Robust Support Vector Machines for Multiple Instance Learning

Mohammad H. Poursaeidi* O. Erhun Kundakcioglu*†

May 31, 2012

Abstract

This paper presents the multiple instance classification problem that can be used for drug and molecular activity prediction, text categorization, image annotation, and object recognition. In order to model a more robust representation of outliers, hard margin loss formulations that minimize the number of misclassified instances are proposed. Although the problem is $\mathcal{NP}$-hard, computational studies show that medium sized problems can be solved to optimality in reasonable time using integer programming and constraint programming formulations. A three-phase heuristic algorithm is proposed for larger problems. Furthermore, different loss functions such as hinge loss, ramp loss, and hard margin loss are empirically compared in the context of multiple instance classification. The proposed heuristic for robust support vector machines with hard margin loss is shown to be superior to other approaches for multiple instance learning in terms of generalization performance.

Keywords: Support vector machines, multiple instance learning, constraint programming, robust classification.

1 Introduction

Multiple Instance Learning (MIL) is a supervised machine learning problem, where class labels are defined on the sets, referred to as *bags*, instead of individual data instances. Each instance in a negative bag is negative, whereas positive bags may contain false positives. This notion of *bags* makes multiple instance learning particularly useful for numerous interesting applications. For instance, in drug activity prediction, unless there is at least one effective

*Department of Industrial Engineering, University of Houston, E209 Engineering Bldg. 2, Houston, TX 77204, USA. e-mail: mhpoursaeidi@uh.edu, erhun@uh.edu

†This work is supported by University of Houston New Faculty Research Grant.

ingredient (*actual positive instance*), a drug (*bag*) is ineffective (*negative labeled*). Similarly, in molecular activity prediction, in order to observe a particular activity (*positive labeled*) for a molecule (*bag*), there has to be at least one conformation (*instance*) that exhibits the desired behavior (*actual positive*). Text categorization deals with matching a document (*bag*) with a topic of interest (*positive label*) based on a set of keywords that have been frequently used in the same concept (*actual positive instances*). In image annotation, pictures with an object of interest (*positive labeled bags*) are not expected to include that object in all segments, but only in subsets (*actual positive instances*).

A number of different approaches have been proposed to perform classification for MIL data. Employed methods include diverse density, decision trees, nearest neighbor algorithm, and support vector machines. In this paper, we propose a robust approach for MIL based on hard margin Support Vector Machine (SVM) formulations. Cross validation results show that our approach provides more accurate predictions than a traditional SVM approach to MIL. In general, the term *robustness* implies a non-drastic change in performance under different settings such as noisy environment or worst case scenario depending on the context. In the context of classification, we use *robustness* to indicate minimal influence of *outliers* on the classifier, thus better generalization performance.

The remainder of this paper is organized as follows: In Section 2, we provide basics of SVM with different loss functions, MIL, and a brief literature survey. Section 3 defines the problem and presents exact integer programing and constraint programming formulations. In Section 4, we propose a three-phase heuristic to be used for larger problems for both linear and nonlinear classification. Section 5 presents the optimality performance of our heuristic and cross validation results for the proposed approach on publicly available data sets. In order to show the hard margin loss is of the essence for robustness, we also demonstrate cross validation results for linear classification using hinge loss, ramp loss, and hard margin loss on randomly generated data sets. We provide brief concluding remarks and directions for future research in Section 6.

2 Background

2.1 Support Vector Machines

SVMs are supervised machine learning methods that are originally used to classify pattern vectors which belong to two linearly separable sets from two different classes [29]. The classification is achieved by a hyperplane that maximizes the distance between the convex hulls of both classes. Although extensions are proposed for regression and multi-class classification, SVMs are particularly useful for binary (2-class) classification due to strong fundamentals from the statistical learning theory, implementation advantages (e.g., sparsity), and generalization performance. When misclassified instances are penalized in the linear form, SVM

classifiers are proven to be universally consistent [24]. A classifier is *consistent* if the probability of misclassification (in expectation) converges to a Bayes' optimal rule when the number of data instances increase. A classifier is *universally consistent* if it is consistent for all distributions of data. SVMs can also perform nonlinear classification utilizing separating curves by implicitly embedding original data in a nonlinear space using *kernel functions*. SVMs have a wide range of applications including pattern recognition [4], text categorization [14], biomedicine [3, 16, 21], brain-computer interface [19, 16], and financial applications [28, 13].

In a typical *binary classification* problem, class $\mathbf{S}^+$ and $\mathbf{S}^-$ are composed of pattern vectors $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, n$. If $\mathbf{x}_i \in \mathbf{S}^+$, it is given the label $y_i = 1$; if $\mathbf{x}_i \in \mathbf{S}^-$, then it is given the label $y_i = -1$. The ultimate goal is to determine which class a new pattern vector $\mathbf{x}_i \notin \{\mathbf{S}^+ \cup \mathbf{S}^-\}$ belongs to. SVM classifiers solve this problem by finding a hyperplane $(\mathbf{w}, b)$ that separates instances in classes $\mathbf{S}^+$ and $\mathbf{S}^-$ with the maximum interclass margin. The original *hinge loss* 2-class SVM problem is as follows:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \tag{1a}$$

$$\text{subject to} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \qquad \forall i \tag{1b}$$

$$\xi \geq 0 \qquad \forall i. \tag{1c}$$

In this formulation, $\mathbf{w}$ is the normal vector and b is the offset parameter for the separating hyperplane. ξ_i are slack variables for misclassified pattern vectors. The goal is to maximize the interclass margin [1] and minimize misclassification. The role of scalar C in the objective function is to control the trade-off between margin violation and regularization. It should be noted that parameter C might differ for positive and negative class (e.g., C_1 and C_2) to cover *unbalanced* classification problems.

Lagrangian dual formulation for (1) leads to an optimization problem where input vectors only appear in the form of dot products and a suitable kernel function can be introduced for nonlinear classification [8]. This dual problem is a concave maximization problem, which can be solved efficiently. The dual for hinge loss formulation in (1) is given as

$$\max \quad \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{2a}$$

$$\text{subject to} \quad \sum_i y_i \alpha_i = 0 \tag{2b}$$

$$0 \leq \alpha_i \leq C \qquad \forall i. \tag{2c}$$

Using a hinge loss function for ξ_i as in (1a) or a quadratic loss function results in an increased sensitivity to outliers due to penalization of continuous measure of misclassification [2, 27, 32]. Different loss functions are proposed in the literature to model a better

[1] Maximizing interclass margin is identical to minimizing $\|\mathbf{w}\|$ when functional distance $\langle \mathbf{w}, \mathbf{x}_i \rangle + b$ is bounded as in (1b). See [29] for details.

representation of the outliers that leads to more robust classifiers. These functions ensure that the distance from the hyperplane has a limited (if any) effect on the quality of the solution for misclassified instances. For instance, *hard margin loss* considers the number of misclassifications instead of their distances to the hyperplane [2]. Minimizing the number of misclassified points is proven to be $\mathcal{NP}$-hard [6]. Orsenigo and Vercellis [22] use a similar approach called discrete SVM (DSVM), and propose a heuristic algorithm to generate local optimum decision trees. Recently, Brooks [2] formulate the hard margin loss formulation using a set of binary variables v_i, which are equal to one if the instance is misclassified.

$$\min_{\mathbf{w},b,\boldsymbol{v}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i v_i \tag{3a}$$

$$\text{subject to} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1, \text{if } v_i = 0 \qquad \forall i \tag{3b}$$

$$v_i \in \{0,1\} \qquad \forall i \tag{3c}$$

Constraints (3b) can be linearized using a sufficiently large constant M as follows:

$$\min_{\mathbf{w},b,\boldsymbol{v}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i v_i \tag{4a}$$

$$\text{subject to} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - Mv_i \qquad \forall i \tag{4b}$$

$$v_i \in \{0,1\} \qquad \forall i. \tag{4c}$$

In SVM classifiers, functional distance (i.e., $\langle \mathbf{w}, \mathbf{x}_i\rangle + b$) is expected to be equal to 1 (-1) for correctly classified *positive (negative) labeled instances that provide support*. Therefore, a positive and a negative labeled instance can be on the desired sides of the hyperplane yet incur misclassification penalties when functional distances are in $(0, 1)$ and $(-1, 0)$, respectively. In order to smooth out this effect, an approach is to penalize misclassified instances with a functional distance in $(-1, 1)$ based on their distance and incur a fixed penalty for those out of $(-1, 1)$ range [2, 18]. This approach is called *ramp loss* or *robust hinge loss*, which can be formulated as

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{v}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C(\sum_i \xi_i + 2\sum_i v_i) \tag{5a}$$

$$\text{subject to} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i, \text{ if } v_i = 0 \qquad \forall i \tag{5b}$$

$$v_i \in \{0,1\} \qquad \forall i \tag{5c}$$

$$0 \leq \xi_i \leq 2 \qquad \forall i, \tag{5d}$$

where the conditional constraint (5b) can be linearized using M as follows:

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{v}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C(\sum_i \xi_i + 2\sum_i v_i) \tag{6a}$$

$$\text{subject to} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i - Mv_i \qquad \forall i \tag{6b}$$

$$v_i \in \{0, 1\} \qquad \forall i \tag{6c}$$

$$0 \leq \xi_i \leq 2 \qquad \forall i. \tag{6d}$$

Shen et al. [23] use optimization with ramp loss but the solution method does not guarantee global optimality. Xu et al. [32] solve the non-convex optimization problem using semi-definite programming techniques but state that the procedure works inefficiently. Wang et al. [31] propose a concave-convex procedure (CCCP) to transform the associated non-convex optimization problem into a convex problem and use Newton optimization technique in the primal space.

In the next section, we focus on MIL and present methods that are employed highlighting a set of SVM studies.

2.2 Multiple Instance Learning

The MIL setting is introduced by Dietterich et al. [9] for the task of drug activity prediction and design. Same setting has also been studied for applications such as identification of proteins [26], content based image retrieval [34], object detection [30], prediction of failures in hard drives [20] and text categorization [1]. In contrast to a typical classification setting where instance labels are known with certainty, MIL deals with uncertainty in labels. In multiple instance binary classification, a positive bag label shows that there is at least one actual positive instance in the bag which is a *witness* for the label. On the other hand, all instances in a negative bag must belong to the negative class so there is no uncertainty on negative labeled bags.

Several methods have been applied to solve MIL problems, from expectation maximization methods with diverse density (EM-DD) [7, 33], to deterministic annealing [12], to extensions of k-NN, citation k-NN, and diverse density methods [10], to kernel based SVM methods [1].

SVM methods have first been employed by Andrews et al. [1] for MIL. In this study, integer variables are used to indicate witness status of points in positive bags. Witness point has to be placed on the positive side of the decision boundary, otherwise a penalty is incurred. Selecting each of these representations leads to a heuristic for solving the resulting mixed-integer program approximately. In contrast, Mangasarian and Wild [17] introduce continuous variables to represent the convex combination of each positive bag, which must be placed on the positive side of the separating plane. This representation leads to an optimization problem that contains both linear and bilinear constraints, which is solved to a

local optimum solution through a linear programming algorithm. An integer programming formulation that penalizes negative labeled instances without a bag notion is proposed in [15]. The setting leads to a maximum margin hyperplane between a selection of instances from positive bags and all instances from negative bags. This problem is proven to be $\mathcal{NP}$-hard and a branch and bound algorithm is proposed.

Next, we introduce our robust classification approach through different hard margin loss formulations for MIL.

3 Mathematical Modeling

Despite the large number of approaches for MIL, to the best of our knowledge, our study is the first one that utilizes a robust SVM classifier for MIL. Instead of a continuous measure for misclassification, we use a hard margin loss formulation and minimize the number of misclassified instances to overcome the aforementioned outlier sensitivity issue.

The data consists of pattern vectors (instances) $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, n$ and bags $j = 1, \ldots, m$. Each data instance belongs to one and only bag. Bags are labeled positive or negative and sets of positive and negative bags are represented as $J^+ = \{j : y_j = 1\}$ and $J^- = \{j : y_j = -1\}$, respectively. Note that, labels y_j are associated with bags, rather than instances. Next, we introduce instances in positive and negative bags as $I^+ = \{i : i \in I_j \wedge j \in J^+\}$, $I^- = \{i : i \in I_j \wedge j \in J^-\}$, respectively. The goal in our robust SVM model is to maximize the interclass margin where a fixed penalty (independent from the distance) is incurred for a bag if

- the bag is positive labeled and all instances in the bag are misclassified (on the negative side),
- the bag is negative labeled and at least one instance in the bag is misclassified (on the positive side).

Here we present three integer programming and two constraint programming formulations for the described model.

3.1 Integer Programming Formulations

In order to use hard margin loss for multiple instance data, we define a set of variables η_i to indicate actual positive instances from each positive bag. η_i is one when we select positive instance i (as witness) and zero otherwise. We consider one selected instance from each positive bag as the witness of all instances in that bag. In order to incorporate the effect of misclassifying a bag in the objective function, we introduce two sets of variables v_j^+, v_j^- that indicate misclassification of positive and negative bags, respectively. A positive bag is

misclassified ($v_j^+ = 1$) if all the instances in that positive bag is misclassified ($v_i = 1\ \forall i \in I_j, j \in J^+$). A negative bag is misclassified ($v_j^- = 1$) if at least one instance in that bag is misclassified ($\exists i \in I_j, j \in J^- | v_i = 1$). Therefore, the multiple instance hard margin SVM (MIHMSVM) can be formulated as follows:

$$\textbf{MIHMSVM} \quad \min_{\mathbf{w},b,\boldsymbol{\eta},\boldsymbol{v},\boldsymbol{v}^-} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j\in J^-} v_j^- + C\sum_{i\in I^+} v_i \tag{7a}$$

$$\text{subject to} \quad -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - Mv_i \qquad \forall i \in I^- \tag{7b}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - Mv_i - M(1-\eta_i) \qquad \forall i \in I^+ \tag{7c}$$

$$\sum_{i\in I_j} \eta_i = 1 \qquad \forall j \in J^+ \tag{7d}$$

$$v_i \leq v_j^- \qquad \forall j \in J^-, i \in I_j \tag{7e}$$

$$0 \leq v_j^- \leq 1 \qquad \forall j \in J^- \tag{7f}$$

$$v_i \in \{0,1\} \qquad \forall i \tag{7g}$$

$$\eta_i \in \{0,1\} \qquad \forall i \in I^+. \tag{7h}$$

In this formulation, (7c) is always satisfied for all positive instances that are not witnesses (i.e., $\eta_i = 0$), which sets $v_i = 0$ due to nature of the objective function. Therefore, only the witness of a positive bag with $\eta_i = 1$ might deteriorate the objective function. This ensures that a positive bag does not incur any penalty if at least one instance is correctly classified. On the other hand, v_i values for negative instances are calculated as in a typical classification problem. However, (7e) ensures that the maximum of these values are penalized in the objective function and a negative bag does not incur a penalty if all instances are correctly classified. It should be noted that MIHMSVM is $\mathcal{NP}$-hard since a special case with a single instance in each bag is proven to be $\mathcal{NP}$-hard [17].

This formulation utilizes $2|I^+| + |I^-|$ binary variables and $|J^-|$ continuous variables. Instead of using constraints (7e), we can use the binaries inside separation constraints directly. This will not only reduce the number of binary variables, but eliminate the need for continuous variables as well. We obtain a simpler formulation with $2|I^+| + |J^-|$ binary variables as follows:

$$\textbf{IP1} \quad \min_{\mathbf{w},b,\boldsymbol{\eta},\boldsymbol{v},\boldsymbol{v}^-} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j\in J^-} v_j^- + C\sum_{i\in I^+} v_i \tag{8a}$$

$$\text{subject to} \quad -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - Mv_j^- \qquad \forall j \in J^-, i \in I_j \tag{8b}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - Mv_i - M(1-\eta_i) \qquad \forall i \in I^+ \tag{8c}$$

$$\sum_{i\in I_j} \eta_i = 1 \qquad \forall j \in J^+ \tag{8d}$$

$$v_i, \eta_i \in \{0,1\} \qquad \forall i \in I^+ \tag{8e}$$

$$v_j^- \in \{0,1\} \qquad \forall j \in J^-. \tag{8f}$$

Next formulation, influenced by Mangasarian and Wild [17], considers the fact that it is enough to select the instances with minimum misclassification from positive bags. Therefore, we utilize variables v_j^+, for positive bags that shows the minimum misclassification associated with that bag. By penalizing this variable in the objective function, we obtain

$$\begin{aligned}
\min_{\mathbf{w},b,\boldsymbol{\eta},\boldsymbol{v},\boldsymbol{v}^+,\boldsymbol{v}^-} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j\in J^-} v_j^- + C\sum_{j\in J^+} v_j^+ && && \text{(9a)}\\
\text{subject to} \quad & -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - Mv_j^- && \forall j \in J^-, i \in I_j && \text{(9b)}\\
& \langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - Mv_i && \forall i \in I^+ && \text{(9c)}\\
& v_j^+ = \sum_{i\in I_j} \eta_i v_i && \forall j \in J^+ && \text{(9d)}\\
& \sum_{i\in I_j} \eta_i = 1 && \forall j \in J^+ && \text{(9e)}\\
& v_j^+ \in \{0,1\} && \forall j \in J^+ && \text{(9f)}\\
& v_j^- \in \{0,1\} && \forall j \in J^- && \text{(9g)}\\
& v_i, \eta_i \in \{0,1\} && \forall i \in I^+. && \text{(9h)}
\end{aligned}$$

In order to linearize (9d), we introduce new variables $\hat{z}_i$ that should be equal to $\eta_i v_i$. We relax the integrality of η_i and v_i^+ and come up with the following formulation with $|I^+|+|J^-|$ binary variables:

$$\textbf{IP2} \quad \min_{\mathbf{w},b,\boldsymbol{\eta},\boldsymbol{v},\boldsymbol{v}^+,\boldsymbol{v}^-,\hat{\boldsymbol{z}}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j\in J^-} v_j^- + C\sum_{j\in J^+} v_j^+ \tag{10a}$$

$$\text{subject to} \quad -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - Mv_j^- \qquad \forall j \in J^-, i \in I_j \tag{10b}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - Mv_i \qquad \forall i \in I^+ \tag{10c}$$

$$v_j^+ = \sum_{i\in I_j} \hat{z}_i \qquad \forall j \in J^+ \tag{10d}$$

$$\hat{z}_i \geq -1 + \eta_i + v_i \qquad \forall i \in I^+ \tag{10e}$$

$$\hat{z}_i \leq v_i \qquad \forall i \in I^+ \tag{10f}$$

$$\hat{z}_i \leq \eta_i \qquad \forall i \in I^+ \tag{10g}$$

$$\sum_{i\in I_j} \eta_i = 1 \qquad \forall j \in J^+ \tag{10h}$$

$$0 \leq v_j^+ \leq 1 \qquad \forall j \in J^+ \tag{10i}$$

$$0 \leq \hat{z}_i \leq 1 \qquad \forall i \in I^+ \tag{10j}$$

$$0 \leq \eta_i \leq 1 \qquad \forall i \in I^+ \tag{10k}$$

$$v_j^- \in \{0,1\} \qquad \forall j \in J^- \tag{10l}$$

$$v_i \in \{0,1\} \qquad \forall i \in I^+. \tag{10m}$$

It should be noted that constraints (10f) and (10g) are redundant since the summation of $\hat{z}_i$ is to be minimized.

Next, we obtain a novel formulation using the number of instances in positive bags to identify positive bag witnesses. Our experience with the following formulation is that it is far superior compared to **IP1** and **IP2**. We use the fact that, a positive bag is misclassified if all instances in that bag are misclassified, i.e., $\sum_{i\in I_j} v_i = |I_j|$. We also relax the integrality of v_i^+ and obtain a formulation with $|I^+| + |J^-|$ binary variables:

$$\textbf{IP3} \quad \min_{\mathbf{w},b,\boldsymbol{v},\boldsymbol{v}^+,\boldsymbol{v}^-} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j\in J^-} v_j^- + C\sum_{j\in J^+} v_j^+ \tag{11a}$$

$$\text{subject to} \quad -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - Mv_j^- \qquad \forall j \in J^-, i \in I_j \tag{11b}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - Mv_i \qquad \forall i \in I^+ \tag{11c}$$

$$v_j^+ \geq \sum_{i\in I_j} v_i - |I_j| + 1 \qquad \forall j \in J^+ \tag{11d}$$

$$0 \leq v_j^+ \leq 1 \qquad \forall j \in J^+ \tag{11e}$$

$$v_j^- \in \{0,1\} \qquad \forall j \in J^- \tag{11f}$$

$$v_i \in \{0,1\} \qquad \forall i \in I^+. \tag{11g}$$

Suppose j' is a positive bag with $|I_{j'}|$ instances. When all of the instances in the bag are misclassified (i.e., $v_i = 1, \forall i \in I_{j'}$) then $\sum_{i \in I_{j'}} v_i = |I_{j'}|$ and $v_i^+ = 1$ is forced. Otherwise, $\sum_{i \in I_{j'}} v_i \leq |I_{j'}| + 1$ and v_i^+ will be free and set to 0 due to the objective function.

Next, we present two constraint programming formulations for benchmarking purposes. In contrast to integer programming approaches, constraint programming prioritize exploiting special functions and finding a feasible solution during the computational procedure.

3.2 Constraint Programming Formulations

In order to evaluate the performance of IP formulations and take advantage of the special structure of the problem, we introduce two constraint programming formulations. IBM ILOG CPLEX CP Optimizer [25] is employed that utilize robust constraint propagation and search algorithms.

Our first constraint programming formulation is as follows:

$$\textbf{CP1} \quad \min_{\mathbf{w},b,\boldsymbol{v}^+,\boldsymbol{v}^-} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j \in J^-} v_j^- + C\sum_{j \in J^+} v_j^+ \tag{12a}$$

$$\text{subject to} \quad -(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \quad \vee \quad v_j^- \geq 1 \qquad \forall j \in J^-, i \in I_j \tag{12b}$$

$$\bigvee_{i \in I_j} \langle \mathbf{w}, \mathbf{x}_i \rangle + b \geq 1 \quad \vee \quad v_j^+ \geq 1 \qquad \forall j \in J^+ \tag{12c}$$

$$0 \leq v_j^- \leq 1 \qquad \forall j \in J^- \tag{12d}$$

$$0 \leq v_j^+ \leq 1 \qquad \forall j \in J^+. \tag{12e}$$

In **CP1**, (12b) is defined for all negative labeled instances and ensures that each negative labeled instance is correctly classified OR its corresponding bag is misclassified (i.e., $v_j^- = 1$). On the other hand, (12c) is defined for all positive bags and forces either one of the instances in the bag to be correctly classified OR the bag is misclassified (i.e., $v_j^+ = 1$).

Next, we propose a hybrid approach using constraint programming with the constraint set from a fast IP implementation, **IP3**. The formulation is as follows:

$$\textbf{CP2} \quad \min_{\mathbf{w},b,\boldsymbol{v},\boldsymbol{v}^+,\boldsymbol{v}^-} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j\in J^-} v_j^- + C\sum_{j\in J^+} v_j^+ \tag{13a}$$

$$\text{subject to} \quad -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 \quad \vee \quad v_j^- \geq 1 \qquad \forall j \in J^-, i \in I_j \tag{13b}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 \quad \vee \quad v_i \geq 1 \qquad \forall i \in I^+ \tag{13c}$$

$$v_j^+ \geq \sum_{i\in I_j} v_i - |I_j| + 1 \qquad \forall j \in J^+ \tag{13d}$$

$$0 \leq v_j^- \leq 1 \qquad \forall j \in J^- \tag{13e}$$

$$0 \leq v_j^+ \leq 1 \qquad \forall j \in J^+ \tag{13f}$$

$$0 \leq v_i \leq 1 \qquad \forall i \in I^+. \tag{13g}$$

In **CP2**, constraints on bag misclassification are partially adapted from **CP1** and **IP3**. Next, we present the nonlinear hard margin loss formulation for MIL.

3.3 Nonlinear Classification

By making the substitution $\mathbf{w} = \sum_{i=1}^n y_i x_i \alpha_i$ with nonnegative α_i variables for $i = 1, \ldots, n$ in (7), we obtain the following nonlinear classification formulation for multiple instance hard margin SVM:

$$\textbf{NLMIHMSVM} \quad \min_{\boldsymbol{\alpha},b,\boldsymbol{\eta},\boldsymbol{v},\boldsymbol{v}^-} \quad \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle \alpha_i \alpha_j + C\sum_{j\in J^-} v_j^- + C\sum_{i\in I^+} v_i \tag{14a}$$

$$\text{subject to} \quad -\sum_{j=1}^n y_j \langle \mathbf{x}_j, \mathbf{x}_i\rangle \alpha_j - b \geq 1 - Mv_i \qquad \forall i \in I^- \tag{14b}$$

$$\sum_{j=1}^n y_j \langle \mathbf{x}_j, \mathbf{x}_i\rangle \alpha_j + b \geq 1 - Mv_i - M(1-\eta_i) \qquad \forall i \in I^+ \tag{14c}$$

$$\alpha_i \geq 0 \qquad \forall i \tag{14d}$$

$$\alpha_i \leq M\eta_i \qquad \forall i \tag{14e}$$

$$\sum_{i\in I_j} \eta_i = 1 \qquad \forall j \in J^+ \tag{14f}$$

$$v_i \leq v_j^- \qquad \forall j \in J^-, i \in I_j \tag{14g}$$

$$v_i \in \{0,1\} \qquad \forall i \tag{14h}$$

$$0 \leq v_j^- \leq 1 \qquad \forall j \in J^- \tag{14i}$$

$$\eta_i \in \{0,1\} \qquad \forall i \in I^+. \tag{14j}$$

The use of (14) is that the original data can be embedded in a nonlinear space by replacing the dot products with a suitable kernel function $\mathbf{K}$ in (14a), (14b), and (14c). It

should be noted that (14e) ensures instances that are not selected do not play a role on the hyperplane. Therefore, for a given set of $\boldsymbol{\eta}$ values, the formulation reduces to the hard margin loss formulation in [2].

Note that, both linear and nonlinear formulations presented in this section can utilize different penalty terms to solve unbalanced classification problems. Next, we present formulations for different loss functions for the multiple instance classification problem.

3.4 Multiple Instance Classification with Hinge and Ramp Loss

In this section, we develop formulations for multiple instance hinge loss support vector machines and multiple instance ramp loss support vector machines for benchmarking purposes.

In order to incorporate bags in the objective function of hinge loss SVM, i.e., formulation (1), two sets of new variables ξ_j^+, ξ_j^- are introduced that incorporate the positive and negative bag misclassification, respectively. ξ_j^+ should be equal to minimum ξ_i in each positive bag to select the actual positive of that bag. For negative bags, ξ_j^- should be greater than or equal to each instance's ξ_i in that bag. Therefore, the problem can be formulated as

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^+,\boldsymbol{\xi}^-,\boldsymbol{\eta}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\Big(\sum_{j\in J^-}\xi_j^- + \sum_{j\in J^+}\xi_j^+\Big) \tag{15a}$$

$$\text{subject to} \quad -\left(\langle \mathbf{w}, \mathbf{x}_i\rangle + b\right) \geq 1 - \xi_i \qquad \forall i \in I^- \tag{15b}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i \qquad \forall i \in I^+ \tag{15c}$$

$$\xi_j^+ = \sum_{i\in I_j}\eta_i\xi_i \qquad \forall j \in J^+ \tag{15d}$$

$$\sum_{i\in I_j}\eta_i = 1 \qquad \forall j \in J^+ \tag{15e}$$

$$\xi_i \leq \xi_j^- \qquad \forall j \in J^-, i \in I_j \tag{15f}$$

$$\eta_i \in \{0,1\} \qquad \forall\, i \in I^+ \tag{15g}$$

$$\xi_i \geq 0 \qquad \forall i, \tag{15h}$$

which can be linearized as

$$
\begin{aligned}
\textbf{MIHLSVM} \quad \min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^+,\boldsymbol{\xi}^-,\boldsymbol{\eta},\boldsymbol{z}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C(\sum_{j\in J^-}\xi_j^- + \sum_{j\in J^+}\xi_j^+) && && \text{(16a)}\\
\text{subject to} \quad & -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i && \forall i \in I^- && \text{(16b)}\\
& \langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i && \forall i \in I^+ && \text{(16c)}\\
& \xi_j^+ = \sum_{i\in I_j} z_i && \forall j \in J^+ && \text{(16d)}\\
& z_i \geq \xi_i - M(1-\eta_i) && \forall i \in I^+ && \text{(16e)}\\
& z_i \leq \xi_i && \forall i \in I^+ && \text{(16f)}\\
& z_i \leq M\eta_i && \forall i \in I^+ && \text{(16g)}\\
& \sum_{i\in I_j}\eta_i = 1 && \forall j \in J^+ && \text{(16h)}\\
& \xi_i \leq \xi_j^- && \forall j \in J^-, i \in I_j && \text{(16i)}\\
& \eta_i \in \{0,1\} && \forall\, i \in I^+ && \text{(16j)}\\
& z_i \geq 0 && \forall i \in I^+ && \text{(16k)}\\
& \xi_i \geq 0 && \forall i. && \text{(16l)}
\end{aligned}
$$

Next, we formulate ramp loss for MIL. Similar to the previous formulations, variables $\xi_j^+, \xi_j^-, v_j^+, v_j^-$ are defined to incorporate the misclassification of positive and negative bags with the ramp loss definition discussed in Section (2). The resulting formulation for ramp loss SVM for MIL data is

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^+,\boldsymbol{\xi}^-,\boldsymbol{v}^+,\boldsymbol{v}^-,\boldsymbol{v},\boldsymbol{\eta}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C(\sum_{j\in J^-}\xi_j^- + \sum_{j\in J^+}\xi_j^+ + 2\sum_{j\in J^-}v_j^- + 2\sum_{j\in J^+}v_j^+) \tag{17a}$$

$$\text{subject to} \quad -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i - Mv_j^- \qquad \forall j \in J^-, i \in I_j \tag{17b}$$

$$\langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i - Mv_i \qquad \forall i \in I^+ \tag{17c}$$

$$\xi_j^+ = \sum_{i\in I_j}\eta_i\xi_i \qquad \forall j \in J^+ \tag{17d}$$

$$v_j^+ = \sum_{i\in I_j}\eta_i v_i \qquad \forall j \in J^+ \tag{17e}$$

$$\sum_{i\in I_j}\eta_i = 1 \qquad \forall j \in J^+ \tag{17f}$$

$$\xi_i \leq \xi_j^- \qquad \forall j \in J^-, i \in I_j \tag{17g}$$

$$v_i, \eta_i \in \{0,1\} \qquad \forall i \in I^+ \tag{17h}$$

$$v_j^+ \in \{0,1\} \qquad \forall j \in J^+ \tag{17i}$$

$$v_j^- \in \{0,1\} \qquad \forall j \in J^- \tag{17j}$$

$$0 \leq \xi_i \leq 2 \qquad \forall i, \tag{17k}$$

which can be linearized using two sets of variables,

$$\gamma_j^+ = \xi_j^+ + 2v_j^+ \quad \forall j \in J^+$$

$$\gamma_j^- = \xi_j^- + 2v_j^- \quad \forall j \in J^-,$$

as follows:

$$
\begin{aligned}
\textbf{MIRLSVM} \quad \min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\gamma}^+,\boldsymbol{\gamma}^-,\boldsymbol{v}^-,\boldsymbol{v},\boldsymbol{\eta},\boldsymbol{z}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C(\sum_{j\in J^-}\gamma_j^- + \sum_{j\in J^+}\gamma_j^+) && && \text{(18a)}\\
\text{subject to} \quad & -(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i - Mv_j^- && \forall j \in J^-, i \in I_j && \text{(18b)}\\
& \langle \mathbf{w}, \mathbf{x}_i\rangle + b \geq 1 - \xi_i - Mv_i && \forall i \in I^+ && \text{(18c)}\\
& \gamma_j^+ = \sum_{i\in I_j} z_i && \forall j \in J^+ && \text{(18d)}\\
& z_i \geq (\xi_i + 2v_i) - M(1-\eta_i) && \forall i \in I^+ && \text{(18e)}\\
& z_i \leq (\xi_i + 2v_i) && \forall i \in I^+ && \text{(18f)}\\
& z_i \leq M\eta_i && \forall i \in I^+ && \text{(18g)}\\
& \sum_{i\in I_j}\eta_i = 1 && \forall j \in J^+ && \text{(18h)}\\
& 2v_i^- + \xi_i \leq \gamma_j^- && \forall j \in J^-, i \in I_j && \text{(18i)}\\
& v_i, \eta_i \in \{0,1\} && \forall i \in I^+ && \text{(18j)}\\
& v_j^- \in \{0,1\} && \forall j \in J^- && \text{(18k)}\\
& 0 \leq \xi_i \leq 2 && \forall i && \text{(18l)}\\
& z_i \geq 0 && \forall i \in I^+. && \text{(18m)}
\end{aligned}
$$

Next section presents a heuristic algorithm for larger problems to be solved using hard margin loss formulation, where exact methods may be computationally intractable.

4 Three-Phase Heuristic Algorithm

In this section, we develop a three-phase heuristic for the proposed MIHMSVM model. First, we explore the details of the algorithm for linear classification and present the pseudocode. Next, we highlight the modifications needed to perform nonlinear classification.

4.1 Linear Classification

The idea of our algorithm is to start with a feasible hyperplane and fine tune the orientation considering MIL restrictions. Instead of starting with a random hyperplane, we take advantage of the efficiency of SVM on a typical classification problem. Therefore, the first phase of the algorithm consists of applying hinge loss SVM classifier on all instances considering their labels regardless of their bags. We use LIBSVM [5] since a fast classification of the data set is needed. The optimal separating hyperplane in this step $(\mathbf{w}_1, b_1)$ gives a rough idea on positioning of bags. Next, we select a representative for each bag. Bag representatives may be interpreted as witnesses for positive bags. Although MIL setting does not entail negative

bag witnesses, the reason we select representatives for negative bags is to keep the number of positive and negative labeled instances balanced and avoid biased classifications for the next step. The choice of bag representatives is based on the maximum functional distance from the hyperplane, which is in line with margin maximization objective considering MIL setting. This approach provides furthest correctly classified (or least misclassified) instances in positive bags and closest correctly classified (or most misclassified) instances in negative bags as representatives. Next, we use hinge loss SVM classifier for selected instances from all bags. The optimal separating hyperplane of this step is $(\mathbf{w}_2, b_2)$ that supposedly gives a better representation of data. This classifier will be used to find the correctly classified negative bags (where all instances are on negative side) and positive bags (where at least one instance is on positive side) as an initial solution at the end of the first phase.

In the second phase, a hard separation problem is solved. The instance with maximum functional distance from $(\mathbf{w}_2, b_2)$ in each correctly classified positive bag constitute the positive labeled training set. On the other hand, all instances in correctly classified negative bags are included in the negative labeled training set. Note that, a hard separation problem (i.e., formulation (3) where $v_i = 0, \forall i$) is polynomially solvable, and the resulting solution from phase one assures there will be no misclassification at this step. Since there are no misclassification terms for instances, an imbalance (possibly large number of negative labeled instances) does not imply a biased classifier. Let $(\mathbf{w}_3, b_3)$ be the optimal separating hyperplane at the end of this step. Next, we search for fast inclusion of misclassified bags while maintaining feasibility of the hard separation problem by fixing $(\mathbf{w}_3, b_3)$. Finally, we compute current objective function value of MIHMSVM using $\|\mathbf{w}_3\|^2$ and number of misclassified bags. This hyperplane also becomes the *current* best solution.

In the third (improvement) phase, we employ a more rigorous inclusion process. Misclassified bags are sorted in ascending order of their distance from their corresponding *support* hyperplane and considered as *candidates* to be correctly classified one by one. Distance between a positive bag and the support hyperplane is defined as the distance between closest instance and the positive support hyperplane (i.e., $\langle \mathbf{w}, \mathbf{x}_i \rangle + b = 1$). On the other hand, distance between a negative bag and the support hyperplane is defined as the distance between furthest instance and the negative support hyperplane (i.e., $\langle \mathbf{w}, \mathbf{x}_i \rangle + b = -1$). This approach is in line with our model assumptions in Section 3. If a positive bag is considered, instance with the smallest distance will be temporarily added to the training set. If a negative bag is selected, all instances in the bag will be temporarily added to the training set. Next, training set is examined for feasibility and if the problem is feasible, hyperplane $(\mathbf{w}_4, b_4)$ is obtained. If hard margin loss objective function is less than the current best objective, candidate bag will be added to the solution and best hyperplane is updated. The objective functions are compared based on the fact that by adding a bag, we decrease the misclassification by one in trade of a change in the norm of the hyperplane. Thus, in an iteration, if $(\|\mathbf{w}_4\|^2 - \|\mathbf{w}_{best}\|^2)/2$ is less than C, then we conclude the overall objective is reduced.

The search will continue until no improvement is possible and the final best solution is the heuristic solution for the problem.

Algorithm 1 Three-Phase Heuristic Algorithm (Linear Classification)
INPUT: $\mathbf{x}_1, \ldots, \mathbf{x}_n$, J^+, J^-, I^+, I^-, C
OUTPUT: $\mathbf{w}_{best}, b_{best}, Objective$

{**PHASE I**}
$P \leftarrow I^+$
$N \leftarrow I^-$
$\mathbf{w}_1, b_1 \leftarrow$ regular hinge-loss SVM hyperplane that separates P and N
Empty P and N
for all $j \in J^+$ **do**
  $P \leftarrow P \cup arg\max_{i \in I_j} \langle \mathbf{w}_1, x_i \rangle + b_1$
end for
for all $j \in J^-$ **do**
  $N \leftarrow N \cup arg\max_{i \in I_j} \langle \mathbf{w}_1, x_i \rangle + b_1$
end for
$\mathbf{w}_2, b_2 \leftarrow$ regular hinge-loss SVM hyperplane that separates P and N

{**PHASE II**}
Empty P and N
$number\ of\ misclassified\ bags \leftarrow 0$
for all $j \in J^+$ **do**
  if $\max_{i \in I_j} \langle \mathbf{w}_2, x_i \rangle + b_2 > 0$ **then**
    $P \leftarrow P \cup arg\max_{i \in I_j} \langle \mathbf{w}_2, x_i \rangle + b_2$
  else
    $number\ of\ misclassified\ bags \leftarrow number\ of\ misclassified\ bags + 1$
  end if
end for
for all $j \in J^-$ **do**
  if $\max_{i \in I_j} \langle \mathbf{w}_2, x_i \rangle + b_2 < 0$ **then**
    $N \leftarrow N \cup I_j$
  else
    $number\ of\ misclassified\ bags \leftarrow number\ of\ misclassified\ bags + 1$
  end if
end for
$\mathbf{w}_3, b_3 \leftarrow$ hard separation SVM hyperplane that separates P and N
{***Fast Inclusion***}
for all $j \in J^+$ **do**
  if $I_j \cap P = \emptyset$ AND $\max_{i \in I_j} \langle \mathbf{w}_3, x_i \rangle + b_3 > 1$ **then**
    $P \leftarrow P \cup arg\max_{i \in I_j} \langle \mathbf{w}_3, x_i \rangle + b_3$
    $number\ of\ misclassified\ bags \leftarrow number\ of\ misclassified\ bags - 1$
  end if
end for
for all $j \in J^-$ **do**
  if $I_j \cap N = \emptyset$ AND $\max_{i \in I_j} \langle \mathbf{w}_3, x_i \rangle + b_3 < -1$ **then**
    $N \leftarrow N \cup I_j$
    $number\ of\ misclassified\ bags \leftarrow number\ of\ misclassified\ bags - 1$
  end if
end for
$Objective \leftarrow \frac{1}{2}\|\mathbf{w}_3\|^2 + C \times number\ of\ misclassified\ bags$

{**PHASE III**}
$active\_set \leftarrow \emptyset$
$\mathbf{w}_{best} \leftarrow \mathbf{w}_3$
$b_{best} \leftarrow b_3$
for all $j \in (J^+ \cup J^-)$ **do**
  if $I_j \cap (P \cup N) \neq \emptyset$ **then**
    $active\_set \leftarrow active\_set \cup j$
  end if
end for
while $active\_set \neq \emptyset$ **do**
  if $\min_{j \in (active\_set \cap J^+)}[-\max_{i \in I_j}(\langle \mathbf{w}_{best}, x_i \rangle + b_{best} - 1)] < \min_{j \in (active\_set \cap J^-)}[\max_{i \in I_j}(\langle \mathbf{w}_{best}, x_i \rangle + b_{best} + 1)]$ **then**
    $candidate \leftarrow \ arg\min_{j \in (active\_set \cap J^+)}[-\max_{i \in I_j}(\langle \mathbf{w}_{best}, x_i \rangle + b_{best} - 1)]$
    $P \leftarrow P \cup arg\max_{i \in I_{candidate}}(\langle \mathbf{w}_{best}, x_i \rangle + b_{best} - 1)$
  else
    $candidate \leftarrow \ arg\min_{j \in (active\_set \cap J^-)}[\max_{i \in I_j}(\langle \mathbf{w}_{best}, x_i \rangle + b_{best} + 1)]$
    $N \leftarrow N \cup I_{candidate}$
  end if
  $active\_set \leftarrow active\_set \setminus candidate$
  if hard separation for P and N is feasible **then**
    $\mathbf{w}_4, b_4 \leftarrow$ hard separation SVM hyperplane that separates P and N
    if $\frac{1}{2}\|w_4\|^2 - \frac{1}{2}\|w_{best}\|^2 < C$ **then**
      $\mathbf{w}_{best} \leftarrow \mathbf{w}_4$
      $b_{best} \leftarrow b_4$
      $Objective \leftarrow Objective + \frac{1}{2}\|w_4\|^2 - \frac{1}{2}\|w_{best}\|^2 - C$
    else
      $P \leftarrow P \setminus I_{candidate}$
      $N \leftarrow N \setminus I_{candidate}$
    end if
  else
    $P \leftarrow P \setminus I_{candidate}$
    $N \leftarrow N \setminus I_{candidate}$
  end if
end while

4.2 Nonlinear Classification

Nonlinear extension of Algorithm 1 utilizes a number of modifications. In the first phase, regular hinge loss SVM is substituted with nonlinear SVM with a kernel function to obtain $(\boldsymbol{\alpha}_1, b_1)$. Next, in the construction of P and N, $\langle \mathbf{w}_1, x_i \rangle$ are substituted with $\sum_{j=1}^{n} y_j \mathbf{K}(\mathbf{x}_j, \mathbf{x}_i)\alpha_{1j}$ to calculate the distances. At the last step of the first phase, nonlinear SVM with kernel is employed again to obtain $(\boldsymbol{\alpha}_2, b_2)$. Likewise, in the second phase, $\langle \mathbf{w}_2, x_i \rangle$ are substituted with $\sum_{j=1}^{n} y_j \mathbf{K}(\mathbf{x}_j, \mathbf{x}_i)\alpha_{2j}$.

In order to obtain a nonlinear hard separation in Phase 2, we used the following formulation based on [2]:

$$\min_{\boldsymbol{\alpha},b} \quad \frac{1}{2}\sum_{i\in P\cup N}\sum_{j\in P\cup N} y_i y_j \mathbf{K}(\mathbf{x}_i,\mathbf{x}_j)\alpha_i\alpha_j \tag{19a}$$

$$\text{subject to} \quad \sum_{j\in P\cup N} y_j \mathbf{K}(\mathbf{x}_j,\mathbf{x}_i)\alpha_j + b \geq 1 \qquad \forall i \in P \tag{19b}$$

$$-\sum_{j\in P\cup N} y_j \mathbf{K}(\mathbf{x}_j,\mathbf{x}_i)\alpha_j - b \geq 1 \qquad \forall i \in N \tag{19c}$$

$$\alpha_i \geq 0 \qquad \forall i \in P\cup N \tag{19d}$$

Optimal solution to (19) provides $(\boldsymbol{\alpha}_3, b_3)$ that is used for fast inclusion. For distance calculation and in order to ensure hard separability, $\langle \mathbf{w}_3, x_i\rangle$ are substituted with $\sum_{j=1}^{n} y_j \mathbf{K}(\mathbf{x}_j,\mathbf{x}_i)\alpha_{3j}$. At the last step of Phase 2, $\|\mathbf{w}_3\|^2$ is substituted with the optimal objective function value of (19), i.e., $1/2\sum_{i\in P\cup N}\sum_{j\in P\cup N} y_i y_j \mathbf{K}(\mathbf{x}_i,\mathbf{x}_j)\alpha_{3i}\alpha_{3j}$.

As expected, in the third phase, instead of working with $\mathbf{w}$, we keep considering $\boldsymbol{\alpha}$ vectors. Decision of *candidate* instance for inclusion is performed by substituting dot products $\langle \mathbf{w}_{best}, x_i\rangle$ with $\sum_{j=1}^{n} y_j \mathbf{K}(\mathbf{x}_j,\mathbf{x}_i)\alpha_{bestj}$. Hard separation with $(\mathbf{w}_4, b_4)$ is also substituted with $(\boldsymbol{\alpha}_4, b_4)$ which gets the optimal solution for formulation (19). Next, we report computational performance for the proposed algorithm. We also show hard margin loss is virtually more robust and better in terms of generalization performance compared to other loss functions.

5 Computational Results

In this section, we first present the superior performance of hard margin loss in practice compared to ramp and hinge loss functions using randomly generated data sets. Next, we evaluate the performance of our heuristic in terms of time and proximity to the optimal solution. Finally, we show the cross validation performance of the proposed heuristic on the publicly available data sets. All computations are performed on a 2.93 GHz Intel Core 2 Duo computer with 4.0 GB RAM. The algorithms are implemented in C++ and used in conjunction with MATLAB 7.11.0 (2010b) environment in which the data resides.

We use MUSK1 and MUSK2 data set from UCI Machine Learning Repository [11]. MUSK1 data set consists of descriptions of 92 molecules (bags) with different shapes or conformations. Among them 47 of molecules judged by human experts are labeled as musks (positive bags) and remaining 45 molecules are labeled as non-musks (negative bags). The total number of conformations (instances) are 476 that gives an average of 5.2 conformations for each molecule (bag). MUSK2 data set consists of descriptions of 102 molecules in which 39 of molecules are labeled as musks and remaining 63 molecules are labeled as non-musks. Total number of conformations is 6,598 which gives an average of 64.7 conformations for each molecule. Each conformation in data sets is represented with a vector of 166 features extracted from surface properties.

Leave One Bag Out Cross Validation

Traditional cross validation methods (e.g., leave one out, n-fold) cannot reflect a fair assessment of multiple instance approaches due to ambiguity with actual instance labels. Therefore, we employ an extension that we refer to as *leave one bag out cross validation* (LOBOCV), which uses one bag from the original data set for validation (test data) and remaining instances as training data. After the separating hyperplane is obtained, label of the test bag is predicted and compared with its actual label. This routine is repeated until each bag in the sample is validated once and the percentage of correctly classified bags is reported.

5.1 Robustness of MIHMSVM

The robustness of the objectives will be discussed based on randomly generated data and the results obtained using IBM ILOG CPLEX Optimization Studio 12.2 [25]. Table 1 shows the cross validation results for three loss functions presented, namely hard margin loss (MIHMSVM) in (7), ramp loss (MIRLSVM) in (18), and hinge loss (MIHLSVM) in (16). In our computational studies, we consider a number of different C values. Small values result in a larger number of misclassified bags, which is not desired. On the other hand, values greater than 1 do not lead to a drastic decrease in the number of misclassifications (see [2]). Therefore, we set $C = 1$ for our experiments in this section. This penalty parameter also provides the best generalization performance for larger data sets, as shown in Section 5.3. Problem instances are generated using predetermined number of bags and features and the following pattern vector distributions:

TB1 Normal distribution: Features for instances in negative bags are normally distributed with mean 0, standard deviation 1. The mean of features for a positive bag are normally distributed with mean 1, standard deviation 5, and instances within each positive bag are offset using a normal distribution with mean 0, standard deviation 1. There are 4 instances in each positive and negative bag.

TB2 Uniform distribution: Features for instances in negative bags are uniformly distributed between -1 and 2. The mean of features for a positive bag are uniformly distributed between -2 and 4, and instances within each positive bag are offset uniformly between -1 and 1. There are 4 instances in each positive and negative bag.

TB3 Randomly selected features and bags from MUSK1 data set

The results shows hard margin loss is usually superior in practice compared to other loss functions. Loss functions would have minimal effect on classifiers for easy problems where a clean separation is possible. This can be observed in Table 1 when the ratio of number

Testbed	# of Bags	# of Features	Hard Margin Loss (MIHMSVM)	Ramp Loss (MIRLSVM)	Hinge Loss (MIHLSVM)
TB1	15	60	60.00%	60.00%	60.00%
TB2	15	60	80.00%	80.00%	80.00%
TB3	15	60	46.67%	**53.33%**	**53.33%**
TB3	15	60	80.00%	80.00%	80.00%
TB3	15	60	66.67%	66.67%	66.67%
TB1	20	80	50.00%	50.00%	50.00%
TB2	20	80	55.00%	55.00%	55.00%
TB3	20	80	**65.00%**	50.00%	50.00%
TB3	20	80	**45.00%**	40.00%	40.00%
TB3	20	80	**40.00%**	35.00%	35.00%
TB1	25	80	80.00%	80.00%	80.00%
TB2	25	80	88.00%	88.00%	88.00%
TB3	25	80	**56.00%**	36.00%	40.00%
TB3	25	80	**64.00%**	44.00%	40.00%
TB3	25	80	**56.00%**	36.00%	40.00%

Table 1: Leave-one-bag-out cross validation results for randomly generated multiple instance learning problems using different loss functions.

of instances to number of features is relatively low. In fact, for all cases created using TB1 and TB2, we observe the same accuracy for all three loss functions, which are not presented due to space considerations. This behavior changes for (i) odd distributions with outliers, (ii) when there are bags with small number of instances, and (iii) when the ratio of number of instances to number of features is higher. This directly points to MUSK1 data set with larger number of instances as can be seen in the last few rows of Table 1. In order to show this effect on relatively smaller instances, we generate the following instances by injecting outliers:

TB1o Normal distribution: Features for instances in negative bags are normally distributed with mean 0, standard deviation 4. The mean of features for a positive bag are normally distributed with mean 5, standard deviation 4. There are 4 instances in each positive and negative bag. One out of five negative bags are injected one noisy instance that is normally distributed with mean ± 90 and standard deviation 2.

TB2o Uniform distribution: Features for instances in negative bags are uniformly distributed between -10 and 10. The mean of features for a positive bag are uniformly distributed between -5 and 15. There are 4 instances in each positive and negative bag. One out of five negative bags are injected one noisy instance that is uniformly distributed between $\pm(80,100)$.

Table 2 highlights accuracy differences for the three loss functions. Although separating hyperplanes are different, accuracies are the same in cases with 20 bags and 10 features. When the number of bags increase or the number of features decrease, accuracies tend to

Testbed	# of Bags	# of Features	Hard Margin Loss (MIHMSVM)	Ramp Loss (MIRLSVM)	Hinge Loss (MIHLSVM)
TB1o	15	5	**86.67%**	**86.67%**	13.33%
TB1o	15	5	**80.00%**	**80.00%**	60.00%
TB1o	15	5	**93.33%**	**93.33%**	33.33%
TB2o	15	5	53.33%	46.67%	**66.67%**
TB2o	15	5	**53.33%**	**53.33%**	40.00%
TB2o	15	5	**86.67%**	**86.67%**	33.33%
TB1o	20	10	65.00%	65.00%	65.00%
TB1o	20	10	45.00%	45.00%	45.00%
TB1o	20	10	30.00%	30.00%	30.00%
TB2o	20	10	40.00%	40.00%	40.00%
TB2o	20	10	40.00%	40.00%	40.00%
TB2o	20	10	35.00%	35.00%	35.00%
TB1o	25	10	**84.00%**	**84.00%**	40.00%
TB1o	25	10	**96.00%**	**96.00%**	36.00%
TB1o	25	10	**64.00%**	**64.00%**	60.00%
TB2o	25	10	**72.00%**	68.00%	56.00%
TB2o	25	10	**76.00%**	**76.00%**	36.00%
TB2o	25	10	76.00%	76.00%	76.00%

Table 2: Leave-one-bag-out cross validation results for randomly generated multiple instance learning problems with outliers using different loss functions.

change, hard margin usually performing the best among the three. This is more apparent for larger and fuzzier data sets that are presented in Section 5.3. It should be noted ramp loss formulation takes significantly more time that hinge and hard margin loss in all test cases, thus it is omitted from further benchmark problems. The complexity of ramp loss SVM for conventional data is an open problem but we conjecture that multiple instance learning with ramp loss is $\mathcal{NP}$-hard.

5.2 Heuristic Performance: Optimal Solution and Time

In order to assess the capabilities of different formulations, we employ principal component analysis (PCA) on the MUSK1 data set so variability of data can be controlled by choosing a subset of features. When controlling the size of the problems, features with larger (smaller) weights in the first few principal components can be selected to create data sets with more (less) variability. This is a naive process that sheds a light on the analysis since data with less variability is typically harder to separate with a separating hyperplane. We use IBM ILOG CPLEX Optimization Studio 12.2 [25] for all exact formulations and set the time limit to 30 minutes. As values greater than 1 do not lead to a significant decrease in the number of misclassifications but an artificial increase in the optimality gap for our heuristic, we set $C = 1$ for our experiments in this section as well.

Tables 3, 4, and 5 show that formulations **IP3** and **CP2** perform the best. In fact, **IP3** is superior to other formulations in a majority of test instances but **CP2** is particularly

# of Inst.	# of Feat.	CPU Time (sec.)						Objective Value	
		IP1	IP2	IP3	CP1	CP2	3-Phase Heuristic	3-Phase Heuristic	OPT
40	10	1.08	0.41	**0.28**	2.65	1.48	**0.01**	3.86	3.58
40	20	0.11	0.22	0.11	0.26	**0.10**	**0.01**	2.00	1.41
40	40	0.11	0.17	0.10	0.07	**0.05**	**0.01**	0.26	0.24
40	80	0.21	0.21	**0.19**	0.21	**0.19**	**0.01**	0.13	0.12
80	10	0.09	0.08	**0.06**	0.48	0.11	**0.01**	1.00	1.00
80	20	2.12	1.18	**1.17**	5.01	3.15	**0.02**	3.86	2.12
80	40	8.81	3.54	**3.44**	6.31	5.02	**0.03**	1.97	1.66
80	80	6.12	4.67	20.07	3.35	**3.27**	**0.02**	0.32	0.26
120	10	156.59	**3.17**	3.27	N/A	475.74	**0.03**	7.13	7.10
120	20	3.91	3.27	**2.21**	N/A	16.30	**0.02**	4.68	3.25
120	40	1218.48	30.51	**21.95**	N/A	N/A	**0.07**	6.83	4.72
120	80	4.01	5.71	3.94	8.56	**3.38**	**0.06**	1.37	0.79
160	10	N/A	15.58	**13.10**	N/A	N/A	**0.10**	11.25	9.75
160	20	N/A	444.97	**295.91**	N/A	N/A	**0.05**	14.39	10.58
160	40	N/A	**47.55**	52.09	N/A	N/A	**0.06**	5.04	4.26
160	80	N/A	29.01	**21.06**	72.51	54.76	**0.12**	2.38	1.59
200	10	N/A	47.39	**43.43**	N/A	N/A	**0.08**	12.85	11.75
200	20	N/A	49.63	**38.06**	N/A	N/A	**0.05**	9.21	7.70
200	40	N/A	**123.63**	132.15	N/A	N/A	**0.07**	4.83	3.79
200	80	N/A	**15.83**	17.11	301.97	47.35	**0.15**	1.48	1.26
240	10	142.76	6.12	**4.10**	N/A	N/A	**0.13**	9.16	9.01
240	20	N/A	464.55	**291.64**	N/A	N/A	**0.08**	11.07	10.49
240	40	N/A	**173.80**	205.40	N/A	N/A	**0.14**	6.74	5.25
240	80	N/A	**1768.32**	N/A	N/A	N/A	**0.21**	5.14	3.60
280	10	N/A	20.90	**8.76**	N/A	N/A	**0.13**	11.95	11.00
280	20	N/A	N/A	N/A	N/A	N/A	**0.13**	20.65	N/A
280	40	N/A	N/A	N/A	N/A	N/A	**0.20**	11.92	N/A
280	80	N/A	1510.73	**899.54**	N/A	N/A	**0.41**	5.28	3.49
320	10	N/A	885.57	**559.06**	N/A	N/A	**0.22**	25.57	16.88
320	20	N/A	N/A	N/A	N/A	N/A	**0.22**	46.24	N/A
320	40	N/A	N/A	N/A	N/A	N/A	**0.24**	14.51	N/A
320	80	N/A	**1602.74**	N/A	N/A	N/A	**0.56**	6.62	3.71
360	10	N/A	N/A	N/A	N/A	N/A	**0.33**	32.99	N/A
360	20	N/A	N/A	N/A	N/A	N/A	**0.20**	23.22	N/A
360	40	N/A	N/A	N/A	N/A	N/A	**0.29**	12.77	N/A
360	80	N/A	1529.58	**1116.26**	N/A	N/A	**0.68**	9.23	3.93
400	10	N/A	N/A	N/A	N/A	N/A	**0.37**	25.41	N/A
400	20	N/A	N/A	N/A	N/A	N/A	**0.19**	34.42	N/A
400	40	N/A	N/A	N/A	N/A	N/A	**0.38**	14.98	N/A
400	80	N/A	N/A	N/A	N/A	N/A	**0.39**	6.24	N/A

Table 3: Computational results for harder data sets (i.e., subset of MUSK1 with less variability).

successful when number of features increase, which makes separation relatively easier. Our results show that, although we consider a harder generalization of an $\mathcal{NP}$-hard problem

# of Inst.	# of Feat.	CPU Time (sec.)						Objective Value	
		IP1	IP2	IP3	CP1	CP2	3-Phase Heuristic	3-Phase Heuristic	OPT
40	10	0.33	0.12	**0.07**	0.45	0.40	**0.02**	4.20	4.00
40	20	0.06	**0.05**	**0.05**	0.11	**0.05**	**0.01**	1.00	1.00
40	40	**0.06**	0.08	0.10	0.26	0.30	**0.01**	1.00	1.00
40	80	0.36	0.43	**0.29**	0.55	0.38	**0.02**	0.75	0.49
80	10	0.11	0.16	**0.07**	1.66	4.63	**0.03**	3.24	3.21
80	20	97.21	2.08	**1.07**	78.36	18.54	**0.04**	7.36	5.87
80	40	1.76	1.71	**1.12**	3.91	1.13	**0.03**	2.72	1.94
80	80	**4.44**	6.34	4.86	9.51	6.74	**0.02**	1.31	1.19
120	10	N/A	2.82	**1.57**	139.29	79.87	**0.04**	11.11	9.05
120	20	N/A	7.23	**4.12**	N/A	639.66	**0.04**	11.02	9.13
120	40	N/A	47.67	**31.89**	N/A	N/A	**0.05**	11.90	6.71
120	80	8.11	**3.51**	9.58	6.21	5.75	**0.06**	0.85	0.85
160	10	N/A	2.75	**1.38**	N/A	997.12	**0.09**	11.34	10.38
160	20	N/A	67.07	**35.90**	N/A	N/A	**0.06**	15.94	12.05
160	40	N/A	**90.21**	91.23	N/A	N/A	**0.07**	8.76	6.37
160	80	1666.50	**23.87**	29.74	N/A	N/A	**0.09**	4.29	3.54
200	10	N/A	9.11	**5.59**	N/A	347.75	**0.12**	14.25	14.22
200	20	N/A	19.19	**14.87**	N/A	N/A	**0.06**	10.12	9.73
200	40	N/A	**103.92**	134.32	N/A	N/A	**0.08**	15.16	9.70
200	80	N/A	**185.59**	194.51	N/A	N/A	**0.20**	7.82	3.93
240	10	55.55	2.87	**1.35**	N/A	N/A	**0.13**	8.77	8.77
240	20	N/A	449.23	**413.07**	N/A	N/A	**0.12**	18.67	15.95
240	40	N/A	**787.16**	1034.43	N/A	N/A	**0.09**	15.50	11.75
240	80	464.77	420.30	**203.53**	N/A	N/A	**0.11**	5.33	4.37
280	10	N/A	11.63	**7.09**	N/A	N/A	**0.21**	14.27	14.25
280	20	N/A	**217.74**	218.41	N/A	N/A	**0.13**	16.67	16.19
280	40	N/A	482.76	**397.70**	N/A	N/A	**0.19**	13.51	10.90
280	80	N/A	**249.66**	434.33	N/A	N/A	**0.21**	7.54	4.30
320	10	N/A	1257.40	**790.38**	N/A	N/A	**0.29**	31.59	30.38
320	20	N/A	372.36	**207.43**	N/A	N/A	**0.24**	17.75	17.49
320	40	N/A	N/A	N/A	N/A	N/A	**0.24**	30.91	N/A
320	80	N/A	N/A	N/A	N/A	N/A	**0.39**	11.77	N/A
360	10	N/A	94.62	**68.36**	N/A	N/A	**0.30**	21.43	21.38
360	20	N/A	744.08	**562.85**	N/A	N/A	**0.31**	20.13	18.96
360	40	N/A	N/A	N/A	N/A	N/A	**0.40**	30.69	N/A
360	80	N/A	N/A	N/A	N/A	N/A	**0.31**	8.52	N/A
400	10	N/A	301.65	**205.37**	N/A	N/A	**0.41**	26.29	26.25
400	20	N/A	**949.36**	1155.72	N/A	N/A	**0.20**	25.67	22.00
400	40	N/A	N/A	N/A	N/A	N/A	**0.24**	19.56	N/A
400	80	N/A	N/A	N/A	N/A	N/A	**0.71**	13.79	N/A

Table 4: Computational results for easier data sets (i.e., subset of MUSK1 with more variability).

in MIL context, medium sized problems can be solved in reasonable time using effective formulations.

# of Inst.	# of Feat.	CPU Time (sec.) IP1	IP2	IP3	CP1	CP2	3-Phase Heuristic	Objective Value 3-Phase Heuristic	OPT
80	166	10.92	9.32	10.28	4.37	**2.37**	**0.03**	0.34	0.30
120	166	222.70	37.73	306.04	67.37	**19.84**	**0.09**	0.34	0.29
160	166	63.78	49.33	173.77	25.59	**17.98**	**0.14**	0.51	0.45
200	166	N/A	138.59	**105.19**	798.40	195.55	**0.31**	1.42	0.99
240	166	N/A	945.99	**464.65**	N/A	838.98	**0.71**	1.45	1.20
280	166	N/A	659.91	373.44	N/A	**353.75**	**0.36**	0.91	0.79
320	166	N/A	655.65	**414.72**	N/A	478.25	**0.61**	1.72	1.04
360	166	N/A	N/A	N/A	N/A	N/A	**1.36**	3.06	N/A
400	166	N/A	N/A	N/A	N/A	N/A	**1.37**	3.53	N/A

Table 5: Computational results for a subset of instances in MUSK1 data set with all features.

Our heuristic also performs well compared to the optimal solution in terms of objective function value. It can be observed that the largest difference in objective function value between the heuristic and optimal solution in harder data sets is close to 9, when the total number of instances are 320 and the number of features was 10, which is a difficult separation problem. Although the optimality gap seems to be large, it should be noted that 8 or less additional bags are misclassified (among more than 60 bags) compared to the optimal solution with significant time savings. Furthermore, we expect proximity of heuristic hyperplane to the optimal hyperplane, thus a subtle difference in cross validation results.

5.3 Robust Classification Performance for Larger Data Sets: Cross Validation Results

5.3.1 Linear Classification

In this section, we present leave one bag out cross validation results for linear classification using the three-phase heuristic. All instances and features of MUSK1 data are used in computing these results. We also use a set of C values to observe the effect on the performance of our algorithm. As Table 6 shows, highest cross validation accuracy of 79.35% is achieved for $C = 1$.

Table 6 also shows the performance of our algorithm against hinge loss formulation (i.e., MIHLSVM) that is solved using CPLEX. Accuracy of our heuristic algorithm for MIHMSVM is consistently higher than MIHLSVM. It should be noted that the time reported in the table is for validation of 92 bags. For a given C value, it usually takes more than 20 days to perform cross validation using hinge loss formulation on CPLEX, whereas our heuristic takes less than 6 minutes.

C	Hard Margin Loss (Heuristic) LOBOCV	CPU Time (sec.)	Hinge Loss (CPLEX) LOBOCV	CPU Time (sec.)
0.1	**75.00%**	147.30	51.09%	**16.34**
1	**79.35%**	**217.43**	76.09%	1,818,460.63
10	**73.91%**	**321.21**	63.04%	1,816,458.85
100	**77.17%**	**312.66**	70.65%	1,819,085.86

Table 6: Leave-one-bag-out cross validation results for MUSK1 data with 476 instances in 92 bags and 166 features.

5.3.2 Nonlinear Classification

In order to assess the performance of our heuristic for nonlinear classification, MUSK2 data is considered with a Gaussian radial basis function. Formally, the Gaussian kernel is represented as

$$\mathbf{K}(\mathbf{x}_j, \mathbf{x}_i) = e^{-\frac{\|\mathbf{x}_j - \mathbf{x}_i\|^2}{2\sigma^2}}.$$

	$C = 0.5$		$C = 1$		$C = 10$		$C = 100$	
$2\sigma^2$	LOBOCV	CPU Time	LOBOCV	CPU Time	LOBOCV	CPU Time	LOBOCV	CPU Time
10	60.78%	22,639.80	63.73%	25,304.07	63.73%	23,736.96	63.73%	23,696.82
25	72.55%	25,804.41	79.41%	12,254.31	81.37%	11,228.20	81.37%	11,834.13
50	57.84%	18,956.22	**84.31%**	3,913.35	80.39%	3,461.50	81.37%	3,397.65
100	56.86%	13,245.11	79.41%	2,180.79	82.35%	1,926.41	81.37%	1,956.40
166	52.94%	13,083.71	76.47%	1,899.21	80.39%	1,559.36	79.41%	1,540.54
200	51.96%	12,998.93	79.41%	1,924.97	78.43%	1,507.11	79.41%	1,409.86
500	49.02%	12,837.94	75.49%	2,138.56	77.45%	1,416.91	79.41%	**1,199.26**
1000	49.02%	12,831.96	44.12%	9,764.32	47.06%	9,287.45	47.06%	9,221.11

Table 7: Leave-one-bag-out cross validation accuracy and CPU time (in seconds) results for MUSK2 data with 6,598 instances in 102 bags and 166 features.

Different C and σ values are compared and the results are presented in Table 7. The default selection in [5] is also considered that sets $2\sigma^2$ equal to the number of features. The best accuracy achieved is 84.31% for $C = 1$ and $\sigma = 5$. It should be noted that $C = 0.1$ is not presented in Table 7 because the regularization term outweighs the misclassification term in the objective function and the same cross validation accuracy of 38.24% is obtained for all values of σ. Our results show that the accuracy tends to decrease when σ increases as this converges to a linear separation. The total time spent for cross validation of 102 bags for our heuristic rarely exceeds an hour for nonextreme values of parameters. It is also noteworthy to mention that the time spent usually reduces with increased C since the misclassification penalty outweighs the quadratic regularization term in the objective function, providing a relatively more tractable problem.

6 Concluding Remarks

In this paper, we propose a robust support vector machine classifier for multiple instance learning. We show that hard margin loss classifiers provide remarkably better generalization performance for multiple instance data in practice, which is in line with theory. We develop 3 integer programs and 2 constraint programs and compare their time performance in achieving optimal solutions. Furthermore, we develop a heuristic that can handle large problem instances within reasonable time. Our heuristic provides higher cross validation accuracy for MIL data compared to conventional hinge loss based SVMs in significantly less time.

In this study, we observe that ramp loss classifiers are slow in practice. Alternative formulations can be developed and problem complexity can be studied for ramp loss SVM for conventional data. Next, ramp loss formulations can be extended to handle multiple instance data. Another important future study is a comparison of approaches for MIL using a fair cross validation scheme (e.g., leave one bag out), instead of random validation schemes that generate varying results in different runs.

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. *Advances in Neural Information Processing Systems*, pages 577–584, 2003.

[2] J. P. Brooks. Support vector machines with the ramp loss and the hard margin loss. *Operations Research*, 59(2):467–479, 2011.

[3] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, and D. Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. In *National Academy of Sciences of the United States of America*, volume 97, pages 262–267, 2000.

[4] H. Byun and S. W. Lee. Applications of support vector machines for pattern recognition: a survey. *Pattern Recognition with Support Vector Machines*, pages 571–591, 2002.

[5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:1–27, 2011.

[6] C. Chen and O. L. Mangasarian. Hybrid misclassification minimization. *Advances in Computational Mathematics*, 5:127–136, 1996.

[7] Y. Chen and J. Z. Wang. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5:913–939, 2004.

[8] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, UK, 2000.

[9] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

[10] D. R. Dooly, Q. Zhang, S. A. Goldman, and R. A. Amar. Multiple instance learning of real valued data. *Journal of Machine Learning Research*, 3:651–678, 2003.

[11] A. Frank and A. Asuncion. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, 2012. URL `http://archive.ics.uci.edu/ml`.

[12] P. V. Gehler and O. Chapelle. Deterministic annealing for multiple-instance learning. In *International Conference on Artificial Intelligence and Statistics*, 2007.

[13] Z. Huang, H. Chen, C.J. Hsu, W.H. Chen, and S. Wu. Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decision Support Systems*, 37(4):543–558, 2004.

[14] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98*, pages 137–142, 1998.

[15] O. E. Kundakcioglu, O. Seref, and P. M. Pardalos. Multiple instance learning via margin maximization. *Applied Numerical Mathematics*, 60(4):358–369, 2010.

[16] T. N. Lal, M. Schroder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, and B. Scholkopf. Support vector channel selection in BCI. *IEEE Transactions on Biomedical Engineering*, 51(6):1003–1010, 2004.

[17] O. L. Mangasarian and E. W. Wild. Multiple instance classification via successive linear programming. *Journal of Optimization Theory and Applications*, 137(3):555–568, 2008.

[18] L. Mason, P. Bartlett, and J. Baxter. Improved generalization through explicit optimization of margins. *Machine Learning*, 38(3):243–255, 2000.

[19] G. N. G. Molina, T. Ebrahimi, and J. M. Vesin. Joint time-frequency-space classification of eeg in a brain-computer interface application. *EURASIP Journal on Applied Signal Processing*, 2003:713–729, 2003.

[20] J. F. Murray, G. F. Hughes, and K. Kreutz-Delgado. Machine learning methods for predicting failures in hard drives: a multiple-instance application. *Journal of Machine Learning Research*, 6(1):783–816, 2006.

[21] W. S. Noble. *Kernel methods in computational biology*, chapter Support vector machine applications in computational biology, pages 71–92. MIT press, 2004.

[22] C. Orsenigo and C. Vercellis. Multivariate classification trees based on minimum features discrete support vector machines. *IMA Journal of Management Mathematics*, 14:221–234, 2003.

[23] X. Shen, G. C. Tseng, X. Zhang, and W. H. Wong. On ψ-learning. *Journal of the American Statistical Association*, 98(463):724–734, 2003.

[24] I. Steinwart. Support vector machines are universally consistent. *Journal of Complexity*, 18:768–791, 2002.

[25] IBM ILOG CPLEX Optimization Studio. 12.0 user manual, 2008.

[26] Q. Tao, S. Scott, N. V. Vinodchandran, and T. T. Osugi. Svm-based generalized multiple-instance learning via approximate box counting. In *Proceedings of the Twenty-First International Conference on Machine Learning*, page 101. ACM, 2004.

[27] T. B. Trafalis and R. C. Gilbert. Robust classification and regression using support vector machines. *European Journal of Operational Research*, 173(3):893–909, 2006.

[28] T. B. Trafalis and H. Ince. Support vector machine for regression and applications to financial forecasting. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, volume 6, pages 348–353. IEEE, 2000.

[29] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[30] P. Viola, J. Platt, and C. Zhang. Multiple instance boosting for object detection. *Advances in Neural Information Processing Systems*, 18, 2006.

[31] L. Wang, H. Ji, and J. Li. Training robust support vector machine with smooth ramp loss in the primal space. *Journal of the American Statistical Association*, 71:3020–3025, 2008.

[32] L. Xu, K. Crammer, and D. Schurmans. Robust support vector machine training via convex outlier ablation. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

[33] Q. Zhang and S. A. Goldman. EM-DD: An improved multiple-instance learning technique. *Advances in Neural Information Processing Systems*, 2:1073–1080, 2002.

[34] Q. Zhang, S. A. Goldman, W. Yu, and J.E. Fritts. Content-based image retrieval using multiple-instance learning. In *Proceedings of the 19th International Conference on Machine Learning*, pages 682–689. Citeseer, 2002.