions are obtained by branching on a binary variable whose value is not set. For each node, upper and lower bounds are obtained and a node is *pruned* (i.e., no further decomposition is necessary) if one of the following conditions hold.

- The node is infeasible.[3]

[3] Note that, in our branch and bound algorithm, a node cannot be pruned by infeasibility since the decompositions (i.e., soft margin classification problems) are always feasible.

- The upper bound is equal to the lower bound.
- The lower bound is larger than the objective function value of the *incumbent* (i.e., current best) solution.

When the upper bound is equal to the lower bound, a node is *pruned by optimality*, since the optimal solution for this decomposition is known and further decomposition is redundant. A node can also be *pruned by bound*, which implies that it does not suggest a better solution than current best solution.

Upper bounds are obtained from the objective function value of feasible solutions. If a feasible solution is better than the incumbent solution, incumbent is set to that solution. Lower bounds on the other hand, are not necessarily feasible but they give a measure of how promising the decomposition is. Tight bounds lead to more pruning and faster convergence. Good branching strategies are also crucial in a successful branch and bound algorithm. Next, we explore our bounding and branching schemes for MIL problem.

3.1. Branching scheme

We will denote binary variables η_i that are set for a partial solution by η_i^c. At an intermediate step where some binary variables are set, we solve the following convex quadratic problem. This problem is a relaxation of the original problem, thus gives a lower bound. We consider relaxing the binary variable restrictions to avoid the intense computational burden accompanying other relaxations where binary variables are kept.

$$z_{LB} = \min_{\psi,b,\xi,\eta} \frac{1}{2}\|\psi\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 \tag{14a}$$

$$\text{s.t.} \quad \langle \psi, \mathbf{x}_i\rangle + b \geqslant 1 - \xi_i, \quad i \in I^+ \wedge \eta_i^c = 1, \tag{14b}$$

$$\langle \psi, \mathbf{x}_i\rangle + b \geqslant 1 - \xi_i - M(1-\eta_i), \quad i \in I_j \wedge j \in J^0 \wedge \eta_i^c \neq 0, \tag{14c}$$

$$\sum_{i\in I_j \wedge \eta_i^c \neq 0} \eta_i = 1, \quad j \in J^0, \tag{14d}$$

$$0 \leqslant \eta_i \leqslant 1, \quad i \in I_j \wedge j \in J^0 \wedge \eta_i^c \neq 0, \tag{14e}$$

$$-\langle \psi, \mathbf{x}_i\rangle - b \geqslant 1 - \xi_i, \quad i \in I^-, \tag{14f}$$

where J^0 is the set of positive bags whose actual positive instances are not discovered, i.e., $J^0 = \{j\colon\ y_j = 1 \wedge \eta_k^c \neq 1,\ \forall k \in I_j\}$. It is easy to see that when constraint (4d) is changed to equality, the optimal objective function value will not change for (4). On the other hand, selection of exactly one data instance per positive bag will significantly reduce the size of the feasible region. Therefore, we use the equality constraint for our lower bounding formulation (14). When an instance is selected for a decomposition, constraint (14d) will automatically ignore remaining instances that share the same bag, thus avoid redundant computational work.

If the obtained solution is integer feasible (i.e., $\eta_i^* \in \{0,1\},\ \forall i\colon\ i \in I_j \wedge y_j = 1$) then we can prune the node since upper and lower bounds are equal (i.e., the optimal solution for that decomposition is known). However, we observe that without a careful selection of parameter M, the above formulation ignores (14c) by setting $0 < \eta_i^* < 1$ and associated ξ^*'s are set to 0. Therefore, we check the feasibility of the hyperplane for each undecided bag explicitly. Formally, a node of the branch and bound tree is pruned if the following boolean function is satisfied where (ψ^*, b^*) define the optimal hyperplane obtained from (14).

$$\bigwedge_{j\in J^0} \bigvee_{i\colon\ i\in I_j} \langle \psi^*, \mathbf{x}_i\rangle + b^* \geqslant 1. \tag{15}$$

If (15) is not satisfied, then *branching* is performed on η_k where

$$k = \arg\max_{i\colon\ i\in I_{j^0}} \langle \psi^*, \mathbf{x}_i\rangle + b^* \tag{16}$$

and

$$j^0 = \arg\min_{j\in J^0} \max_{i\colon\ i\in I_j} \langle \psi^*, \mathbf{x}_i\rangle + b^*. \tag{17}$$

The problem is decomposed into two subproblems with additional constraints $\eta_k = 1$ and $\eta_k = 0$, respectively. The aim here is to branch on the critical bag I_{j^0} that is currently misclassified or closest to being misclassified based on (ψ^*, b^*). (17) selects the critical bag whereas (16) selects the most promising instance from that bag.

Consider the example in Fig. 1. The algorithm starts by solving the relaxation in (14). There is one (circled) instance in one of the positive bags which should be selected and that solution defines the lower bound. The separating hyperplane for the relaxation is shown as a dotted line. The bag whose best instance is the most misclassified is considered next. Branching is performed on the most promising instance in square. For the first decomposition where the instance in square is selected,

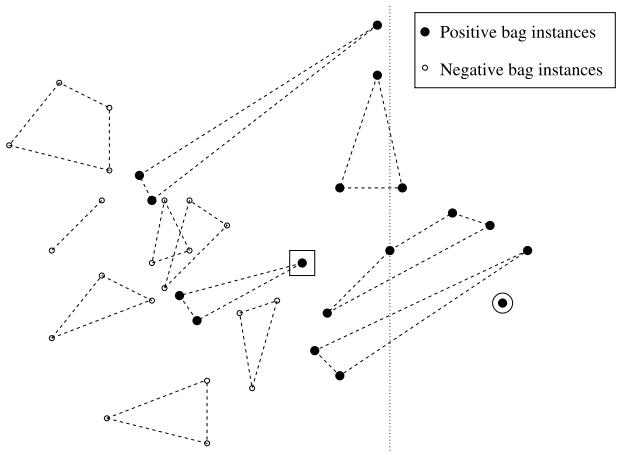

Fig. 1. An example of critical bag.

the corresponding node can be pruned by optimality since (15) is satisfied. When other instances in this bag are considered as actual positive, the lower bounds are larger, thus the optimal solution is obtained. All instances in this bag should be checked in order to conclude optimality if M is too large. In order to achieve optimal solutions quickly, we start with the critical bag and the instances of the critical bag that are promising (i.e., the least misclassified).

3.2. Bounding scheme

To obtain an upper bound, we employ a two phase heuristic approach. In the first step, we find the optimal separating hyperplane considering the previous decisions (i.e., η^c values) and all undecided bags. In the second step, we re-optimize based on a temporary selection of actual positive instances. Formally, the first phase solves the following problem.

$$\min_{\psi,b,\xi} \ \frac{1}{2}\|\psi\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 \tag{18a}$$

$$\text{s.t.} \quad \langle \psi, \mathbf{x}_i\rangle + b \geqslant 1 - \xi_i, \quad i \in I^+ \wedge \eta_i^c = 1, \tag{18b}$$

$$\langle \psi, \mathbf{x}_i\rangle + b \geqslant 1 - \xi_i, \quad i \in I_j \wedge j \in J^0 \wedge \eta_i^c \neq 0, \tag{18c}$$

$$-\langle \psi, \mathbf{x}_i\rangle - b \geqslant 1 - \xi_i, \quad i \in I^-. \tag{18d}$$

For each undecided bag, we select the instance that is furthest away from the optimal hyperplane obtained from (18). Set S of selected instances is defined as

$$S = \left\{ s_j \colon s_j = \arg \max_{i \in I_j \wedge \eta_i^c \neq 0} \left\langle \psi^*, \mathbf{x}_i \right\rangle + b^*, \ j \in J^0 \right\} \tag{19}$$

where (ψ^*, b^*, ξ^*) define the optimal solution for (18).

The second phase computes the upper bound by solving the margin maximization problem based on this temporary selection.

$$z_{UB} = \min_{\psi,b,\xi} \ \frac{1}{2}\|\psi\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 \tag{20a}$$

$$\text{s.t.} \quad \langle \psi, \mathbf{x}_i\rangle + b \geqslant 1 - \xi_i, \quad i \in I^+ \wedge \eta_i^c = 1, \tag{20b}$$

$$\langle \psi, \mathbf{x}_i\rangle + b \geqslant 1 - \xi_i, \quad i \in S, \tag{20c}$$

$$-\langle \psi, \mathbf{x}_i\rangle - b \geqslant 1 - \xi_i, \quad i \in I^-. \tag{20d}$$

Next, we present the computational results and show the performance of our branch and bound algorithm on public data sets.

4. Computational study

To demonstrate the capabilities of our algorithm, we report results on data sets from [3] and [2]. Two data sets from [3] represent the molecular activity prediction data sets. Molecules judged by human experts are labeled as musks or non-musks. The goal for MIL is to discriminate these two categories given the exact shape and conformation of each molecule.

Table 1
Size information for the molecular activity prediction and the image annotation data sets.

Data set	Features (Nonzero)	+ Bags	+ Instances	− Bags	− Instances
Musk1	166	47	207	45	269
Musk2	166	39	1017	63	5581
Elephant	230(143)	100	762	100	629
Fox	230(143)	100	647	100	673
Tiger	230(143)	100	544	100	676

Table 2
Time (in seconds) to achieve the optimal solution for Our Branch and Bound Scheme vs. CPLEX Default Branch and Bound Algorithm for the Image Annotation Data.

	n	$\lvert J^+\rvert$	d	Our B&B Scheme	CPLEX 10.1
ELEPHANT	20	2	10	0.04	0.01
	20	2	5	0.01	0.01
	40	3	10	0.14	0.03
	40	3	5	0.20	0.03
	80	6	10	259.29	1.95
	80	6	5	91.56	3.00
FOX	20	2	10	0.17	0.01
	20	2	5	0.14	0.01
	40	3	10	0.89	0.06
	40	3	5	0.45	0.01
	80	6	10	231.81	9.29
	80	6	5	618.01	86.87
TIGER	20	2	10	0.20	0.01
	20	2	5	0.03	0.01
	40	4	10	0.26	0.01
	40	4	5	0.20	0.05
	80	8	10	265.71	12.18
	80	8	5	399.95	36.23

Three data sets from [2] correspond to an image annotation task where the goal is to determine whether or not a given animal is present in an image. Color images from Corel data set are segmented with Blobworld system. Set of segments in each picture are characterized by color, shape, and texture descriptors. The sizes of these data sets are presented in Table 1.

All computations are performed on a 3.4 GHz Pentium IV desktop computer with 2.0 Gb RAM. The algorithms are implemented in C++ and used in conjunction with MATLAB 7.3 environment in which the data resides. In our algorithm, we solved the convex minimization problems (i.e., formulations (14), (18), and (20)) using CPLEX 10.1 [14]. For benchmarking purposes, formulation (4) is solved using CPLEX 10.1 with default settings. In all experiments, trade-off parameter C between training error and margin is set to $(\sum\langle x, x\rangle/n)^{-1}$, which is scaled based on the input vector.

In our attempt to find the global minimum for formulation (4), we report the best integer solution obtained (i.e., UB), optimality gap (i.e., UB-LB) and solution times instead of the prediction accuracy results for generalization. In cases where an algorithm terminates with optimality in the given timeframe, the lower bound is equal to the upper bound (i.e., incumbent solution), thus zero optimality gap.

In order to show the computational limitations of exact algorithms, all instances are obtained by a random feature and bag selection. Because the number of instances is restricted, the last bag selected might not have all instances from the original data set. The results show that when the number of instances increases, our algorithm outperforms CPLEX in terms of the best objective function value. However, when the number of features increases, there is additional computational task at each node of branch and bound tree that might deteriorate the performance of our implementation. Nevertheless, feature selection can be used to scale the problem whereas the instances are crucial.

Table 2 shows the performance of exact algorithms for small test instances. The computation times to achieve optimal solutions are presented with different data sets and implementations. As seen on this table, CPLEX outperforms our branch and bound scheme in small instances due to its preprocessing power and fast implementation at each node of the tree. Note that, neither our algorithm nor CPLEX is able to solve instances with more than 120 data instances to optimality in 3600 seconds.

Next, we consider larger problem sets. Tables 3 and 4 present benchmark results for our branch and bound implementation and CPLEX default implementation with time limits of 3 and 30 minutes, respectively. In these tests, all instances from the molecular activity prediction data set are used and random feature selection is performed. Number of features selected is denoted by d.

Table 3
Computational Results for Our Branch and Bound Scheme vs. CPLEX Default Branch and Bound Algorithm for the Molecular Activity Prediction Data (Musk1) with 3 minutes time limit.

d	Our B&B Scheme			CPLEX 10.1		
	UB	UB-LB	Time	UB	UB-LB	Time
5	**10 304.05**	10 029.83	180	11 263.03	**9714.50**	180
10	**10 802.55**	**10 801.06**	180	12 259.66	11 082.57	180

Table 4
Computational Results for Our Branch and Bound Scheme vs. CPLEX Default Branch and Bound Algorithm for the Molecular Activity Prediction Data (Musk1) with 30 minutes time limit.

d	Our B&B Scheme			CPLEX 10.1		
	UB	UB-LB	Time	UB	UB-LB	Time
5	**11 876.10**	11 104.30	1800	13 305.71	**10 612.31**	1800
10	**10 178.45**	10 087.73	1800	11 691.09	**9367.82**	1800

Table 5
Computational Results for Our Branch and Bound Scheme vs. CPLEX Default Branch and Bound Algorithm for the Image Annotation Data with 3 minutes time limit.

Data set	n	$\lvert J^+\rvert$	d	Our B& B Scheme			CPLEX 10.1		
				UB	UB-LB	Time	UB	UB-LB	Time
ELEPHANT	400	26	20	**974.12**	**767.45**	180	986.23	787.63	180
	400	26	10	**3065.92**	**3064.59**	180	3425.26	3230.09	180
	400	26	5	**3072.25**	3072.24	180	3305.80	**2915.97**	180
	800	50	20	**3792.26**	**3792.22**	180	4397.07	4295.40	180
	800	50	10	**6272.77**	**6272.77**	180	6757.60	6563.97	180
	800	50	5	**6557.27**	**6557.27**	180	7501.58	7308.48	180
	1200	78	20	**6585.39**	**6585.39**	180	9637.13	9637.13	180
	1200	78	10	**10 062.24**	**10 062.24**	180	11 072.95	11 072.95	180
	1200	78	5	**9952.44**	**9952.44**	180	11 821.95	11 631.28	180
FOX	400	33	20	**3282.89**	3088.27	180	3388.99	**3017.85**	180
	400	33	10	4751.69	4548.62	180	**4578.98**	**3999.80**	180
	400	33	5	**4532.63**	4239.47	180	4558.33	**3977.54**	180
	800	63	20	8792.20	8792.20	180	**8618.59**	**8429.70**	180
	800	63	10	10 216.73	10 050.18	180	**9517.32**	**9321.82**	180
	800	63	5	10 045.32	9878.48	180	**9681.97**	**9485.00**	180
	1200	93	20	**13 034.06**	**13 034.06**	180	15 440.33	15 417.24	180
	1200	93	10	15 395.31	15 395.22	180	**14 486.01**	**14 309.68**	180
	1200	93	5	15 547.77	15 380.59	180	**14 653.29**	**14 456.20**	180
TIGER	400	33	20	1699.07	1699.01	180	**1562.03**	**1484.61**	180
	400	33	10	**2886.13**	2693.32	180	3058.04	**2679.99**	180
	400	33	5	**3287.02**	3093.72	180	3422.92	**3033.86**	180
	800	71	20	**4761.77**	**4761.77**	180	5472.10	5345.19	180
	800	71	10	**6946.20**	**6946.20**	180	7353.32	6953.16	180
	800	71	5	**8519.69**	8519.59	180	8898.57	**8157.97**	180
	1144	100	20	**7480.61**	**7453.07**	180	10 433.51	10 176.86	180
	1144	100	10	**10 522.63**	**10 250.09**	180	12 190.93	11 805.41	180
	1144	100	5	**11 994.36**	**11 605.10**	180	12 774.59	11 997.72	180

Tables 3 and 4 show that our algorithm achieves better solutions than CPLEX in all tests. However, the lower bounds obtained by CPLEX are tighter. Musk2 is not used in our computational studies because only nonlinear classifiers are used on this data set in the literature.

Next, we study the image annotation data. In order to observe how the algorithms scale up, instance selection is performed as well as feature selection. Number of instances is denoted by n and number of positive bags is denoted by $|J^+|$.

Table 5 shows that our algorithm scales up well and obtains generally better solutions than CPLEX for larger problems in 3 minutes. There are cases where CPLEX performs better but in these cases the differences are subtle. Table 6 shows that when the time limit is increased to 30 minutes, our algorithm still achieves better solutions in the majority of tests. There might be cases where the best solution found by an algorithm is optimal but there are active nodes that have lower bounds less than the incumbent solution, therefore optimality is not guaranteed. We do not report the number of remaining active nodes explicitly. However, it should be noted that CPLEX has significantly more number of active nodes than our algorithm on the average. It should also be noted that, lower bounds obtained by CPLEX are generally better than that of our implementation.

Table 6
Computational Results for Our Branch and Bound Scheme vs. CPLEX Default Branch and Bound Algorithm for the Image Annotation Data with 30 minutes time limit.

Data set	n	$\lvert J^+\rvert$	d	Our B&B Scheme			CPLEX 10.1		
				UB	UB-LB	Time	UB	UB-LB	Time
ELEPHANT	400	26	20	711.05	**132.45**	1800	711.05	293.41	1800
	400	26	10	2956.12	2954.31	1800	**2924.76**	**2482.72**	1800
	400	26	5	3037.73	3037.57	1800	**3022.99**	**2442.02**	1800
	800	50	20	**3482.16**	**3482.11**	1800	4379.25	4193.66	1800
	800	50	10	**6272.77**	**6272.58**	1800	6594.63	6397.13	1800
	800	50	5	**6540.22**	**6540.20**	1800	7092.43	6707.46	1800
	1200	78	20	**6585.39**	**6585.39**	1800	7637.07	7470.57	1800
	1200	78	10	**10 062.24**	**10 062.24**	1800	10 564.25	10 370.84	1800
	1200	78	5	**9874.41**	**9874.41**	1800	11 599.74	11 402.11	1800
FOX	400	33	20	3130.62	2919.98	1800	**3008.31**	**2553.51**	1800
	400	33	10	4115.66	3886.59	1800	**4074.98**	**3468.40**	1800
	400	33	5	**4504.43**	4007.91	1800	4543.71	**3773.47**	1800
	800	63	20	**8246.68**	8246.56	1800	8406.36	**8212.65**	1800
	800	63	10	**9121.37**	9121.26	1800	9402.43	**9020.16**	1800
	800	63	5	**9387.90**	9175.31	1800	9539.56	**9154.95**	1800
	1200	93	20	**13 034.06**	**13 034.06**	1800	13 588.41	13 293.90	1800
	1200	93	10	14 532.07	14 531.79	1800	**14 419.72**	**14 222.29**	1800
	1200	93	5	14 849.72	14 650.02	1800	**14 639.85**	**14 246.02**	1800
TIGER	400	33	20	1429.96	1429.68	1800	**1425.15**	**1208.82**	1800
	400	33	10	2785.38	2589.39	1800	**2765.82**	**2061.83**	1800
	400	33	5	**3287.02**	**2971.33**	1800	3381.63	2973.43	1800
	800	71	20	**4705.98**	4705.98	1800	4813.83	**4653.93**	1800
	800	71	10	**6943.98**	6943.83	1800	7156.77	**6530.96**	1800
	800	71	5	**8099.99**	7903.04	1800	8307.19	**7808.33**	1800
	1144	100	20	**7480.61**	7447.46	1800	7973.10	**7347.56**	1800
	1144	100	10	**10 522.63**	10 250.09	1800	11 225.02	**10 107.72**	1800
	1144	100	5	**11 193.67**	**10 803.19**	1800	12 202.86	11 174.82	1800

Table 7
Benchmark results for tests with time limits.

	Best solution		Optimality gap	
	Our B&B	CPLEX 10.1	Our B&B	CPLEX 10.1
#	**43**	15	25	**33**
AVG	**10.13**%	3.19%	**15.32**%	7.07%
BEST	**46.34**%	8.77%	**121.53**%	25.59%

The bags are harder to separate when the number of features decreases. Therefore, the optimality gap with less number of features is usually larger. Tables 5 and 6 show that our algorithm usually finds better solutions than CPLEX despite larger optimality gap.

Table 7 summarizes the results for cases where an optimal solution is not achieved. # denotes the number of tests an algorithm outperforms the other. Average and largest improvements achieved by an algorithm over the other are denoted by AVG and BEST, respectively. As seen on the table, our algorithm achieves significantly better solutions than CPLEX in general. Although optimality gap for CPLEX is smaller than our algorithm in 33 of 58 tests, the average improvement is relatively small. On the other hand, when our algorithm has a smaller optimality gap, the improvement over CPLEX is much more significant.

To sum up, when the number of problem instances is small and number of features is large, CPLEX default implementation can be more suitable because of its preprocessing power. Our algorithm, on the other hand, outperforms CPLEX for practical cases, where number of instances is large and feature selection is applied.

5. Concluding remarks

This paper presents the mathematical formulation, kernel trick application, complexity results, and a branch and bound algorithm for linear classification through margin maximization for multiple instance data. Experimental results show additional benefits of intelligent bounding and branching schemes. Our branch and bound algorithm outperforms a leading commercial solver for practical cases where the number of instances increases. We observe that the proposed heuristic gives tight upper bounds, but the lower bounding scheme needs to be improved. The lower bounding technique we propose helps mostly with pruning by optimality, but rarely with pruning by bound.

An interesting future study might be the selection of M in formulation (4) based on input data. This number should satisfy the selection criteria, but it should be small enough to have tight lower bounds with the relaxations as well. Alternatively, M selection can be avoided by an alternative formulation and a performance benchmark for different formulations can be investigated.

Acknowledgements

The authors thank J. Cole Smith for his discussions. This work is partially supported by NSF and Air Force grants.

References

[1] S. Andrews, T. Hofmann, I. Tsochantaridis, Multiple instance learning with generalized support vector machines, in: Eighteenth National Conference on Artificial Intelligence, American Association for Artificial Intelligence, Menlo Park, CA, USA, 2002, pp. 943–944.
[2] S. Andrews, I. Tsochantaridis, T. Hofmann, Support vector machines for multiple-instance learning, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Neural Information Processing Systems, in: Advances in Neural Information Processing Systems, vol. 15, MIT Press, Vancouver, British Columbia, Canada, 2003, pp. 561–568.
[3] A. Asuncion, D.J. Newman, UCI machine learning repository. URL http://mlearn.ics.uci.edu/, 2007.
[4] T. Brow, B. Settles, M. Craven, Classifying biomedical articles by making localized decisions, in: Proceedings of the Fourteenth Text Retrieval Conference (TREC), 2005.
[5] G. Carneiro, A.B. Chan, P.J. Moreno, N. Vasconcelos, Supervised learning of semantic classes for image annotation and retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (3) (2007) 394–410.
[6] Y. Chen, J. Bi, J.Z. Wang, MILES: Multiple-instance learning via embedded instance selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (12) (2006) 1931–1947.
[7] Y. Chen, J.Z. Wang, Image categorization by learning and reasoning with regions, Journal of Machine Learning Research 5 (2004) 913–939.
[8] S.C. Chuang, Y.Y. Xu, H.-C. Fu, Neural network based image retrieval with multiple instance learning techniques, in: Lecture Notes in Computer Science, vol. 3682, Springer, Berlin, Heidelberg, 2005, pp. 1210–1216.
[9] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press, Cambridge, UK, 2000.
[10] T.G. Dietterich, R.H. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, Artificial Intelligence 89 (1997) 31–71.
[11] D.R. Dooly, Q. Zhang, S.A. Goldman, R.A. Amar, Multiple-instance learning of real-valued data, Journal of Machine Learning Research 3 (2002) 651–678.
[12] G. Fung, M. Dundar, B. Krishnapuram, R.B. Rao, Multiple instance learning for computer aided diagnosis, in: B. Schölkopf, J. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems, vol. 19, MIT Press, Vancouver, British Columbia, Canada, 2007, pp. 425–432.
[13] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, 1979.
[14] ILOG, CPLEX 10.1, Incline Village, Nevada. URL http://www.ilog.com/products/cplex/, 2008.
[15] A.N. Jain, T.G. Dietterich, R.H. Lathrop, D. Chapman, R.E. Critchlow, B.E. Bauer, T.A. Webster, T. Lozano-Perez, A shape-based machine learning tool for drug design, Journal of Computer-Aided Molecular Design 8 (6) (1994) 635–652.
[16] O.L. Mangasarian, Nonlinear Programming, SIAM, Philadelphia, 1994.
[17] O.L. Mangasarian, E.W. Wild, Multiple instance classification via successive linear programming, Journal of Optimization Theory and Applications 137 (3) (2008) 555–568.
[18] J.F. Murray, G.F. Hughes, K. Kreutz-Delgado, Machine learning methods for predicting failures in hard drives: A multiple-instance application, The Journal of Machine Learning Research 6 (2005) 783–816.
[19] D.R. Musicant, J.M. Christensen, J.F. Olson, Supervised learning by training on aggregate outputs, in: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, ICDM '07, IEEE Computer Society, Washington, DC, USA, 2007, pp. 252–261.
[20] X. Qi, Y. Han, Incorporating multiple SVMs for automatic image annotation, Pattern Recognition 40 (2007) 728–741.
[21] V.C. Raykar, B. Krishnapuram, J. Bi, M. Dundar, R.B. Rao, Bayesian multiple instance learning: Automatic feature selection and inductive transfer, in: Proceedings of the 25th International Conference on Machine Learning, ICML '08, ACM, New York, NY, USA, 2008, pp. 808–815.
[22] O. Seref, O.E. Kundakcioglu, O.A. Prokopyev, P.M. Pardalos, Selective support vector machines, Journal of Combinatorial Optimization 17 (1) (2009) 3–20.
[23] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[24] P. Viola, J.C. Platt, C. Zhang, Multiple instance boosting for object detection, in: Neural Information Processing Systems, vol. 18, MIT Press, Vancouver, British Columbia, Canada, 2006, pp. 1419–1426.
[25] J. Wang, J. Zucker, Solving the multiple-instance problem: A lazy learning approach, in: Proc. 17th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2000, pp. 1119–1125.
[26] L.A. Wolsey, Integer Programming, Wiley-Interscience, New York, 1998.
[27] Q. Zhang, S. Goldman, EM-DD: An improved multiple-instance learning technique, in: Neural Information Processing Systems, vol. 14, MIT Press, Vancouver, British Columbia, Canada, 2001, pp. 1073–1080.