A Sphere-Description-Based Approach for Multiple-Instance Learning

Yanshan Xiao, Bo Liu, and Zhifeng Hao

Abstract—Multiple-instance learning (MIL) is a generalization of supervised learning which addresses the classification of bags. Similar to traditional supervised learning, most of the existing MIL work is proposed based on the assumption that a representative training set is available for a proper learning of the classifier. That is to say, the training data can appropriately describe the distribution of positive and negative data in the testing set. However, this assumption may not be always satisfied. In real-world MIL applications, the negative data in the training set may not sufficiently represent the distribution of negative data in the testing set. Hence, how to learn an appropriate MIL classifier when a representative training set is not available becomes a key challenge for real-world MIL applications. To deal with this problem, we propose a novel Sphere-Description-Based approach for Multiple-Instance Learning (SDB-MIL). SDB-MIL learns an optimal sphere by determining a large margin among the instances, and meanwhile ensuring that each positive bag has at least one instance inside the sphere and all negative bags are outside the sphere. Enclosing at least one instance from each positive bag in the sphere enables a more desirable MIL classifier when the negative data in the training set cannot sufficiently represent the distribution of negative data in the testing set. Substantial experiments on the benchmark and real-world MIL datasets show that SDB-MIL obtains statistically better classification performance than the MIL methods compared.

Index Terms—Multiple-instance learning, classification

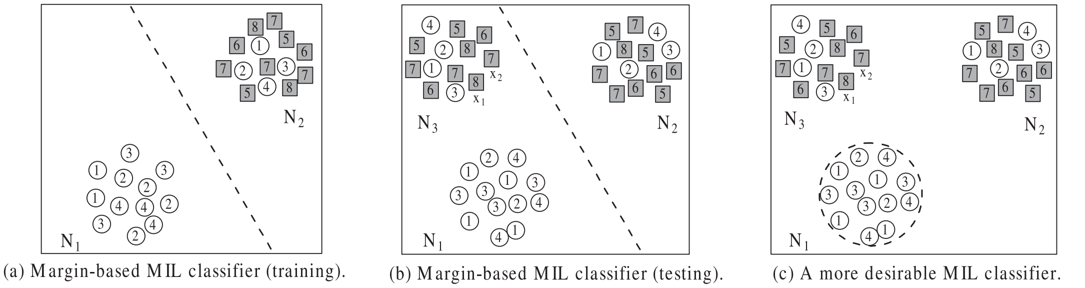
1 Introduction

Multiple-instance learning (MIL) [1], [2] is a paradigm in supervised learning which addresses the classification of bags. In MIL, a bag is labelled as positive if it includes at least one positive instance. It is labelled as negative if all of its instances are negative. Compared to traditional supervised learning, the key challenge of MIL is that the label of any single instance in a positive bag can be unavailable [3], [4]. This means that a positive bag may contain negative instances in addition to one or more positive instances. Many real-world applications can be solved by applying MIL. Take image categorization as an example. In MIL setting, an image is segmented into a number of regions. Each image is considered as a bag and each region is as an instance. A test image is predicted as positive if it contains at least one region related to the user interest. Otherwise, it is predicted as negative.

Similar to traditional supervised learning, most of the existing MIL work [5], [6], [7], [8], [9] is proposed based on the assumption that a representative training set is available for a proper learning of the classifier. Namely, the training data can sufficiently describe the distribution of positive and negative data in the testing set. However, this assumption may not be always satisfied. In real-world MIL applications, it may be difficult to label a representative set of negative training data which can sufficiently describe the distribution of negative data in the testing set. Consider an example in image categorization. Assume that the user is interested in the images relevant to “dog”, and the classifier is trained to predict whether an image is relevant to “dog”. To learn the classifier, we collect a number of images relevant to “dog” (positive bags), and those irrelevant to “dog” (negative bags) from the Internet. Gathering the positive images is relatively easy since they are related to the targeted topic - “dog”. However, collecting a representative set of negative images can be delicate and arduous. An image is labeled as negative if it is irrelevant to “dog”. In the Internet, there are all kinds of images and the number of image topics irrelevant to “dog” can be extensively large. It is difficult for us to determine all the negative topics irrelevant to “dog”. Moreover, even if all negative topics irrelevant to “dog” can be identified, it is too expensive to collect such a large quantity of images to represent each negative topic. Hence, the negative images collected for training the classifier may represent only a part of negative topics irrelevant to “dog”, and there may be some negative topics which appear in the testing set, but not in the training set. For example, suppose that the negative images in the training set are related to four negative topics -“computer”, “flower”, “piano” and “house”, which are irrelevant to “dog”. The testing data is an image from the Internet, and it can belong to any topic, e.g., “ship”, besides the four negative topics in the training set. If we predict the test image which is on “ship”, by using the classifier learnt on the images relevant to “dog” and those relevant to the four negative topics - “computer”, “flower”, “piano” and “house”, the classification performance may be unsatisfactory since the negative topic “ship” is not represented in the training set, and the negative training data cannot sufficiently describe

- *Y. Xiao and Z. Hao are with the School of Computers, Guangdong University of Technology, Guangzhou 510006, China. E-mail: xiaoyanshan@gmail.com, mazfhao@scut.edu.cn.*
- *B. Liu is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China. E-mail: csbliu@gmail.com.*

Manuscript received 20 Mar. 2014; revised 9 Feb. 2016; accepted 1 Mar. 2016. Date of publication 8 Mar. 2016; date of current version 16 Jan. 2017. Recommended for acceptance by D. Forsyth.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2016.2539952



(a) Margin-based MIL classifier (training). (b) Margin-based MIL classifier (testing). (c) A more desirable MIL classifier.

Fig. 1. An illustrative toy example. "○" signs represent the instances from positive bags and "□" signs stand for the instances from negative bags. The number inside the "○" or "□" sign indicates the bag ID.

the topics in the negative testing data. This characteristic of negative MIL data is usually neglected by traditional MIL methods.

When the negative data in the training set cannot sufficiently represent the distribution of negative data in the testing set, the learnt classifier may be biased. Let us consider an example in Figs. 1a and 1b. In Fig. 1a, a toy MIL training set consists of the instances from distributions N_1 and N_2, and a traditional margin-based MIL classifier (e.g., mi-SVM [5]) is learnt by maximizing the margin between the data from N_1 and N_2. It is seen that the MIL classifier can correctly classify the training bags since each positive bag has at least one instance classified as positive and all instances from negative bags are classified as negative. In Fig. 1b, the learnt classifier is used to predict a MIL testing set, which contains not only the instances from N_1 and N_2, but also N_3. It is observed that some instances (e.g., $\mathbf{x}_1$ and $\mathbf{x}_2$) which are from negative bags and should be negative, are misclassified to the positive class by the MIL classifier. This is because, the data distribution N_3 is described only in the testing set, not in the training set. The MIL classifier is not learnt on a representative training set. Hence, how to learn an appropriate MIL classifier when a representative training set is not available remains a key challenge for real-world MIL applications.

In this paper, we address the multi-instance learning problem where the negative data in the training set cannot sufficiently represent the distribution of negative data in the testing set. We propose a novel Sphere-Description-Based approach for Multiple-Instance Learning (SDB-MIL). SDB-MIL learns an optimal sphere by determining a large margin among the instances and requiring that each positive bag has at least one instance inside the sphere and all negative bags are outside the sphere. The convex concave constrained procedure (CCCP) [10] is applied to solve the learning problem.

The main characteristics of SDB-MIL are as follows:

- We put forward a sphere-description-based approach for MIL, which is able to incorporate the bag-level supervised information and instance-level unsupervised information into learning the sphere. In MIL, the bags are labelled, but the instances in positive bags are unlabelled. SDB-MIL constructs the sphere by considering both the bag-level labels and the unlabelled instances.
- We propose to solve the SDB-MIL learning problem by employing the CCCP technique, which is effective for resolving the nonconvex problems.
- Substantial experiments on MIL datasets have shown that SDB-MIL obtains statistically better classification performance than the MIL methods compared.

The rest of this paper is organized as follows. Section 2 reviews the related work. The details of SDB-MIL are presented in Section 3. Experiments are shown in Section 4. Section 5 concludes the paper and offers the future work.

2 Related Work

We first briefly review the existing work on MIL in Section 2.1. Since our proposed method is a sphere-based classifier, we then introduce the basic ideas of two representative sphere-based classifiers - multiple-instance optimal ball (MIOptimalBall) and support vector data description with negative instances (SVDD-NEG) in Sections 2.2 and 2.3, respectively.

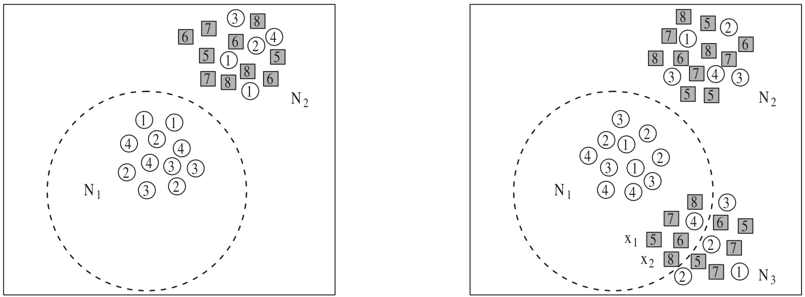
2.1 Multiple-Instance Learning

The MIL problem was initially proposed for drug activity prediction [1], [3], [11]. Over the past few years, many applications, e.g., image retrieval [12], [13], text categorization [14] and natural scene classification [15], are formulated as MIL problems. To solve these problems, many MIL methods are proposed [5], [6], [7], [16].

Some methods are specially designed to solve MIL problems. APR [1] represents the target concept by an axis-aligned rectangle. Diverse Density (DD) [17] seeks a data point (target concept) that is nearest to the positive bags and farthest from the negative bags. The target concept is learned by maximizing the DD function. EM-DD [18] combines expectation-maximization (EM) and the DD function by utilizing EM to accelerate the search of the target concept.

Some other methods [4], [7], [19] transform MIL problems into single-instance learning problems by introducing the MIL constraints. Andrews et al. [5] proposed two variants of SVM-based MIL methods: mi-SVM and MI-SVM, which classify the bags by maximizing the margin between the positive bags and negative bags. Gehler and Chapelle [19] employed deterministic annealing to the SVM formulation of MIL problems for finding better local mimima. Gärtner et al. [20] designed different kernels for MIL. Zhou and Xu [21] extended semi-supervised SVM to solve the MIL problems.

Recently, some methods [22] try to map a bag of instances into a data point and convert MIL problems into single-instance learning problems. DD-SVM [8] learns a collection of instance prototypes and uses them to map a bag of instances to a point in the bag space. SVM is then used to classify the "bag-level" vectors. In MILES [23], the bag is embedded



(a) MIOptimalBall on the training set. (b) MIOptimalBall on the testing set.

Fig. 2. Illustration of MIOptimalBall (base classifier) on toy MIL training and testing sets.

in a new feature space and 1-norm SVM is applied to select the important features (instances) for prediction.

However, most of the existing MIL work is proposed based on the assumption that the training data can appropriately represent the distribution of the testing data. Nevertheless, in real-world MIL applications, the negative data in the training set may not sufficiently describe the distribution of negative data in the testing set. In this case, how to learn a proper MIL classifier becomes a key challenge for real-world MIL applications. This motivates the work in our paper.

2.2 Multiple-Instance Optimal Ball (MIOptimalBall)

Different from the margin-based MIL classifiers [5], [8], [19], [20], MIOptimalBall [6], [24] is proposed to determine a ball classifier by enclosing at least one instance from each positive bag into the ball. The MIOptimalBall classifier is constructed by capturing the characteristics of the positive data using an optimal ball and has demonstrated impressive classification accuracy on MIL datasets [3], [6], [24].

MIOptimalBall is an AdaBoost-based approach and its base classifier is a ball-based classifier which works as follows. First, it computes the distance $d(\mathbf{x}, B)$ from each instance $\mathbf{x}$ in each positive bag to all other bags B and $d(\mathbf{x}, B) = \min_{\mathbf{x}' \in B} d(\mathbf{x}, \mathbf{x}')$. Second, for each instance $\mathbf{x}$, $d(\mathbf{x}, B)$ are sorted in an increasing order. The ordered distance list of $\mathbf{x}$ is denoted as $\{d'_1(\mathbf{x}), \ldots, d'_n(\mathbf{x})\}$. Third, each instance $\mathbf{x}$ in each positive bag is assumed to be a ball center in turn, and the ball radius is computed as $R = \frac{d'_{i-1}(\mathbf{x}) + d'_i(\mathbf{x})}{2}$ $(i = 1, \ldots, n)$. Lastly, the ball reaching the highest training accuracy with the maximized radius is selected as the optimal ball. A new bag is predicted as positive if it has instances in the ball. Otherwise, it is negative.

MIOptimalBall utilizes an optimal ball to enclose the positive data. However, the ball learnt by MIOptimalBall may be a relatively loose one which cannot ensure an appropriate description of the boundary of the positive data. Fig. 2a shows the ball learnt by MIOptimalBall on a toy MIL training set. It is seen that the training accuracy of MIOptimalBall is 100 percent since each positive bag has at least one instance included in the ball and all instances of negative bags are outside the ball. Moreover, we can observe that the ball learnt by MIOptimalBall is relatively loose. MIOptimalBall selects the ball with the maximum radius as the optimal one if more than one ball have the same highest training accuracy. In Fig. 2a, there are more than one ball obtaining 100 percent training accuracy. Among these balls, MIOptimalBall chooses the ball with the maximum radius. In Fig. 2b, the learnt ball is used to predict a toy MIL testing set. Some instances of N_3 (e.g., $\mathbf{x}_1$ and $\mathbf{x}_2$) which are from negative bags and should be negative, are enclosed in the ball and misclassified to the positive class. Hence, MIOptimalBall may be a relatively loose ball which may not guarantee a proper description of the data boundary and lead to data misclassification when the negative data in the training set cannot sufficiently represent the distribution of negative data in the testing set.

In this paper, we propose a sphere-description-based MIL method. Distinctive from MIOptimalBall, our method is a tight sphere around the instances from positive bags, which enables a more explicit description of the data boundary and helps to improve the classification accuracy when a representative set of negative data is unavailable for training the classifier.

2.3 SVDD with Negative Instances (SVDD-NEG)

Similar to MIOptimalBall, SVDD-NEG [25] learns a sphere to separate the positive and negative data. Given n_+ positive instances and n_- negative instances, SVDD-NEG seeks a sphere to enclose the positive instances in the sphere and exclude the negative instances from the sphere, as follows:

$$\begin{aligned} &\min_{\xi_i \geq 0} R^2 + C \sum \xi_i \\ &s.t. \| \mathbf{x}_i - \mathbf{o} \|^2 \leq R^2 + \xi_i, \quad i = 1, \ldots, n_+ \\ &\quad\;\; \| \mathbf{x}_i - \mathbf{o} \|^2 \geq R^2 - \xi_i, \quad i = n_+ + 1, \ldots, n_+ + n_-, \end{aligned} \tag{1}$$

where $\mathbf{o}$ and R are the sphere center and radius, respectively; ξ_i are training errors; C is a regularization parameter. A new instance $\mathbf{x}$ is predicted as positive if it satisfies $\| \mathbf{x} - \mathbf{o} \|^2 \leq R^2$. Otherwise, it is negative.

SVDD-NEG constructs a tight sphere to classify the positive and negative instances. However, SVDD-NEG is proposed for single-instance learning, where a label is associated with an instance. In MIL, a label is associated with a bag of instances and the instance labels are unknown. If we extend SVDD-NEG to MIL by making the instance label the same as its bag label, misclassification may occur since positive bags may contain negative instances, in addition to positive instances.

Inspired by the basic idea of SVDD-NEG, we learns a tight sphere by enclosing at least one instance from each positive bag into the sphere and excluding all negative bags from the sphere. Compared to SVDD-NEG, our method is able to incorporate the bag-level supervised information and instance-level unsupervised information into learning the MIL classifier.

3 Proposed Algorithm

3.1 Problem Statement

Suppose that the training set contains m_1 positive bags $(B_1, \ldots, B_{m_1})$ and m_2 negative bags $(B_{m_1+1}, \ldots, B_{m_1+m_2})$, where B_i $(i = 1, \ldots, m_1 + m_2)$ denotes the ith bag; m_1 and m_2 are the corresponding numbers of positive and negative bags. Let $m = m_1 + m_2$ be the total number of training bags. Each bag B_i contains a number of instances B_{ij}, where B_{ij} $(j = 1, \ldots, |B_i|)$ represents the jth instance in B_i; $|B_i|$ is the number of instances included in B_i. The numbers of instances in the positive bags and the negative bags are $n_1 = |B_1| + \cdots + |B_{m_1}|$ and $n_2 = |B_{m_1+1}| + \cdots + |B_m|$, respectively. $n = n_1 + n_2$ denotes the number of instances in all the bags. The task of MIL is to learn a classifier on the training set, and use it to predict an unknown bag.

3.2 Formulation

As discussed in Section 1, when the negative training data cannot appropriately describe the distribution of negative data in the testing set, it is more desirable to separate the data using a sphere, rather than only maximizing the margin between the positive and negative bags. Hence, we aim at seeking an optimal sphere, such that each positive bag has at least one instance in the sphere and all instances in the negative bags are outside the sphere. Based on this, one can tackle

$$\begin{aligned}
\min \quad & R^2 + C_1\left(\frac{1}{m_1}\sum_i \xi_i + \frac{1}{n_2}\sum_{i,j}\zeta_{ij}\right) \\
s.t. \quad & \bigcup_{j=1}^{|B_i|} I\Big(||B_{ij} - \mathbf{o}||^2 \le R^2 + \xi_i\Big) = 1,\ i = 1, \ldots, m_1 \\
& ||B_{ij} - \mathbf{o}||^2 \ge R^2 - \zeta_{ij},\ j \in B_i,\ i = m_1 + 1, \ldots, m \\
& \xi_i \ge 0,\ \zeta_{ij} \ge 0,
\end{aligned} \tag{2}$$

where $I(expression)$ is an indication function. It has $I(expression) = 1$ when the $expression$ holds true. Otherwise, $I(expression) = 0$. " $\bigcup$ " is an union operator.

The positive bag may contain negative instances, in addition to one or more positive instances. The labels of instances in the positive bag are unavailable. In the positive bag B_i, if an instance B_{ij} is identified to be positive, it should be enclosed in the sphere and satisfy the constraint $||B_{ij} - \mathbf{o}\ ||^2 \le R^2 + \xi_i$. Otherwise, it should be excluded from the sphere and the constraint $||B_{ij} - \mathbf{o}||^2 \le R^2 + \xi_i$ cannot be met. Hence, when $\bigcup_{j=1}^{|B_i|} I(||B_{ij} - \mathbf{o}||^2 \le R^2 + \xi_i) = 1$ holds, it implies that for all instances in the positive bag B_i, there is at least one positive instance B_{ij} which is included in the sphere and meets the constraint $||B_{ij} - \mathbf{o}||^2 \le R^2 + \xi_i$, with $I(||B_{ij} - \mathbf{o}||^2 \le R^2 + \xi_i) = 1$ being true. Furthermore, the second set of constraints in (2) requires that the instances in negative bags should be excluded from the sphere since all of them are negative.

Problem (2) involves the computation of indication functions $I(expression)$, which are difficult to solve. Therefore, we transform it to a simplified form, as follows:

$$\begin{aligned}
\min \ & R^2 + C_1\left(\frac{1}{m_1}\sum_i \xi_i + \frac{1}{n_2}\sum_{i,j}\zeta_{ij}\right) \\
s.t. \ & \min_{j\in B_i} ||B_{ij} - \mathbf{o}||^2 \le R^2 + \xi_i,\ i = 1, \ldots, m_1 \\
& ||B_{ij} - \mathbf{o}||^2 \ge R^2 - \zeta_{ij},\ j \in B_i,\ i = m_1 + 1, \ldots, m \\
& \xi_i \ge 0,\ \zeta_{ij} \ge 0.
\end{aligned} \tag{3}$$

In problem (3), we utilize the minimization function $min(expression)$ to replace the indication function $I(expression)$. It is easy to deduce that for a positive bag B_i, when $\bigcup_{j=1}^{|B_i|} I(||B_{ij} - \mathbf{o}||^2 \ \le R^2 + \xi_i) = 1$ comes true, the instance B_{ij} $(j = \arg\min_{j\in B_i} ||B_{ij} - \mathbf{o}||^2)$ which has the minimum distance to the sphere center, must satisfy the constraint $||B_{ij} - \mathbf{o}||^2 \le R^2 + \xi_i$. Therefore, problem (2) can be reduced into the learning problem in (3).

3.3 Enforcing Large Margin Constraint

The first set of constraints in problem (3) implies that for each positive bag, the instance which has the minimum distance to the sphere center, should be included in the sphere. However, in practice, the number of positive instances in the positive bag is unknown to us and the positive bag may contain far more than one instance. If we consider only one instance (the one nearest to the sphere center) from each positive bag to learn the classifier, and neglect the remaining positive instances in positive bags, some positive instances could be excluded from the sphere and the obtained classifier may not provide an appropriate description of the data boundary.

To cope with this problem, we introduce the large margin constraints and attempt to construct an optimal sphere by determining a large margin between the sphere surface and the instances outside the sphere. Supposing that ρ is a variable representing the margin between the sphere surface and the instances outside the sphere, the large margin constraints can be formulated as follows:

$$\begin{aligned}
& y_{ij}\left(||B_{ij} - \mathbf{o}||^2 - R^2 - \frac{1}{2}\rho\right) \le -\frac{1}{2}\rho + \eta_{ij}, \\
& \quad j \in B_i,\ j \ne \arg\min_{k\in B_i} ||B_{ik} - \mathbf{o}||^2,\ i = 1, \ldots, m_1,
\end{aligned} \tag{4}$$

where $y_{ij} \in \{1, -1\}$ is the label of instance B_{ij} which is needed to be optimized and η_{ij} are error terms.

In each positive bag, the instance nearest to the sphere center is enclosed in the sphere, as shown in the first set of constraints in (3). For the other instances in the positive bag, their labels are unavailable and they are required to satisfy the large margin constraint (4). According to the constraint (4), for an instance B_{ij} $(j \ne \arg\min_{k\in B_i} ||B_{ik} - \mathbf{o}||^2)$, if its label is positive, i.e., $y_{ij} = 1$, the constraint (4) is transformed into $||B_{ij} - \mathbf{o}||^2 \le R^2 + \eta_{ij}$, which indicates that B_{ij} is positive and should be included in the sphere. If its label is negative, i.e., $y_{ij} = -1$, the constraint (4) is changed into $||B_{ij} - \mathbf{o}||^2 \ge R^2 + \rho - \eta_{ij}$, which implies that the negative instance B_{ij} should lie outside the sphere and keep a certain margin ρ from the sphere surface.

By introducing the large margin constraint (4), a large number of instances whose labels are unavailable can be utilized to refine the decision boundary and improve the classifier accuracy. Nevertheless, the instance labels y_{ij} in the large margin constraint (4) are unknown to us and it is very time consuming to optimize over the instance labels y_{ij}. To speed up the optimization efficiency, we transform the large margin constraint (4) into the following form

$$\begin{aligned}&\left|\|B_{ij}-\mathbf{o}\|^2-R^2-\frac{1}{2}\rho\right|\le-\frac{1}{2}\rho+\eta_{ij},\\&\qquad j\in B_i,\ j\ne\arg\min_{k\in B_i}\|B_{ik}-\mathbf{o}\|^2,\ i=1,\ldots,m_1,\end{aligned}\tag{5}$$

where $|\cdot|$ is an absolute function.

Compared to the large margin constraint (4), the constraint (5) is much easier to solve, since it does not involve the computation of instance labels y_{ij} and thus the number of variables that need to be optimized can be largely reduced. By imposing the revised large margin constraint (5), the learning problem can be given by minimizing the sphere radius R and maximizing the margin ρ, as follows:

$$\begin{aligned}&\min\ R^2-C\rho+C_1\left(\frac{1}{m_1}\sum_i\xi_i+\frac{1}{n_2}\sum_{i,j}\zeta_{ij}\right)+\frac{C_2}{n_1}\sum_{i,j}\eta_{ij}\\&s.t.\ \min_{j\in B_i}\|B_{ij}-\mathbf{o}\|^2\le R^2+\xi_i,\quad i=1,\ldots,m_1\\&\|B_{ij}-\mathbf{o}\|^2\ge R^2+\rho-\zeta_{ij},\ j\in B_i,\ i=m_1+1,\ldots,m\\&\left|\|B_{ij}-\mathbf{o}\|^2-R^2-\frac{1}{2}\rho\right|\le-\frac{1}{2}\rho+\eta_{ij},\\&j\in B_i,\ j\ne\arg\min_{k\in B_i}\|B_{ik}-\mathbf{o}\|^2,\ i=1,\ldots,m_1\\&\rho\ge0,\ \xi_i\ge0,\quad\zeta_{ij}\ge0,\quad\eta_{ij}\ge0.\end{aligned}\tag{6}$$

3.4 Enforcing Class Balance Constraint

Besides the large margin constraint, one has to enforce the class balance constraint. It is imposed to avoid the trivially "optimal" solution, where all the instances are assigned to one class and an unbounded margin is obtained.

In SDB-MIL, the optimal label y_{ij} of instance B_{ij} is calculated as $y_{ij}=sgn(R^2-\|B_{ij}-\mathbf{o}\|^2)$. However, involving the $sgn(\cdot)$ function in the learning problem could make the optimization procedure much more complicated. If we consider $\widetilde{y}_{ij}=R^2-\|B_{ij}-\mathbf{o}\|^2$ as the "soft label" of B_{ij}, the class balance constraint can be given as

$$-l\le\sum_{i=1}^{m}\sum_{j\in B_i}\left(R^2-\|B_{ij}-\mathbf{o}\|^2\right)\le l,\tag{7}$$

where l is a parameter controlling the class balance. By imposing the constraint (7), problem (6) is varied into

$$\begin{aligned}&\min\ R^2-C\rho+C_1\left(\frac{1}{m_1}\sum_i\xi_i+\frac{1}{n_2}\sum_{i,j}\zeta_{ij}\right)+\frac{C_2}{n_1}\sum_{i,j}\eta_{ij}\\&s.t.\ \min_{j\in B_i}\|B_{ij}-\mathbf{o}\|^2\le R^2+\xi_i,\quad i=1,\ldots,m_1\\&\|B_{ij}-\mathbf{o}\|^2\ge R^2+\rho-\zeta_{ij},\ j\in B_i,\ i=m_1+1,\ldots,m\\&\left|\|B_{ij}-\mathbf{o}\|^2-R^2-\frac{1}{2}\rho\right|\le-\frac{1}{2}\rho+\eta_{ij},\\&j\in B_i,\ j\ne\arg\min_{k\in B_i}\|B_{ik}-\mathbf{o}\|^2,\ i=1,\ldots,m_1\\&-l\le\sum_{i=1}^{m}\sum_{j\in B_i}\left(R^2-\|B_{ij}-\mathbf{o}\|^2\right)\le l,\\&\rho\ge0,\ \xi_i\ge0,\quad\zeta_{ij}\ge0,\quad\eta_{ij}\ge0.\end{aligned}\tag{8}$$

After solving problem (8), we can obtain the sphere center $\mathbf{o}$ and radius R. However, it is difficult to get the exact solution of problem (8) efficiently. In the following, we relax problem (8) and approximate it using the CCCP technique [10], which is extensively and successfully applied in the optimization of SVM-based problems [26], [27], [28].

3.5 CCCP Decomposition

In the learning problem (8), the first set of constraints, i.e., $\min_{j\in B_i}\|B_{ij}-\mathbf{o}\|^2\le R^2+\xi_i\quad(i=1,\ldots,m_1)$, involves the "min" function. The third set of constraints, i.e., $|\|B_{ij}-\mathbf{o}\|^2-\ R^2-\frac{1}{2}\rho|\le-\frac{1}{2}\rho+\eta_{ij}\ \ (j\in B_i,\ j\ne\arg\min_{k\in B_i}\|B_{ik}-\mathbf{o}\|^2,\ i=1,\ldots,\ m_1)$, contains the absolute function and the "min" function. Therefore, problem (8) is difficult to solve. In order to resolve it, we consider to relax the third set of constraints as $|\|B_{ij}-\mathbf{o}\|^2-R^2-\frac{1}{2}\rho|\le-\frac{1}{2}\rho+\eta_{ij}$ $(j\in B_i,i=1,\ldots,m_1)$. After this relaxation, the third set of constraints does not contain the "min" function. Moreover, we define $s=\mathbf{o}^T\mathbf{o}-R^2$, as done in [29]. By substituting s and imposing the relaxed constraints, the learning problem (8) is changed into

$$\begin{aligned}&\min\ \mathbf{o}^T\mathbf{o}-s-C\rho+C_1\left(\frac{1}{m_1}\sum_i\xi_i+\frac{1}{n_2}\sum_{i,j}\zeta_{ij}\right)+\frac{C_2}{n_1}\sum_{i,j}\eta_{ij}\\&s.t.\ \min_{j\in B_i}\left(s-2\mathbf{o}^TB_{ij}+(B_{ij})^TB_{ij}\right)\le\xi_i,\quad i=1,\ldots,m_1\\&s-2\mathbf{o}^TB_{ij}+(B_{ij})^TB_{ij}\ge\rho-\zeta_{ij},\ j\in B_i,\ i=m_1+1,\ldots,m\\&\left|s-2\mathbf{o}^TB_{ij}+(B_{ij})^TB_{ij}-\frac{1}{2}\rho\right|\le-\frac{1}{2}\rho+\eta_{ij},\\&j\in B_i,\ i=1,\ldots,m_1\\&-l\le\sum_{i=1}^{m}\sum_{j\in B_i}\left(2\mathbf{o}^TB_{ij}-s-(B_{ij})^TB_{ij}\right)\le l,\\&\rho\ge0,\ \xi_i\ge0,\quad\zeta_{ij}\ge0,\quad\eta_{ij}\ge0.\end{aligned}\tag{9}$$

In problem (8), the constraints are quadratic. By defining s, as in [29], the constraints in problem (9) become linear and easier to be solved. However, problem (9) is still not convex due to the nonconvex constraints. Fortunately, it is the difference between two convex functions and we can solve it using CCCP, which is proposed to deal with the optimization problems with a concave-convex objective function under concave-convex constraints. In the following, we will show how to employ CCCP to solve problem (9).

To simplify the presentation, we first let $h(\mathbf{o},s,\rho,i,j)=|s-2\mathbf{o}^TB_{ij}+(B_{ij})^TB_{ij}-\frac{1}{2}\rho|$ and $g(\mathbf{o},s,i)\ =\min_{j\in B_i}\ (s-2\mathbf{o}^TB_{ij}\ +(B_{ij})^TB_{ij})$. Then, the first set of constraints in the learning problem (9) is changed into $g(\mathbf{o},s,i)\le\xi_i$, and the second set of constraints turns into $h(\mathbf{o},s,\rho,i,j)\le-\frac{1}{2}\rho+\eta_{ij}$. Based on this, CCCP is employed to decompose the nonconvex problem (9) into a series of convex problems. Given an initial point $(\mathbf{o}^{(0)},s^{(0)},\rho^{(0)})$, CCCP computes $(\mathbf{o}^{(t+1)},s^{(t+1)},\rho^{(t+1)})$ from $(\mathbf{o}^{(t)},s^{(t)},\rho^{(t)})$ iteratively by replacing $g(\mathbf{o},s,i)$ and $h(\mathbf{o},s,\rho,i,j)$ with their corresponding first-order Taylor expansions at $(\mathbf{o}^{(t)}, s^{(t)}, \rho^{(t)})$. The resulting problem is solved until the stopping criterion is met.

Specifically, to apply CCCP in solving the learning problem (9), we first calculate the first-order Taylor expansions of $g(\mathbf{o}, s, i)$ at $(\mathbf{o}^{(t)}, s^{(t)}, \rho^{(t)})$. However, $g(\mathbf{o}, s, i)$ is a non-smooth function at $(\mathbf{o}^{(t)}, s^{(t)}, \rho^{(t)})$. To solve this problem, we replace its gradient with subgradient when computing its tangent. Then, $g(\mathbf{o}, s, i)$ is decomposed at $(\mathbf{o}^{(t)}, s^{(t)}, \rho^{(t)})$ as

$$g(\mathbf{o}, s, i) = \sum_{j\in B_i} \theta_{ij}^{(t)}(s - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij}), \tag{10}$$

where it has

$$\theta_{ij}^{(t)} = \begin{cases} 1, & j = \arg\min_{j\in B_i}(s^{(t)} - 2(\mathbf{o}^{(t)})^T B_{ij} + (B_{ij})^T B_{ij}), \\ 0, & Otherwise. \end{cases}$$

Similarly, we decompose $h(\mathbf{o}, s, \rho, i, j)$ at $(\mathbf{o}^{(t)}, s^{(t)}, \rho^{(t)})$ as

$$h(\mathbf{o}, s, \rho, i, j) = b_{ij}^{(t)}(s - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij} - \frac{1}{2}\rho). \tag{11}$$

where $b_{ij}^{(t)} = sgn(s^{(t)} - 2(\mathbf{o}^{(t)})^T B_{ij} + (B_{ij})^T B_{ij} - \frac{1}{2}\rho^{(t)})$.

In the learning problem (9), we replace $g(\mathbf{o}, s, i)$ and $h(\mathbf{o}, s, \rho, i, j)$ with their corresponding first order approximations (10) and (11). Then, the convex optimization problem for the tth iteration of CCCP is given as:

$$\begin{aligned}
\min\ & \mathbf{o}^T\mathbf{o} - s - C\rho + C_1\left(\frac{1}{m_1}\sum_i \xi_i + \frac{1}{n_2}\sum_{i,j}\zeta_{ij}\right) + \frac{C_2}{n_1}\sum_{i,j}\eta_{ij} \\
s.t.\ & \sum_{j\in B_i}\theta_{ij}^{(t)}\left(s - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij}\right) \le \xi_i, \quad i = 1, \ldots, m_1 \\
& s - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij} \ge \rho - \zeta_{ij}, \ j \in B_i, \ i = m_1 + 1, \ldots, m \\
& b_{ij}^{(t)}\left(s - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij} - \frac{1}{2}\rho\right) \le -\frac{1}{2}\rho + \eta_{ij}, \\
& j \in B_i, \ i = 1, \ldots, m_1 \\
& -l \le \sum_{i=1}^{m}\sum_{j\in B_i}\left(2\mathbf{o}^T B_{ij} - s - (B_{ij})^T B_{ij}\right) \le l, \\
& \rho \ge 0, \ \xi_i \ge 0, \quad \zeta_i \ge 0, \quad \eta_{ij} \ge 0.
\end{aligned} \tag{12}$$

3.6 Dual Form

The learning problem (12) is a quadratic programming (QP) problem which is usually solved via the dual form. Hence, we give the dual form of problem (12) in this section.

Before obtaining the dual form, we first re-define the following notations. Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{m_1}, \mathbf{x}_{m_1+1}, \ldots, \mathbf{x}_{m_1+n_2}, \mathbf{x}_{m_1+n_2+1}, \ldots, \mathbf{x}_{m_1+n_2+n_1}, \mathbf{x}_{m_1+n_2+n_1+1}, \mathbf{x}_{m_1+n_2+n_1+2}\}$, where the first m_1 elements are equal to $\sum_{j\in B_i}\theta_{ij}^{(t)} B_{ij}(i = 1, \ldots, m_1)$; the (m_1+1)th to (m_1+n_2)th elements correspond to $B_{ij}(j \in B_i, i = m_1+1, \ldots, m)$; the (m_1+n_2+1)th to $(m_1+n_2+n_1)$th elements are $B_{ij}(j \in B_i, i = 1, \ldots, m_1)$; the $(m_1+n_2+ \ n_1+1)$th and $(m_1+n_2+n_1+2)$th elements are given as $\frac{1}{n_1+n_2}\sum_{i=1}^{m}\sum_{j\in B_i} B_{ij}$. Define $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_{m_1+n_2+n_1+2}\}$, where the first m_1 elements correspond to $\sum_{j\in B_i}\theta_{ij}^{(t)}(B_{ij})^T B_{ij} \quad (i = 1, \ldots, m_1)$; the (m_1+1)th to (m_1+n_2)th elements are $(B_{ij})^T B_{ij}(j \in B_i, i = m_1+1, \ldots, m)$; the (m_1+n_2+1)th to $(m_1+n_2+n_1)$th elements are $(B_{ij})^T B_{ij} \quad (j \in B_i, i = 1, \ldots, m_1)$; the $(m_1+n_2+n_1+1)$th and $(m_1+n_2+n_1+2)$th elements are equal to $\frac{1}{n_1+n_2}\sum_{i=1}^{m}\sum_{j\in B_i}(B_{ij})^T B_{ij}$. Furthermore, we make $\mathbf{P}^{(t)} = \{p_1^{(t)}, \ldots, p_{m_1}^{(t)}, p_{m_1+1}^{(t)}, \ldots, p_{m_1+n_2}^{(t)}, p_{m_1+n_2+1}^{(t)}, \ldots, p_{m_1+n_2+n_1}^{(t)}, p_{m_1+n_2+n_1+1}^{(t)}, p_{m_1+n_2+n_1+2}^{(t)}\}$ where $[p_1^{(t)}, \ldots, p_{m_1}^{(t)}]$ and $p_{m_1+n_2+n_1+1}^{(t)}$ are given as 1; $[p_{m_1+1}^{(t)}, \ldots, p_{m_1+n_2}^{(t)}]$ and $p_{m_1+n_2+n_1+2}^{(t)}$ correspond to -1; $[p_{m_1+n_2+1}^{(t)}, \ldots, p_{m_1+n_2+n_1}^{(t)}]$ are defined as $b_{ij}^{(t)}$. Let $\mathbf{U}^{(t)} = \{u_1^{(t)}, \ldots, u_{m_1+n_2+n_1+2}^{(t)}\}$ where $[u_{m_1+1}^{(t)}, \ldots, u_{m_1+n_2}^{(t)}]$ are 1; $[u_{m_1+n_2+1}^{(t)}, \ldots, u_{m_1+n_2+n_1}^{(t)}]$ are equal to $\frac{1}{2}(1 - b_{ij}^{(t)})(j \in B_i, i = m_1+1, \ldots, m)$; the other elements are 0. Make $\mathbf{C} = \{c_1, \ldots, c_{m_1+n_2+n_1}\}$ where $[c_1, \ldots, c_{m_1}]$ are $\frac{C_1}{m_1}$; $[c_{m_1+1}, \ldots, c_{m_1+n_2}]$ are $\frac{C_1}{n_2}$; $[c_{m_1+n_2+1}, \ldots, c_{m_1+n_2+n_1}]$ are $\frac{C_2}{n_1}$. Last, we define $\Gamma = \{\gamma_1, \ldots, \gamma_{m_1+n_2+n_1}\}$, where $[\gamma_1, \ldots, \gamma_{m_1}]$ are ξ_i $(i = 1, \ldots, m_1)$; $[\gamma_{m_1+1}, \ldots, \gamma_{m_1+n_2}]$ are ζ_{ij} $(j \in B_i, j = m_1+1, \ldots, m)$; $[\gamma_{m_1+n_2+1}, \ldots, \gamma_{m_1+n_2+n_1}]$ are η_{ij} $(j \in B_i, j = 1, \ldots, m_1)$. Based on these notations, problem (12) is rewritten as

$$\begin{aligned}
\min\ & \mathbf{o}^T\mathbf{o} - s - C\rho + \sum_{i=1}^{n_c} c_i\gamma_i \\
s.t.\ & p_i^{(t)}\left(s - 2\mathbf{o}^T\mathbf{x}_i + \mathbf{z}_i\right) + u_i^{(t)}\rho \le \gamma_i, \quad i = 1, \ldots, n_c \\
& p_i^{(t)}\left(s - 2\mathbf{o}^T\mathbf{x}_i + \mathbf{z}_i\right) + u_i^{(t)}\rho \le \frac{l}{n}, \quad i = n_c + 1, n_c + 2 \\
& \rho \ge 0, \ \gamma_i \ge 0, \ i = 1, \ldots, n_c,
\end{aligned} \tag{13}$$

where it has $n_c = m_1 + n_2 + n_1$ and $n = n_1 + n_2$.

By introducing the Lagrange multipliers $\alpha_i \ge 0$ $(i = 1, \ldots, n_c + 2)$, $\beta_i \ge 0 \ (i = 1, \ldots, n_c)$ and $\sigma_1 \ge 0$, the Lagrange function of problem can be obtained. Differentiating the Lagrange function with variables $\mathbf{o}$, s, ρ and γ_i results in

$$\frac{\partial L}{\partial \mathbf{o}} = 2\mathbf{o} - 2\sum_{i=1}^{n_c+2}\alpha_i p_i^{(t)}\mathbf{x}_i = 0, \tag{14}$$

$$\frac{\partial L}{\partial s} = -1 + \sum_{i=1}^{n_c+2}\alpha_i p_i^{(t)} = 0, \tag{15}$$

$$\frac{\partial L}{\partial \rho} = -C + \sum_{i=1}^{n_c+2}\alpha_i u_i^{(t)} - \sigma_1 = 0, \tag{16}$$

$$\frac{\partial L}{\partial \gamma_i} = c_i - \alpha_i - \beta_i = 0, \quad i = 1, \ldots, n_c. \tag{17}$$

According to Equations (14) and (15), we obtain

$$\mathbf{o} = \sum_{i=1}^{n_c+2}\alpha_i p_i^{(t)}\mathbf{x}_i, \tag{18}$$

$$\sum_{i=1}^{n_c+2}\alpha_i p_i^{(t)} = 1. \tag{19}$$

From Equations (16) and (17), we get

$$\sum_{i=1}^{n_c+2} \alpha_i u_i^{(t)} \geq C, \tag{20}$$

$$0 \leq \alpha_i \leq c_i, \quad i = 1, \ldots, n_c. \tag{21}$$

Substituting Equations (18)-(21) into the Lagrange function, the dual form of problem (13) can be given by

$$\begin{aligned} \max & \sum_{i=1}^{n_c+2} \alpha_i p_i^{(t)} \mathbf{z}_i - \sum_{i=1}^{n_c+2} \sum_{j=1}^{n_c+2} \alpha_i p_i^{(t)} (\mathbf{x}_i)^T \mathbf{x}_j p_j^{(t)} \alpha_j - \frac{l}{n} \sum_{i=n_c+1}^{n_c+2} \alpha_i \\ & \sum_{i=1}^{n_c+2} \alpha_i p_i^{(t)} = 1, \\ & \sum_{i=1}^{n_c+2} \alpha_i u_i^{(t)} \geq C, \\ & 0 \leq \alpha_i \leq c_i, \quad i = 1, \ldots, n_c. \end{aligned} \tag{22}$$

After solving (22), we can obtain the values of α_i and the sphere center $\mathbf{o}$ can be computed according to Equation (18).

In problem (12), $\theta_{ij}^{(t)}$ and $b_{ij}^{(t)}$ can be calculated from the solutions of the previous CCCP iteration, and hence they are known to us in the current iteration. Based on this, the first set of constraints in problem (12) is transformed into

$$\begin{aligned} & s - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij} \leq \xi_i, \\ & j : \theta_{ij}^{(t)} = 1, \quad i = 1, \ldots, m_1. \end{aligned} \tag{23}$$

The third set of constraints is changed into

$$\begin{aligned} & s - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij} \leq \eta_{ij}, \\ & b_{ij}^{(t)} = 1, \; j \in B_i, \; i = 1, \ldots, m_1, \end{aligned} \tag{24}$$

$$\begin{aligned} & s - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij} \geq \rho - \eta_{ij}, \\ & b_{ij}^{(t)} = -1, \; j \in B_i, \; i = 1, \ldots, m_1. \end{aligned} \tag{25}$$

Moreover, the second set of constraints is

$$\begin{aligned} & s - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij} \geq \rho - \zeta_{ij}, \\ & j \in B_i, \; i = m_1 + 1, \ldots, m. \end{aligned} \tag{26}$$

According to the KKT conditions [25], the instances with $0 < \alpha_i < c_i$ are support vectors (SVs) whose corresponding constraints become equation and it has $\xi_i = 0$, $\zeta_{ij} = 0$ or $\eta_{ij} = 0$. For all the instances with $0 < \alpha_i < c_i$, we let subset S_1 contain those corresponding to the constraints (23) and (24), and S_2 include those corresponding to the constraints (25) and (26). Hence, for the instances in S_1 and S_2, it has

$$s - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij} = 0, \quad B_{ij} \in S_1, \tag{27}$$

$$s - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij} - \rho = 0, \quad B_{ij} \in S_2. \tag{28}$$

Substituting $s = \mathbf{o}^T\mathbf{o} - R^2$ into (27) and (28) results in

$$R^2 = \mathbf{o}^T\mathbf{o} - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij}, \quad B_{ij} \in S_1, \tag{29}$$

$$\rho + R^2 = \mathbf{o}^T\mathbf{o} - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij}, \quad B_{ij} \in S_2. \tag{30}$$

Based on Equations (29) and (30), the sphere radius R and margin ρ can be computed as

$$R^2 = \frac{1}{|S_1|} \sum_{B_{ij} \in S_1} \Big(\mathbf{o}^T\mathbf{o} - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij} \Big), \tag{31}$$

$$\begin{aligned} \rho = & \frac{1}{|S_2|} \sum_{B_{ij} \in S_2} \Big(\mathbf{o}^T\mathbf{o} - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij} \Big) \\ & - \frac{1}{|S_1|} \sum_{B_{ij} \in S_1} \Big(\mathbf{o}^T\mathbf{o} - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij} \Big). \end{aligned} \tag{32}$$

where $|S_1|$ and $|S_2|$ represent the numbers of instances contained in subsets S_1 and S_2, respectively.

The CCCP iterations repeat until the stopping criterion is met. Based on the optimal solutions obtained, an unknown bag B_i will be classified to the positive class if it satisfies the inequality (33). Otherwise, it is classified to the negative class,

$$\min_{j \in B_i} ||B_{ij} - \mathbf{o}||^2 \leq R^2. \tag{33}$$

3.7 Algorithm Overview

In Table 1, we give an overview of the SDB-MIL algorithm which consists of a series of CCCP loops. Here, $J^{(t)} = \mathbf{o}^T\mathbf{o} - s - C\rho + C_1(\frac{1}{m_1}\sum_i \xi_i + \frac{1}{n_2}\sum_{i,j} \zeta_{ij}) + \frac{C_2}{n_1}\sum_{i,j} \eta_{ij}$ is the objective function value in the tth iteration. $\Delta J^{(t-1)}$ is the difference between the objective function values in two successive iterations, i.e., $\Delta J^{(t-1)} = J^{(t-2)} - J^{(t-1)}$. $|J^{(t-1)}|$ and $|J^{(t-2)}|$ are the absolute values of $J^{(t-1)}$ and $J^{(t-2)}$, respectively. J^* is the maximum value of $|J^{(t-2)}|$ and $|J^{(t-1)}|$.

We employ the same CCCP stopping criterion as in [27], [28] to determine the termination of SDB-MIL, i.e., when the proportion of $\Delta J^{(t-1)}$ and J^* is smaller than a threshold ϵ, the algorithm stops. As in [27], [28], ϵ is set to be 0.01.

4 Experiments

We investigate the performance of SDB-MIL on benchmark and real-world MIL datsets. The benchmark MIL datsets include the Mutagenesis, Musk, Corel, Reuters, Tiger, Elephant and Fox datasets, which are popularly used in MIL studies [3], [8], [30], [31]. The Corel and Reuters datasets are not designed for MIL, and we follow the same routine in [8], [23], [28] to form the sub-datasets, as shown in Table 2. The characteristic of these datasets is that the negative data in the training set cannot sufficiently represent the distribution of negative data in the testing set. Taking Corel 0 subdataset as an example, the negative data in the training set contains class 2, 3, 4, 5, 7, 8, 11, 17 and 18, while the negative class in the testing set includes class 1 to 19. That is to say, class 1, 6, 9, 10, 12, 13, 14, 15, 16 and 19 are in the testing set, but not in the training set. The negative data in the training set cannot appropriately describe the distribution of

TABLE 1

Algorithm: The overview of SDB-MIL algorithm

```
Input:
1. Training set: $m_1$ positive bags ($B_1 \ldots, B_{m_1}$) and $m_2$ negative bags ($B_{m_1+1} \ldots, B_{m_1+m_2}$);
2. Regularization parameters $C$, $C_1$ and $C_2$, class size balance $l$, CCCP precision $\epsilon$;
Output: Predicted bag label $Y_i$;
CCCP Iterations:
1: Initialize $\Delta J^{(-1)} = 10^{-3}$, $J^* = 10^{-3}$;
2: Give initial values of $\mathbf{o}^{(0)}$, $s^{(0)}$, $\rho^{(0)}$;
3: Set $t = 0$;
4: while $\Delta J^{(t-1)}/J^* > \epsilon$ do
5:    Decompose $g(\mathbf{o}, s, i)$ at ($\mathbf{o}^{(t)}$, $s^{(t)}$, $\rho^{(t)}$) as
        $g(\mathbf{o}, s, i) \approx \sum_{j\in B_i} \theta_{ij}^{(t)}(s - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij})$;
6:    Decompose $h(\mathbf{o}, s, \rho, i, j)$ at ($\mathbf{o}^{(t)}$, $s^{(t)}$, $\rho^{(t)}$) as
        $h(\mathbf{o}, s, \rho, i, j) \approx b_{ij}^{(t)}(s - 2\mathbf{o}^T B_{ij} + (B_{ij})^T B_{ij})$
7:    Derive problem (12) by replacing $g(\mathbf{o}, s, i)$ and
      $h(\mathbf{o}, s, \rho, i, j)$ with the decompositions in lines 5 and 6;
8:    Obtain the dual form of problem (12) as (22);
9:    Get ($\mathbf{o}^{(t+1)}$, $s^{(t+1)}$, $\rho^{(t+1)}$) by solving (22);
10:   $t = t + 1$;
11:   $J^* = \max\{|J^{(t-2)}|, |J^{(t-1)}|\}$;
12:   $\Delta J^{(t-1)} = J^{(t-2)} - J^{(t-1)}$;
13: end while
14: Label Prediction:
For bag $B_i$, $Y_i = sgn(R^2 - \min_{j\in B_i} ||B_{ij} - \mathbf{o}||^2)$;
```

negative data in the testing set. Besides the benchmark datasets, we collect a new real-world MIL dataset for image categorization and conduct experiments to evaluate our method on it. All the experiments run on the Windows platform with 2.8 GHz processor and 3 GB DRAM.

The precision, recall and F-measure values are used as evaluation metrics to investigate the performance of our proposed method. The precision value is calculated by $TP/(TP + FP)$ and the recall value is computed from $TP/(TP + FN)$. Here, TP is the number of bags whose true label is 1 and the predicted label is also 1; FP is the number of bags whose true label is -1, but the predicted label is 1; FN is the number of bags whose true label is 1, but the predicted label is -1. Based on these, the F-measure value is defined as $2 * precision * recall/(precision + recall)$.

For comparison, mi-SVM [5], MI-SVM [5], SVDD-NEG [25], MIOptimalBall [6], EM-DD [18] and DD-SVM [8] are utilized as baselines. Considering that SVDD-NEG [25] is originally designed for single-instance learning, it is extended to multi-instance learning by making the instance label the same with its bag label. Among these methods, mi-SVM and MI-SVM build up the classifier by only maximizing the margin between the positive and negative data. These baselines are used to evaluate the performance of sphere-based classifiers in comparison with the traditional margin-based classifiers when the negative training data cannot sufficiently represent the distribution of negative data in the testing set. SVDD-NEG and MIOptimalBall construct sphere-based (ball-based) classifiers to separate the data. They are utilized to investigate the improvement of SDB-MIL over the existing sphere-based classifiers. EM-DD applies a diverse density measurement to determine a target concept and DD-SVM maps a bag of instances into a bag-level vector. They are used to test SDB-MIL compared to the other representative MIL classifiers.

The RBF kernel is employed for the SVM-based algorithms, i.e., mi-SVM, MI-SVM, DD-SVM, SVDD-NEG and SDB-MIL. As in [3], [18], [24], the Euclidean distance is used to measure the distance between two instances for EM-DD and MIOptimalBall. The kernel parameter in the RBF kernel is chosen from $10^{-5:1:5}$. For simplicity, we make the regularization parameters in (12) as $C = C_1 = C_2$ and select the value from $2^{-5:1:5}$. Following [32], the class balance parameter l is selected from the grid $\{0, 0.001n, 0.01n, 0.1n, n\}$, where n is the instance number in the training set. $\mathbf{o}^{(0)}$, $s^{(0)}$ and $\rho^{(0)}$ are randomly initialized. ϵ is set to be 0.01.

The source codes of MIOptimalBall, DD-SVM, EM-DD, mi-SVM, and MI-SVM are publicly available. MIOptimalball is included in the WEKA software which can be downloaded from http://www.cs.waikato.ac.nz/ml/weka/. DD-SVM is available at http://www.cs.olemiss.edu/~ychen/ddsvm.html. EM-DD, mi-SVM and MI-SVM can be downloaded from http://prlab.tudelft.nl/david-tax/mil.html. Moreover, SDB-MIL is available at https://github.com/syxiao2/SDBMIL.

4.1 Evaluations on Benchmark Datasets

4.1.1 Mutagenesis Dataset

The Mutagenesis dataset[1] consists of three sub-datasets: Muta-atoms, Muta-bonds and Muta-chains. These sub-datasets have different multi-instance representations. In the Muta-atoms sub-dataset, a bag contains all the atoms of a compound molecule. In the Muta-bonds sub-dataset, a bag includes all the atom-bond tuples of a compound molecule. In the Muta-chains sub-dataset, a bag consists of all the adjacent pairs in bonds of a compound molecule. Each of the three sub-datasets has 125 mutagenic bags and 63 non-mutagenic bags. The Muta-atoms sub-dataset has 8.6 instances per bag and three features per instances. The Muta-bonds sub-dataset has 21.25 instances per bag on average and each instance has seven features. The Muta-chains sub-dataset has averagely 28.45 instances per bag and 11 features per instance. Table 2 presents the categories in the training and testing sets.

For each sub-dataset, the mutagenic and non-mutagenic bags are randomly partitioned in half. One is used as the training set and the other is as the testing set. As in [8], [33], [34], the optimal parameters are selected by conducting 10-fold cross validation on the training set. After the parameters are selected, the results on the testing set are recorded. Each experiment is repeated for 10 times with different random splits, and the testing result is reported. Tables 3 and 4 report the F-measure, precision and recall values on the Muta-atoms, Muta-bonds and Muta-chains sub-datasets. In addition, Table 3 also reports the p-values, which are computed by performing the paired t-test comparing all other classifiers to SDB-MIL under the null hypothesis that there is no difference between the F-measure values. When the p-value is smaller than the confidence level 0.05, there is a significant difference between SDB-MIL and the method compared.

[1] Available at http://www.cs.waikato. ac.nz/ml/proper/datasets.html

TABLE 2
The Categories Contained in the Training and Testing Sets

Dataset	Training Set		Testing Set	
	Positive Class	Negative Class	Positive Class	Negative Class
Muta-atoms	Mutagenic	Non-mutagenic	Mutagenic	Non-mutagenic
Muta-bonds	Mutagenic	Non-mutagenic	Mutagenic	Non-mutagenic
Muta-chains	Mutagenic	Non-mutagenic	Mutagenic	Non-mutagenic
Musk 1	Musk	Non-musk	Musk	Non-musk
Musk 2	Musk	Non-musk	Musk	Non-musk
Corel 0	0	2,3,4,5,7,8,11,17,18	0	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19
Corel 1	1	0,2,4,5,7,8,9,11,12,13,14,15	1	0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19
Corel 2	2	4,7,8,9,11,13,14,17,19	2	0,1,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19
Corel 3	3	1,5,6,7,12,13,15,17	3	0,1,2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19
Corel 4	4	3,5,8,9,10,12,16,17	4	0,1,2,3,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19
Corel 5	5	1,3,4,9,10,15,19	5	0,1,2,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18,19
Corel 6	6	2,3,7,8,9,14,15,16	6	0,1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19
Corel 7	7	0,5,6,8,9,10,11,16,18,19	7	0,1,2,3,4,5,6,8,9,10,11,12,13,14,15,16,17,18,19
Corel 8	8	1,6,7,11,14,15,19	8	0,1,2,3,4,5,6,7,9,10,11,12,13,14,15,16,17,18,19
Corel 9	9	2,5,6,8,11,12,14,18,19	9	0,1,2,3,4,5,6,7,8,10,11,12,13,14,15,16,17,18,19
Corel 10	10	1,2,4,5,9,13,15,16,18,19	10	0,1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19
Corel 11	11	1,2,8,9,10,12,15,18	11	0,1,2,3,4,5,6,7,8,9,10,12,13,14,15,16,17,18,19
Corel 12	12	1,2,3,4,5,9,10,11,13,14,18,19	12	0,1,2,3,4,5,6,7,8,9,10,11,13,14,15,16,17,18,19
Corel 13	13	0,1,3,4,6,7,8,10,11,17	13	0,1,2,3,4,5,6,7,8,9,10,11,12,14,15,16,17,18,19
Corel 14	14	1,4,5,9,10,11,13,19	14	0,1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19
Corel 15	15	3,5,6,7,14,18,19	15	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19
Corel 16	16	2,3,5,10,13,15,17	16	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,17,18,19
Corel 17	17	0,2,3,5,7,14,15,18	17	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,18,19
Corel 18	18	3,4,6,7,8,10,11,14,16	18	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19
Corel 19	19	0,2,3,4,10,11,13,15,16,18	19	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18
Reuters 1	1	2,4,5	1	2,3,4,5,6,7
Reuters 2	2	1,6,7	2	1,3,4,5,6,7
Reuters 3	3	1,4	3	1,2,4,5,6,7
Reuters 4	4	3,5,7	4	1,2,3,5,6,7
Reuters 5	5	1,4,6,7	5	1,2,3,4,6,7
Reuters 6	6	2,4	6	1,2,3,4,5,7
Reuters 7	7	2,3,5,6	7	1,2,3,4,5,6
Tiger	Tiger	Other animals	Tiger	Other animals
Elephant	Elephant	Other animals	Elephant	Other animals
Fox	Fox	Other animals	Fox	Other animals

The characteristic of the above sub-datasets is that the negative training data cannot sufficiently represent the distribution of negative data in the testing set. Taking Corel 0 sub-dataset as an example, the negative class in the training set contains class 2, 3, 4, 5, 7, 8, 11, 17 and 18, while the negative class in the testing set includes class 1 to 19. That is to say, class 1, 6, 9, 10, 12, 13, 14, 15, 16 and 19 are in the testing set, but not in the training set.

It is seen that SDB-MIL outperforms the sphere-based classifiers SVDD-NEG and MIOptimalBall, and meanwhile attains the best F-measure values on all the Mutagenesis sub-datasets. In contrast with SVDD-NEG, SDB-MIL delivers markedly better F-measure values. The F-measure values of SDB-MIL on the Muta-atoms, Muta-bonds and Muta-chains sub-datasets are 0.828, 0.842 and 0.795, respectively, which are statistically better than SVDD-NEG, as the p-values are smaller than 0.05. This is because SVDD-NEG is designed for single-instance learning, where each instance is explicitly associated with a label. Nevertheless, in multi-instance learning, the labels of instances in positive bags are unknown. If SVDD-NEG is adapted to MIL by making all instances in positive bags as positive, mislabelling may occur since the positive bag may contain negative instances, in addition to positive instances.

Moreover, compared with MIOptimalBall, SDB-MIL shows an explicit improvement on classification performance. There is a significant difference in F-measure values between SDB-MIL and MIOptimalBall on the Mutagenesis sub-dataset, as seen from the p-values. MIOptimalBall is a sphere-based classifier which is proposed for multi-instance learning. However, MIOptimalBall learns a relatively loose ball which may not guarantee a proper description of the data boundary. Hence, there may exist some instances which are negative, but misclassified to the positive class by the loose ball. This is especially the case when the negative training data cannot appropriately represent the distribution of negative testing data.

4.1.2 Musk Dataset

The Musk dataset[2] has two sub-datasets: Musk1 and Musk2. The Musk1 sub-dataset contains 47 positive bags and 45 negative bags with about 5.17 instances per bag. The number of instances per bag varies from 2 to 40. The Musk2 sub-dataset includes 39 positive bags and 63 negative bags with around 64.69 instances per bag. The number of instances in each bag ranges from 1 to 1,044. Each instance is represented by a 166-dimensional feature vector.

As shown in Tables 3 and 4, SDB-MIL exhibits the best classification performance on the Musk1 sub-dataset. The

[2] Available at http://kdd.ics.uci.edu

TABLE 3
The F-Measure Values on the Benchmark Datasets

	EM-DD	DD-SVM	mi-SVM	MI-SVM	SVDD-NEG	MIOptimalBall	SDB-MIL
Muta-atoms	0.795* (*0.014*)	0.801* (*0.037*)	0.807* (*0.028*)	0.802* (*0.019*)	0.796* (*< 0.01*)	0.809* (*0.028*)	**0.828**
Muta-bonds	0.806* (*0.012*)	0.811* (*0.024*)	0.804* (*< 0.01*)	0.829* (*0.023*)	0.798* (*< 0.01*)	0.805* (*0.035*)	**0.842**
Muta-chains	0.771* (*0.021*)	0.782* (*0.044*)	0.779* (*0.036*)	0.754* (*0.015*)	0.769* (*0.011*)	0.769* (*0.029*)	**0.795**
Musk 1	0.805* (*0.017*)	0.857* (*0.026*)	0.871* (*0.029*)	0.802* (*< 0.01*)	0.774* (*< 0.01*)	0.782* (*< 0.01*)	**0.898**
Musk 2	0.792* (*0.012*)	**0.876*** (*0.032*)	0.798* (*0.018*)	0.806* (*0.025*)	0.758* (*< 0.01*)	0.687* (*< 0.01*)	0.838
Corel 0	0.634* (*0.013*)	0.611* (*< 0.01*)	0.652* (*0.028*)	0.661* (*0.042*)	0.623* (*0.012*)	0.645* (*0.036*)	**0.675**
Corel 1	0.676* (*0.048*)	0.640* (*0.019*)	0.663* (*0.037*)	0.656* (*0.026*)	0.635* (*0.015*)	0.646* (*0.025*)	**0.698**
Corel 2	0.702* (*0.016*)	0.720* (*0.023*)	0.737* (*0.034*)	0.742* (*0.042*)	0.727* (*0.032*)	0.732* (*0.024*)	**0.762**
Corel 3	0.830* (*0.012*)	0.846* (*0.025*)	0.834* (*0.014*)	0.843* (*0.032*)	0.840* (*0.023*)	0.850* (*0.037*)	**0.877**
Corel 4	0.939* (*< 0.01*)	0.953* (*0.015*)	0.965* (*0.027*)	0.972* (*0.027*)	0.968* (*0.031*)	0.972* (*0.034*)	**0.995**
Corel 5	0.670* (*0.027*)	0.676* (*0.021*)	0.670* (*0.015*)	0.683* (*0.033*)	0.693* (*0.012*)	0.681* (*0.023*)	**0.718**
Corel 6	0.706* (*0.015*)	0.701* (*< 0.01*)	0.734* (*0.048*)	0.722* (*0.037*)	0.710* (*0.022*)	0.719* (*0.026*)	**0.751**
Corel 7	0.888* (*0.023*)	0.872* (*0.015*)	0.889* (*0.033*)	0.884* (*0.024*)	0.878* (*0.021*)	0.891* (*0.038*)	**0.910**
Corel 8	0.639* (*0.012*)	0.615* (*0.018*)	0.620* (*0.032*)	0.615* (*0.024*)	0.626* (*0.032*)	0.617* (*0.025*)	**0.660**
Corel 9	0.818* (*< 0.01*)	0.823* (*0.019*)	0.843* (*0.036*)	0.838* (*0.032*)	0.814* (*0.024*)	0.828* (*0.031*)	**0.862**
Corel 10	0.566* (*0.042*)	0.578 (*0.366*)	0.580 (*0.348*)	**0.609** (*0.152*)	0.571* (*0.046*)	0.566* (*0.038*)	0.585
Corel 11	0.716* (*0.028*)	0.718* (*0.015*)	0.721* (*0.035*)	0.706* (*0.011*)	0.698* (*< 0.01*)	0.712* (*0.026*)	**0.746**
Corel 12	0.538* (*0.023*)	0.546* (*0.032*)	**0.578** (*0.239*)	0.562* (*0.041*)	0.545* (*0.034*)	0.548* (*0.038*)	0.575
Corel 13	0.491* (*0.032*)	0.471* (*0.014*)	0.475* (*0.017*)	0.477* (*0.025*)	0.476* (*0.027*)	0.457* (*< 0.01*)	**0.521**
Corel 14	0.561* (*0.021*)	0.575* (*0.028*)	0.571* (*0.024*)	0.564* (*0.017*)	0.554* (*0.013*)	0.583* (*0.045*)	**0.603**
Corel 15	0.594* (*< 0.01*)	0.627* (*0.021*)	0.633* (*0.033*)	0.619* (*0.023*)	0.609* (*0.011*)	0.611* (*0.012*)	**0.658**
Corel 16	0.751* (*0.018*)	0.782* (*0.041*)	0.765* (*0.034*)	0.774* (*0.039*)	0.752* (*0.012*)	0.758* (*0.027*)	**0.802**
Corel 17	0.684* (*0.023*)	0.680* (*0.015*)	0.706* (*0.037*)	0.689* (*0.029*)	0.682* (*0.017*)	0.695* (*0.025*)	**0.722**
Corel 18	0.513* (*< 0.01*)	0.519* (*< 0.01*)	0.558* (*0.025*)	0.564* (*0.021*)	0.541* (*0.011*)	0.553* (*0.019*)	**0.599**
Corel 19	0.624* (*0.017*)	0.634* (*0.023*)	0.647* (*0.028*)	0.664* (*0.035*)	0.629* (*0.014*)	0.665* (*0.042*)	**0.684**
Reuters 1	0.695* (*0.022*)	0.671* (*< 0.01*)	0.685* (*0.026*)	0.701* (*0.037*)	0.683* (*0.018*)	0.703* (*0.036*)	**0.731**
Reuters 2	0.443* (*0.026*)	0.464* (*0.035*)	0.456* (*0.038*)	0.473* (*0.045*)	0.451* (*0.029*)	0.454* (*0.031*)	**0.486**
Reuters 3	0.468* (*< 0.01*)	0.480* (*0.012*)	0.495* (*0.026*)	0.485* (*0.017*)	0.503* (*0.022*)	0.503* (*0.036*)	**0.540**
Reuters 4	0.605* (*0.021*)	0.615* (*0.026*)	0.631* (*0.035*)	0.631* (*0.033*)	0.611* (*0.015*)	0.624* (*0.029*)	**0.655**
Reuters 5	0.491* (*0.035*)	0.504* (*0.046*)	0.528 (*0.387*)	**0.536** (*0.065*)	0.491* (*0.038*)	0.503* (*0.048*)	0.520
Reuters 6	0.597* (*0.014*)	0.621* (*0.031*)	0.614* (*0.023*)	0.623* (*0.035*)	0.601* (*0.016*)	0.610* (*0.024*)	**0.647**
Reuters 7	0.485* (*0.026*)	0.493* (*0.029*)	0.513* (*0.044*)	0.501* (*0.036*)	0.485* (*0.013*)	0.496* (*0.027*)	**0.531**
Tiger	0.744* (*0.018*)	0.739* (*0.013*)	0.757* (*0.022*)	0.773* (*0.029*)	0.718* (*0.016*)	0.652* (*0.014*)	**0.804**
Elephant	0.772* (*0.015*)	0.789* (*0.037*)	0.798* (*0.039*)	0.751* (*0.024*)	0.747* (*0.013*)	0.742* (*0.011*)	**0.822**
Fox	0.527* (*0.021*)	0.552* (*0.031*)	0.556* (*0.027*)	0.563* (*0.038*)	0.515* (*0.019*)	0.437* (*< 0.01*)	**0.598**

F-measure value of SDB-MIL is 0.898, which is higher than EM-DD (0.805), DD-SVM (0.857), mi-SVM (0.871), MI-SVM (0.802), SVDD-NEG (0.774) and MIOptimalBall (0.782). On the Musk2 sub-dataset, SDB-MIL obtains the F-measure value at 0.838, which is statistically better than most baselines, except for DD-SVM. Although DD-SVM gives a higher F-measure value at the Musk2 sub-dataset, SDB-MIL outperforms DD-SVM on the other experimental datasets.

4.1.3 Corel Dataset

The Corel dataset contains 20 categories and each category has 100 images. To transform the image data into MIL data, each image is segmented into a number of regions [8], [23]. Each image is considered as a bag and each region is regarded as an instance.

Since the Corel dataset contains multiple categories, we conduct the following operations to obtain the experimental datasets. First, as in [8], [23], we choose one category as the positive class in turn, and the remaining 19 categories are considered as the negative class. Hence, twenty sub-datasets (Corel 0 to Corel 19) are obtained. Second, we form the training and testing sets for each sub-dataset. Each sub-dataset has one category in the positive class and 19 categories in the negative class. In the existing MIL work [8], [21], [23], [35], the negative training data is formed by selecting the images uniformly from each of the 19 categories in the negative class, so that the negative training data can appropriately describe the topics of negative images in the testing set. However, in real-world MIL applications, the negative training data may not sufficiently represent the distribution of negative testing data. There may be some image topics which are represented in the negative images of the testing set, but not described in the training set. To bring our experiments one step closer to real-world MIL settings, for the 19 categories in the negative class, we randomly divide them into two groups and put them into the subsets S_{both} and S_{test}, respectively. The training set is formed by selecting 50 percentage of bags from the positive class and each category of S_{both}. The remaining bags in the positive class and S_{both}, as well as all the bags in S_{test}, are used to form the testing set. In this way, the categories of S_{test} are only in the testing set, but not in the training set. Table 2 shows the categories contained in the training and testing sets for the Corel sub-datasets.

Tables 3 and 4 present the results on the Corel 0 to Corel 19 sub-datasets. It is seen that SDB-MIL obtains better F-measure values than the baselines on 18 out of 20 sub-datasets. On the one hand, SDB-MIL outperforms the

TABLE 4
The Precision and Recall Values on the Benchmark Datasets

	EM-DD		DD-SVM		mi-SVM		MI-SVM		SVDD-NEG		MIOptimalBall		SDB-MIL	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Muta-atoms	0.813	0.783	0.829	0.775	0.846	0.772	0.826	0.781	0.845	0.755	0.833	0.786	0.847	0.810
Muta-bonds	0.811	0.802	0.822	0.798	0.821	0.790	0.801	0.860	0.816	0.783	0.804	0.808	0.823	0.863
Muta-chains	0.793	0.751	0.813	0.753	0.799	0.760	0.782	0.729	0.808	0.733	0.774	0.765	0.815	0.775
Musk 1	0.766	0.956	0.782	0.947	0.794	0.963	0.692	0.954	0.651	0.952	0.670	0.941	0.828	0.981
Musk 2	0.833	0.754	0.965	0.801	0.755	0.846	0.764	0.854	0.720	0.802	0.772	0.618	0.743	0.962
Corel 0	0.559	0.732	0.547	0.695	0.558	0.785	0.560	0.805	0.553	0.714	0.558	0.771	0.573	0.817
Corel 1	0.617	0.751	0.579	0.721	0.603	0.735	0.595	0.732	0.595	0.685	0.587	0.719	0.594	0.846
Corel 2	0.661	0.753	0.654	0.802	0.653	0.847	0.645	0.876	0.649	0.827	0.627	0.879	0.669	0.886
Corel 3	0.926	0.754	0.934	0.773	0.904	0.779	0.932	0.771	0.916	0.775	0.930	0.784	0.959	0.807
Corel 4	0.928	0.953	0.929	0.977	0.935	0.999	0.951	0.994	0.962	0.973	0.978	0.967	0.997	0.995
Corel 5	0.665	0.673	0.644	0.716	0.650	0.695	0.654	0.715	0.675	0.712	0.674	0.688	0.683	0.757
Corel 6	0.833	0.623	0.815	0.615	0.802	0.677	0.778	0.675	0.783	0.654	0.801	0.652	0.839	0.680
Corel 7	0.854	0.924	0.866	0.877	0.862	0.917	0.846	0.930	0.853	0.907	0.865	0.917	0.872	0.955
Corel 8	0.689	0.595	0.690	0.557	0.702	0.558	0.692	0.553	0.684	0.577	0.671	0.573	0.710	0.617
Corel 9	0.865	0.775	0.873	0.778	0.870	0.817	0.864	0.814	0.863	0.773	0.871	0.789	0.873	0.851
Corel 10	0.541	0.595	0.547	0.613	0.559	0.601	0.574	0.655	0.534	0.613	0.539	0.597	0.566	0.606
Corel 11	0.763	0.675	0.772	0.671	0.776	0.673	0.767	0.654	0.771	0.637	0.774	0.659	0.778	0.717
Corel 12	0.526	0.551	0.521	0.573	0.552	0.609	0.533	0.595	0.514	0.579	0.509	0.592	0.539	0.615
Corel 13	0.453	0.537	0.456	0.487	0.449	0.503	0.461	0.497	0.444	0.513	0.443	0.471	0.493	0.553
Corel 14	0.610	0.519	0.593	0.557	0.588	0.556	0.578	0.551	0.595	0.517	0.595	0.571	0.621	0.585
Corel 15	0.617	0.573	0.604	0.653	0.611	0.658	0.616	0.623	0.602	0.615	0.586	0.639	0.624	0.697
Corel 16	0.667	0.859	0.665	0.951	0.659	0.917	0.665	0.928	0.661	0.875	0.661	0.887	0.670	0.999
Corel 17	0.659	0.711	0.630	0.737	0.650	0.773	0.633	0.757	0.656	0.709	0.636	0.769	0.668	0.787
Corel 18	0.590	0.453	0.604	0.456	0.612	0.513	0.599	0.533	0.592	0.498	0.601	0.513	0.625	0.575
Corel 19	0.733	0.547	0.708	0.576	0.712	0.595	0.723	0.615	0.701	0.571	0.734	0.613	0.741	0.635
Reuters 1	0.675	0.716	0.655	0.687	0.645	0.733	0.648	0.785	0.655	0.717	0.665	0.745	0.678	0.793
Reuters 2	0.481	0.411	0.491	0.439	0.479	0.437	0.501	0.453	0.498	0.413	0.472	0.437	0.522	0.455
Reuters 3	0.492	0.446	0.485	0.475	0.503	0.487	0.497	0.473	0.508	0.497	0.512	0.495	0.532	0.547
Reuters 4	0.576	0.638	0.583	0.652	0.592	0.677	0.582	0.689	0.571	0.657	0.580	0.676	0.604	0.715
Reuters 5	0.532	0.457	0.545	0.469	0.566	0.495	0.601	0.485	0.539	0.453	0.542	0.471	0.576	0.475
Reuters 6	0.584	0.613	0.592	0.653	0.592	0.635	0.594	0.654	0.573	0.631	0.581	0.650	0.622	0.673
Reuters 7	0.495	0.475	0.490	0.497	0.497	0.531	0.489	0.515	0.483	0.487	0.479	0.515	0.513	0.551
Tiger	0.694	0.801	0.713	0.766	0.732	0.783	0.757	0.789	0.704	0.732	0.719	0.597	0.794	0.814
Elephant	0.758	0.787	0.804	0.775	0.794	0.802	0.710	0.797	0.778	0.718	0.766	0.719	0.833	0.812
Fox	0.629	0.453	0.747	0.438	0.765	0.437	0.732	0.458	0.716	0.402	0.511	0.382	0.796	0.479

sphere-based classifiers SVDD-NEG and MIOptimalBall on all the sub-datasets, which is consistent with the observations in the Mutagenesis and Musk datasets. On the other hand, SDB-MIL obtains statistically better classification performance than the margin-based classifiers mi-SVM and MI-SVM on most of the sub-datasets. Comparing SDB-MIL with mi-SVM, there is a significant difference in F-measure values on 18 out of 20 sub-datasets. Moreover, the F-measure value of SDB-MIL is statistically better than MI-SVM for all cases except for the Corel 10 sub-dataset. The better performance of SDB-MIL over mi-SVM and MI-SVM indicates that when the negative bags in the training set could not sufficiently depict the distribution of negative data in the testing set, it may be more desirable to build up the classifier by enclosing the positive data in an optimal sphere, rather than only maximizing the margin between two classes, since the classification boundary may be biased due to the incomplete negative training information.

4.1.4 Reuters Dataset

This text categorization dataset[3] is from Reuters-21578 collection and originally proposed for multi-label multi-instance learning. Each document is represented as a bag of instances using the sliding window techniques, where each instance corresponds to a text segment enclosed in one sliding window of size 50 (overlapped with 25 words). "Function words" on the SMART stop-list [36] are removed from the vocabulary and the remaining words are stemmed. Instances in the bags adopt the "Bag-of-Words" representation based on term frequency. Without loss of effectiveness, dimensionality reduction is performed by retaining the top 2 percent words with highest document frequency. Hence, each instance is represented as a 243-dimensional vector.

It contains 2,000 bags and each bag includes a number of instances. The instance numbers in bags vary from 2 to 26 with 3.56 on average. Each bag is associated with at most seven labels. We follow the same strategy in [28], [37], [38] to generate the single-label MIL dataset. Specifically, we remove the documents which are associated with more than one label from the dataset. As a result, we obtain seven sub-datasets, i.e., Reuters 1 to Reuters 7, with 1,700 bags left in total. The document in each sub-dataset is associated with one label.

The same as the Corel dataset, each category is considered as the positive class in turn and the other categories are regarded as the negative class. Thereafter, seven sub-datasets (Reuters 1 to Reuters 7) are obtained. Each

[3] Available at http://lamda.nju.edu.cn/data_MIMLtext.ashx

sub-dataset contains one category in the positive class and six categories in the negative class. For the six categories in the negative class, we randomly partition them into two groups and let them contained in the subsets S_{both} and S_{test}, respectively. For each sub-dataset, the training set is formed by randomly sampling half of the bags in the positive class and each category of S_{both}. Together with the remaining bags in the positive class and S_{both}, the bags in S_{test} are used to generate the testing set. Table 2 shows the categories contained in the training and testing sets for the Reuters sub-datasets.

Tables 3 and 4 show the classification results. It is found that SDB-MIL attains the best F-measure value on 6 out of 7 sub-datasets. In particular, SDB-MIL is very competitive with EM-DD and DD-SVM. The F-measure value of SDB-MIL is statistically better than EM-DD and DD-SVM on all the Reuters sub-datasets. EM-DD and DD-SVM are based on the computation of DD values, which are calculated by using the instances from the positive and negative bags. When the negative bags in the training set cannot sufficiently represent the distribution of negative data in the testing set, the obtained DD values may be corrupted and could not reflect the exact distribution information of the testing set.

4.1.5 Tiger, Elephant and Fox Image Datasets

The Tiger, Elephant and Fox datasets which are generated by Stuart Andrews et al. [5], are proposed for image annotation. In the Tiger dataset, the positive class is formed by selecting 100 images from tiger, and the negative class is formed by choosing 100 images from other animals, such as bear, cat, coyote, sheep, lion, eagle, zebra, snake, cougar, antelope and so on. The Elephant and Fox datasets are formed in the similar way to the Tiger dataset. Each image is considered as a bag and each region is treated as an instance. As a result, each dataset contains 100 positive bags and 100 negative bags.

Table 3 shows the F-measure values and p-values on the Tiger, Elephant and Fox datasets, and Table 4 presents the precision and recall values. It is seen that SDB-MIL attains significantly better improvements over the baselines on the three datasets. For example, on the Tiger dataset, the F-measure value of SDB-MIL is 0.804, which is significantly better than EM-DD (0.744), DD-SVM (0.739), mi-SVM (0.757), MI-SVM (0.773), SVDD-NEG (0.718), MIOptimalBall (0.652). In the three datasets, e.g., the Tiger dataset, the negative bags contain a number of negative topics, such as bear, cat, coyote, sheep, lion, eagle, zebra, snake, cougar, antelope and so on. The number of negative bags in the training set is usually limited and may not sufficiently represent each of the negative topics. Compared to the negative bags, the positive bags include only one positive topic - tiger, which is easier to be described. Hence, in MIL, when the negative data in the training set may not sufficiently represent each of the negative topics, it is more desirable to build the classifier by enclosing the positive data in a sphere, rather than separating the positive and negative data using a plane.

4.2 Evaluations on Real-World Dataset

In the above, experiments are conducted to evaluate SDB-MIL on the benchmark datasets popularly used in the existing MIL work. In this section, we collect a new real-world MIL dataset and investigate the performance of SDB-MIL on it.

In MIL, the positive bag contains at least one positive instance, which is relevant to the user interest. In the negative bag, all instances are negative and irrelevant to the user interest. Based on this description of MIL, we collect a new MIL dataset for image categorization. Assume that the user is interested in the images relevant to “dog”. In the Internet, we download 1,000 images relevant to “dog”, which are of the user interest and considered as positive bags. Moreover, we randomly collect 1,000 images irrelevant to “dog”, which are not of the user interest and treated as negative bags. The classification task is to separate the positive data relevant to “dog” and the negative data irrelevant to “dog”.

We follow the same routine in [3], [5], [7], [14], [19], [35] to transform the 2,000 images into MIL data. Specifically, each image is segmented into several regions by employing the Blobworld system [39], which can automatically segment the images and require no parameter tuning of regions. Then, the segmented results are converted into MIL data by conducting the codes[4] provided by Andrews et al. [5]. In the obtained MIL dataset, each bag has 7.8 instances on average and each instance has 230 dimensions, representing the color, texture and shape information. More detailed descriptions of the data features can refer to the documentation within Stuart Andrews’s codes. The transformed MIL data and original images of our new dataset can be downloaded from http://www.dmirlab.com/xiaoyanshan/datasets.html.

Similar to the benchmark datasets, the training set is formed by randomly selecting 500 positive bags and 500 negative bags, and the remaining 1,000 bags are used for testing. The experiment is repeated for 10 times with different random splits. The testing result is reported. Fig. 3 shows the images randomly sampled from the training set and the testing set.

It is relatively easy to specify the topic shared in the positive images, since the positive images are related to the targeted topic that the user is of interest. Figs. 3a and 3b show the positive images in the training set and the testing set, respectively. It can be observed that the positive images in the training set are relevant to “dog”, and those in the testing set are also relevant to “dog”. Hence, the positive images in the training set and the testing set are associated with the same topic “dog” which is of the user interest.

However, compared to the positive images, it is difficult to describe the specific topics underneath the negative images, since they may involve hundreds of thousands of topics which are irrelevant to the user interest. In the new dataset, an image is labeled as negative if it is irrelevant to the targeted topic - “dog”. In the Internet, there are all kinds of images and the number of image topics irrelevant to “dog” can be extensively large. It is difficult for us to determine all the topics irrelevant to “dog”. Moreover, even if all the topics irrelevant to “dog” can be identified, it is too expensive to collect such a large quantity of images to represent each topic.

[4] Available at http://www.cs.columbia.edu/~andrews/mil/data/MIL-Data-2002-Musk-Corel-Trec9.tgz

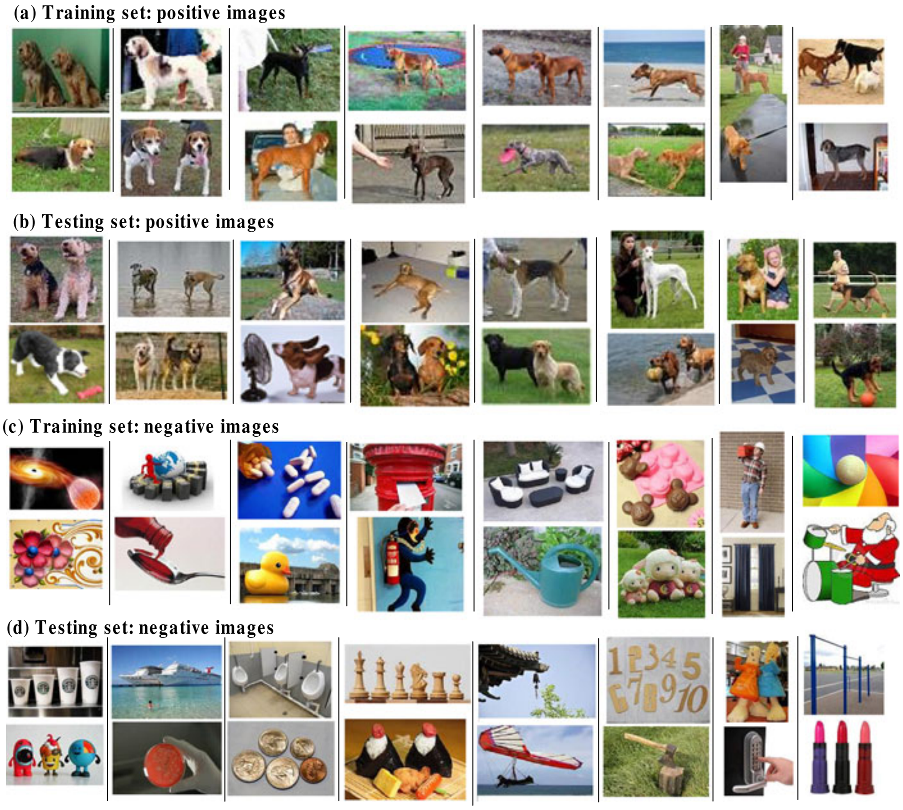

(a) Training set: positive images
(b) Testing set: positive images
(c) Training set: negative images
(d) Testing set: negative images

Fig. 3. Positive and negative images randomly sampled from the training set and the testing set. (1) The images in (a) and (b) are relevant to the targeted topic - “dog” and thus are treated as positive. We can observe that it is relatively easy to describe the topic shared in the positive images of the training set (in (a)) and those of the testing set (in (b)), since both of them are relevant to the targeted topic - “dog”. (2) The images in (c) and (d) are irrelevant to “dog” and thus are considered as negative. It is seen that there are some topics which appear in the testing set (in (d)), but not in the training set (in (c)). For example, the first two images in the testing set (in (d)) are related to “cup” and “ship”, respectively. However, these two topics are not represented in the negative images of the training set (in (c)). In the Internet, there are hundreds of thousands of image topics which are irrelevant to “dog”. However, the number of negative images in the training set is usually limited and can represent only a part of negative topics irrelevant to “dog”. As a result, there may be some negative topics which appear in the testing set, but not in the training set. When the negative training data cannot sufficiently represent the topics of negative data in the testing set, it is more appropriate to build up the classifier by enclosing the positive data, whose topic is relatively explicit and easy to be described, in an optimal sphere, and excluding the negative data outside the sphere.

As a result, the negative images collected for training the classifier may cover only a part of negative topics irrelevant to “dog”, and cannot appropriately represent all the negative topics in the testing set. Figs. 3c and 3d show the negative images in the training set and the testing set, respectively. On the one hand, it is seen that the images in the training set (in Fig. 3c) and those in the testing set (in Fig. 3d) are irrelevant to the targeted topic - “dog”, and thus considered as negative. On the other hand, we can observe that there are some topics which are in the testing set, but not in the training set. For example, the first two images in the testing set (in Fig. 3d) are related to “cup” and “ship”, respectively, but these two topics are not represented in the negative images of the training set (in Fig. 3c). In the Internet, there are hundreds of thousands of image topics which are irrelevant to “dog”. The 500 negative images in the training set can represent only a small number of the negative topics irrelevant to “dog”. Hence, there may be some negative topics (e.g., “cup” and “ship” in Fig. 3d) which are represented in the testing set, but not in the training set. When the negative training data cannot sufficiently represent the topics of negative data in the testing set, it is more desirable to build up the classifier by surrounding the positive data, whose topic is relatively explicit and easy to be depicted, into an optimal sphere, rather than only maximizing the margin between the positive and negative data, as the margin-based MIL methods do.

Table 5 shows the F-measure, precision and recall values on the new dataset. It is observed that SDB-MIL obtains improved classification performance over the margin-based MIL methods, i.e., mi-SVM and MI-SVM. In MIL, the positive data in the training set represents the same topic with that in the testing set, but the negative data in the training set may not sufficiently represent the negative topics in the testing set. In this case, only maximizing the margin between the positive and negative training data, as mi-SVM and MI-SVM do, is not enough to guarantee a good test performance. In contrast, our method not only maximizes the margin between the two classes in the training data, but it also constructs a tight sphere around the positive data, so that the chances of incorrectly classifying negative data as positive can be largely reduced.

Moreover, we present the confusion matrix on the real-world MIL dataset in Tables 6 and 7, where the first row lists the percentage of positive bags classified to each class, and the second row shows the percentage of negative bags assigned to each class. Taking Table 6a as an example, 81.14 percent positive bags (with $Y = 1$) are correctly classified to the positive class (class 1), and 18.86 percent positive bags (with $Y = 1$) are misclassified to the negative class

TABLE 5
Classification Results on the Real-World MIL Dataset

	Precision	Recall	F-measure	p-value
EM-DD	0.706	0.811	0.755*	<0.01
DD-SVM	0.816	0.797	0.806*	0.016
mi-SVM	0.819	0.864	0.840*	0.032
MI-SVM	0.817	0.837	0.827*	0.025
SVDD-NEG	0.759	0.782	0.771*	0.012
MIOptimalBall	0.683	0.769	0.723*	<0.01
SDB-MIL	**0.839**	**0.913**	**0.874**	N/A

(class -1). It is found that our proposed SDB-MIL method attains the highest classification accuracy on both of the positive and negative bags. This once again confirms that when the negative training data cannot appropriately represent the topics of negative data in the testing set, enclosing the positive data in an optimal sphere and meanwhile maximizing the margin between the two classes can help to improve the classification performance.

4.3 Performance Variation with CCCP Iterations

We will investigate the performance variation of SDB-MIL when the number of CCCP iterations changes. Considering that there are as many as 33 sub-datasets in the experiments, we pick up nine sub-datasets to report the results, i.e., Muta-atoms, Muta-bonds, Muta-chains sub-datasets from the Mutagenesis dataset, Corel 0, 1 and 2 sub-datasets from the Corel dataset, and Reuters 1, 2 and 3 sub-datasets from the Reuters dataset. The F-measure values after each CCCP iteration are summarized and recorded.

Figs. 4a, 4b, 4c present the F-measure value of SDB-MIL on the selected Mutagenesis, Corel and Reuters sub-datasets, respectively, when the number of CCCP iterations increases from 1 to 6. The X axis denotes the number of CCCP iterations, and the Y axis stands for the F-measure value. On the one hand, we can observe that the F-measure value goes up when the number of CCCP iterations increases. This is because the CCCP algorithm solves the learning problem (9) by iteratively replacing the nonconvex counterparts with their first-order Taylor expansions. As the CCCP iteration increases, the solution is closer to an optimal one and the classification performance improves correspondingly. On the other hand, it is seen that the F-measure value remains relatively stable after only a few number of iterations, which implies the convergence of our proposed method. SDB-MIL can converge to an optimal solution after only a small number of iterations using the CCCP technique. The similar findings can also be observed from the related CCCP studies [27], [28]. Furthermore, by applying the termination criterion in Table 1, SDB-MIL stops after 3.4 iterations on the Mutagenesis sub-datasets, 4.8 iterations on the Corel sub-datasets, and 4.6 iterations on the Reuters sub-datasets averagely.

TABLE 6
The Confusion Matrix on the Real-World Dataset (Numbers in Bold Indicate the Classification Accuracy for Each Class)

(a) EM-DD

	Class 1	Class -1
Y= 1	**81.14%**	18.86%
Y= -1	33.86%	**66.14%**

(b) DD-SVM

	Class 1	Class -1
Y= 1	**79.70%**	20.30%
Y= -1	17.98%	**82.02%**

(c) mi-SVM

	Class 1	Class -1
Y= 1	**86.36%**	13.64%
Y= -1	19.14%	**80.86%**

(d) MI-SVM

	Class 1	Class -1
Y= 1	**83.70%**	16.30%
Y= -1	18.76%	**81.24%**

(e) SVDD-NEG

	Class 1	Class -1
Y= 1	**78.24%**	21.76%
Y= -1	24.78%	**75.22%**

(f) MIOptimalBall

	Class 1	Class -1
Y= 1	**76.92%**	23.08%
Y= -1	35.76%	**64.24%**

TABLE 7
Confusion Matrix on the Real-World Dataset (cons.)

(a) SDB-MIL

	Class 1	Class -1
Y= 1	**91.30%**	8.70%
Y= -1	17.58%	**82.42%**

5 Conclusions and Future Work

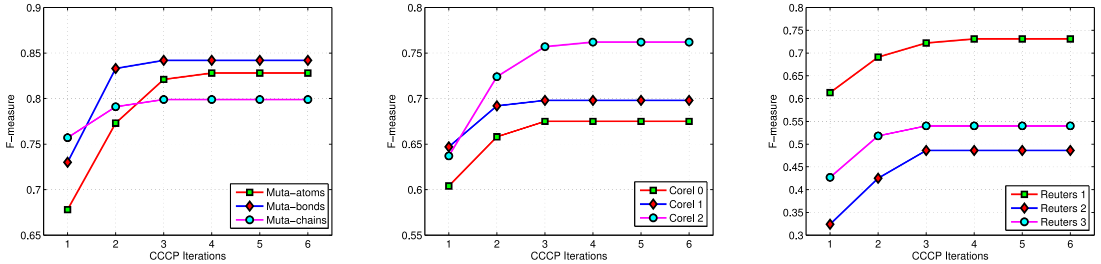
5.1 Contributions of This Work

In this paper, we address the multi-instance learning problem where the negative data in the testing set may not sufficiently represent the distribution of negative data in the testing set. To deal with this problem, we propose a novel Sphere-Description-Based approach for Multiple-Instance Learning, which constructs an optimal sphere by determining a large margin among the instances and requiring that each positive bag has at least one instance in the sphere and all negative bags are outside the sphere. Compared to the traditional margin-based MIL classifiers [5], [8], [19], [20], which learn the classifier by only maximizing the margin between two classes, SDB-MIL builds up the classifier by enclosing the positive data, whose topic is relative explicit and easy to be described, into an optimal sphere, so that the chances of incorrectly classifying negative data as positive can be largely reduced. Substantial experiments on MIL datasets have demonstrated that SDB-MIL obtains statistically better classification performance than the MIL methods compared.

5.2 Limitations and Future Work

The learning problem (8) of SDB-MIL is solved via CCCP. By employing the CCCP technique, the learning problem is decomposed into a series of QP problems (9). Assuming that the computational cost of solving a QP problem with n instances is $O(n^2)$, the time complexity of SDB-MIL is roughly $O(k*(n+m_1+2)^2)$. Here, $n=n_1+n_2$ is the number of instances in the training set. m_1 is the number of positive training bags. k is the CCCP iteration number, which is a relatively small value, as shown in Section 4.3.

SDB-MIL is much more efficient than EM-DD and DD-SVM. For example, the training time of SDB-MIL on the Muta-chains sub-dataset is 9.54 seconds, which is markedly lower than EM-DD (107.32 seconds) and DD-SVM (126 minutes). Moreover, the training efficiency of SDB-MIL is generally lower than mi-SVM and MI-SVM. The running time of SDB-MIL, mi-SVM and MI-SVM on the Muta-chains sub-dataset is 9.54, 9.27 and 2.55 seconds, respectively. Though SDB-MIL has a relatively higher cost, it is able to obtain significantly better classification performance than mi-SVM and MI-SVM, which can be observed in Section 4.



(a) Mutagenesis sub-datasets (b) Corel sub-datasets (c) Reuters sub-datasets

Fig. 4. Performance variation with different numbers of CCCP iterations.

In the future, we would like to exploit optimization techniques to speed up SDB-MIL. It is seen that SDB-MIL is comprised with a few numbers of QP problems and it can be accelerated by speeding up the resolving of QP problems. For example, Pegasos [40] or NORMA [41] can be adapted to solve the QP problem (9). By applying these methods, the running time of solving a QP problem can increase linearly with the number of non-zero features in each instance, and does not depend directly on the dataset size. These optimization techniques can speed up the resolving of QP problems, and thus reduce the time complexity of SDB-MIL. The application of these efficient QP optimization methods will be a valuable consideration in our work.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China (61472089, 61472090), in part by the NSFC-Guangdong Joint Found (U1501254), in part by the Natural Science Foundation of Guangdong Province (2014A030308008, 2015A030313486), in part by the Guangdong Province Natural Science Funds for Distinguished Young Scholar (S2013050014133), and in part by the Key Technology Research and Development Programs of Guangdong Province (2015B010131015, 2015B010108006). Bo Liu is the corresponding author.

References

[1] T. Dietterich, R. Lathrop, and T. Lozano-Pérez, “Solving the multiple instance problem with axis-parallel rectangles,” *Artif. Intell.*, vol. 89, no. 1, pp. 31–71, 1997.

[2] Q. Tao, S. D. Scott, N. V. Vinodchandran, T. T. Osugi, and B. Mueller, “Kernels for generalized multi-instance learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 12, pp. 2084–2098, Dec. 2008.

[3] A. Zafra and S. Ventura, “G3P-MI: A genetic programming algorithm for multiple instance learning,” *Inform. Sci.*, vol. 180, no. 23, pp. 4496–4513, 2010.

[4] J. T. Kwok and P.-M. Cheung, “Marginalized multi-instance kernels,” in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, pp. 901–906.

[5] S. Andrews, I. Tsochantaridis, and T. Hofmann, “Support vector machines for multi-instance learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 561–568.

[6] P. Auer and R. Ortner, “A boosting approach to multiple instance learning,” in *Proc. 15th Eur. Conf. Mach. Learn.*, 2004, pp. 63–74.

[7] R. C. Bunescu and R. J. Mooney, “Multiple instance learning for sparse positive bags,” in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 105–112.

[8] Y. Chen and J. Wang, “Image categorization by learning and reasoning with regions,” *J. Mach. Learn. Res.*, vol. 5, pp. 913–939, 2004.

[9] W. Li and D. Yeung, “Mild: Multi-instance learning via disambiguation,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 1, pp. 76–89, Jan. 2010.

[10] A. J. Smola, S. V. N. Vishwanathan, and T. Hofmann, “Kernel methods for missing variables,” in *Proc. Int. Workshop Artif. Intell. Statist.*, 2005, pp. 325–332.

[11] G. CMoore, J. Zaretzki, C. M. Breneman, and K. P. Bennett, “Fast bundle algorithm for multiple instance learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1068–1079, Jun. 2012.

[12] H. Zhang, S. R. Cholleti, R. Rahmani, S. A. Goldman, and J. E. Fritts, “Localized content-based image retrieval,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1902–1912, Nov. 2008.

[13] F. Kamangar, F. Nie, H. Wang, H. Huang, and C. H. Q. Ding, “Maximum margin multi-instance learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1–9.

[14] P. M. Cheung and J. T. Kwok, “A regularization framework for multiple-instance learning,” in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 193–200.

[15] O. Maron and A. L. Ratan, “Multiple-instance learning for natural scene classification,” in *Proc. Int. Conf. Mach. Learn.*, 1998, pp. 341–349.

[16] P. Viola, J. C. Platt, and C. Zhang, “Multiple instance boosting for object detection,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1417–1424.

[17] O. Maron and T. Lozano-Pérez, “A framework for multi-instance learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 1998, pp. 570–576.

[18] Q. Zhang and S. Goldman, “EM-DD: An improved multi-instance learning technique,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 1073–1080.

[19] P. V. Gehler and O. Chapelle, “Deterministic annealing for multiple-instance learning,” in *Proc. 11th Int. Conf. Artif. Intell. Statist.*, 2007, pp. 123–130.

[20] T. Gärtner, P. Flach, A. Kowalczyk, and A. Smola, “Multi-instance kernels,” in *Proc. Int. Conf. Mach. Learn.*, 2002, pp. 179–186.

[21] Z. Zhou and J. Xu, “On the relation between multi-instance learning and semi-supervised learning,” in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 1167–1174.

[22] Z. Y. Fu, A. Robles-Kelly, and J. Zhou, “MILIS: Multiple instance learning with instance selection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 958–977, May 2011.

[23] Y. Chen, J. Bi, and J. Wang, “MILE: Multiple instance learning via embedded instance selection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1931–1947, Dec. 2006.

[24] L. Dong, *A Comparison of Multi-Instance Learning Algorithms*. Univ. of Waikato, Hamilton, New Zealand, 2006.

[25] D. M. J. Tax and R. P. W. Duin, “Support vector data description,” *Mach. Learn.*, vol. 54, pp. 45–66, 2004.

[26] O. Chapelle, V. Sindhwani, and S. S. Keerthi, “Optimization techniques for semi-supervised support vector machines,” *J. Mach. Learn. Res.*, vol. 9, pp. 203–233, 2008.

[27] F. Wang, B. Zhao, and C.-S. Zhang, “Linear time maximum margin clustering,” *IEEE Trans. Neural Netw.*, vol. 21, no. 2, pp. 319–332, Feb. 2010.

[28] D. Zhang, F. Wang, S. Luo, and L. Tao, “Maximum margin multiple instance clustering with applications to image and text clustering,” *IEEE Trans. Neural Netw.*, vol. 22, no. 5, pp. 739–751, May 2011.

[29] X.-M. Wang, K. F.-L. Chung, and S.-T. Wang, “Theoretical analysis for solution of support vector data description,” *Neural Netw.*, vol. 24, pp. 360–369, 2011.

[30] H. Wang, F.-P. Nie, and H. Huang, “Learning instance specific distance for multi-instance classification,” in *Proc. Appl. Artif. Intell. Conf.*, 2011, pp. 507–512.

[31] M.-L. Zhang and Z.-H. Zhou, “M3MIML: A maximum margin method for multi-instance multi-label learning,” in *Proc. 8th IEEE Int. Conf. Data Mining*, 2008, pp. 688–697.

[32] K. Zhang, I. W. Tsang, and J. T. Kwok, "Maximum margin clustering made practical," *IEEE Trans. Neural Netw.*, vol. 20, no. 4, pp. 583–596, Apr. 2009.

[33] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proc. Int. Conf. Mach. Learn.*, 1997, pp. 179–186.

[34] G. Wu and E. Y. Chang, "Class-boundary alignment for imbalanced dataset learning," in *Proc. Workshop Learn. Imbalanced Datasets*, 2003, pp. 49–56.

[35] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li, "Multi-instance learning by treating instances as non-I.I.D. samples," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 1249–1256.

[36] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Reading, MA, USA: Addison-Wesley, 1989.

[37] H. Wang, F. Nie, and H. Huang, "Learning instance specific distance for multi-instance classification," in *Proc. Appl. Artif. Intell. Conf.*, 2011.

[38] J. He, D. Zhang, and R. D. Lawrence, "MI2LS: Multi-instance learning from multiple information sources," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 149–157.

[39] S. Belongie, J. M. Hellerstein, C. Carson, M. Thomas, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval," in *Proc. 3rd Int. Conf. Vis. Inf. Inf. Syst.*, 1999, pp. 509–516.

[40] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for SVM," in *Proc. 24th Int. Conf. Mach. Learn.*, Jun. 2007, pp. 807–814.

[41] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 1–12, Aug. 2004.

Yanshan Xiao received the PhD degree in computer science from the University of Technology, Sydney, in 2011. She is with the Faculty of Computer, Guangdong University of Technology. Her research interests include multiple-instance learning and support vector machine.

Bo Liu is with the Faculty of Automation, Guangdong University of Technology. His research interests include machine learning and data mining. He has published papers on *IEEE Transactions on Neural Networks*, *IEEE Transactions on Knowledge and Data Engineering*, etc.

Zhifeng Hao is a professor with the Faculty of Computer, Guangdong University of Technology. His research interests include design and analysis of algorithms, mathematical modeling, and combinatorial optimization.