A Similarity-Based Classification Framework For Multiple-Instance Learning

Yanshan Xiao, Bo Liu, Zhifeng Hao, *Senior Member, IEEE,* and Longbing Cao

***Abstract*—Multiple-instance learning (MIL) is a generalization of supervised learning that attempts to learn useful information from bags of instances. In MIL, the true labels of instances in positive bags are not available for training. This leads to a critical challenge, namely, handling the instances of which the labels are ambiguous (*ambiguous instances*). To deal with these ambiguous instances, we propose a novel MIL approach, called similarity-based multiple-instance learning (SMILE). Instead of eliminating a number of ambiguous instances in positive bags from training the classifier, as done in some previous MIL works, SMILE explicitly deals with the ambiguous instances by considering their similarity to the positive class and the negative class. Specifically, a subset of instances is selected from positive bags as the positive candidates and the remaining ambiguous instances are associated with two similarity weights, representing the similarity to the positive class and the negative class, respectively. The ambiguous instances, together with their similarity weights, are thereafter incorporated into the learning phase to build an extended SVM-based predictive classifier. A heuristic framework is employed to update the positive candidates and the similarity weights for refining the classification boundary. Experiments on real-world datasets show that SMILE demonstrates highly competitive classification accuracy and shows less sensitivity to labeling noise than the existing MIL methods.**

***Index Terms*—Classification, multiple-instance learning.**

I. Introduction

MULTIPLE-instance learning (MIL) [1]–[3] is proposed to address the classification of bags, which has been successfully applied in a wide variety of real-world applications, ranging from drug activity prediction [4]–[6], image retrieval [7]–[9], and natural scene classification [10], [11] to text categorization [12], [13]. In MIL, the labels in the training set are associated with sets of instances, which are called bags. A bag is labeled positive if at least one of its instances is positive; the bag is labeled negative if all of its instances are negative. The task of MIL is to classify unknown bags by using the information from labeled bags. In contrast to standard supervised learning, the key challenge of MIL is that the label of any single instance in a positive bag can be unavailable [14]–[17], since a positive bag may contain negative instances in addition to one or more positive instances. The true labels for the instances in a positive bag may or may not be the same as the corresponding bag label, which results in inherent ambiguity of instance labels in positive bags. We call the instances with ambiguous labels "ambiguous instances."

To handle the MIL ambiguity problem, different supervised methods have been proposed over the years. Since labels of instances in positive bags are not available, a straightforward approach is to transform the MIL into a standard supervised learning problem by labeling all instances in positive bags as positive [18]. However, these MIL methods are based on the assumption that the positive bags consist of fairly rich positive instances. Moreover, mislabeling the negative instances in positive bags as positive may limit the discriminative power of the MIL classifier. To account for this drawback, another group of MIL methods [12], [19]–[22] focuses on selecting a subset of instances from positive bags to learn the classifier. The remaining, unselected instances in positive bags are excluded from the learning phase. For example, RW-SVM [20] designs an instance selection mechanism to select one instance from each positive bag. Together with the negative instances from negative bags, these selected instances are used to build the classifier. EM-DD [21] chooses one instance that is most consistent with the current hypothesis in each positive bag to predict an unknown bag. However, the discriminative ability of these approaches may be restricted. This is because only a subset of instances is used to learn the classifier, while a significant number of remaining instances which may contribute an improvement to the MIL accuracy, is not sufficiently utilized in learning the classifier.

In this paper, we propose a novel multi-instance learning method, termed as similarity-based multiple-instance learning (SMILE). Instead of excluding a number of ambiguous instances from training the classifier, SMILE explicitly deals with the ambiguous instances by considering their similarity to both the positive class and the negative class. Specifically, we select one instance from each positive bag as the initial positive candidate. Based on the positive candidates,

Manuscript received July 30, 2012; revised December 22, 2012 and March 24, 2013; accepted April 01, 2013. Date of publication May 16, 2013; date of current version March 13, 2014. This work is supported in part by the Natural Science Foundation of China under Grant 61070033, Grant 61203280, and Grant 61202270, the Natural Science Foundation of Guangdong Province under Grant 9251009001000005, Grant S2011040004187, and Grant S2012040007078, the Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20124420120004, Australian Research Council Discovery Grant DP1096218, Grant DP130102691, and ARC Linkage Grant LP100200774 and Grant LP120100566. This paper was recommended by Associate Editor E. Santos. *(Corresponding author: B. Liu.)*

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).

L. Cao is with the UTS Advanced Analytics Institute, University of Technology, Sydney, NSW 2007, Australia (e-mail: longbing.cao@uts.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2013.2257749

each instance is associated with two similarity weights that represent the similarity to the positive class and the negative class, respectively. These ambiguous instances, together with their similarity weights, are incorporated into an extended formulation of support vector machine (SVM). Based on a heuristic learning framework (see Section IV-D), the selection of positive candidates and similarity weights can be updated to refine the classification boundary.

The main contributions of our work are as follows.

1) We propose a framework that converts a MIL problem into a supervised learning problem by considering the similarity of ambiguous instances to the classes. The incorporation of ambiguous instances enables a more powerful classifier with better discriminative ability.
2) We put forward a novel scheme to measure the similarity of ambiguous instances to the classes. By being associated with the similarity weights, ambiguous instances can be effectively incorporated in training the classifier.
3) We present an extended formulation of SVM to construct the MIL classifier. Compared to SVM, the extended SVM can incorporate the ambiguous instances, as well as their similarity weights, into the optimization process.
4) We evaluate our approach on real-world datasets. In the experiments, our approach demonstrates highly competitive classification accuracy and shows less sensitivity to the labeling noise than the existing MIL methods.

The rest of this paper is organized as follows. Section II reviews the related work. Section III presents a similarity-based data model and gives an overview of the proposed approach. Section IV gives the details of our approach for binary class classification. Section V extends our approach to multiclass classification by implementing some minor modifications. Experiments are conducted in Section VI. Section VII concludes the paper and outlines the future work.

II. Related Work

The proposed SMILE method is an SVM-based MIL approach. In this section, we will review the previous works on MIL and then introduce the basic idea of standard SVM.

A. Multi-Instance Learning

The initial MIL algorithms are presented in [1], [23], and [24], which are based on hypothesis classes consisting of axis-aligned rectangles. Then, many MIL methods from different perspectives have been proposed [25]–[29].

The first category of works sets the instance labels in positive bags as positive, and a standard supervised learning method or an iterative framework is adopted to train the classifier. For example, Ray and Craven [18] label all the instances in positive bags as positive and standard SVM is used to train the classifier straightforwardly. However, this method relies on the positive bags being fairly rich in positive instances. mi-SVM [12] initializes all instances in positive bags as positive and trains an SVM classifier iteratively until each positive bag has at least one instance classified to be positive. Clearly, mi-SVM focuses on obtaining 100% training accuracy of positive bags. However, if labeling noise exists in positive bags, the accuracy of mi-SVM may be greatly reduced. In MLBoost [30], all instances initially get the same label as the bag label and then a boosting framework is adopted to learn the classifier. However, MLBoost is based on the framework of boosting and may sometimes be less robust [31].

The second category of works [32], [33] designs mechanisms to map a bag of instances into a "bag-level" training vector. Typical examples include DD-SVM [32] and MILES [33]. DD-SVM learns a number of instance prototypes and utilizes them to map every bag to a point. In MILES, the bags are embedded in a new feature space and 1-norm SVM is applied to select the important features (instances) for prediction. However, DD-SVM and MILES may transform the MIL problem into a high dimensionality problem. The dimension of "bag-level" vectors is dependent on the number of training instances. If the instance number is large, the "bag-level" vector may turn out to be extremely high dimensional. To solve this problem, MILIS [34] proposes to select one instance from each positive bag to produce the instance prototypes so that the dimension of "bag-level" vectors can be largely reduced. However, the MIL classification accuracy may be biased since a large number of unselected instances in positive bags cannot be sufficiently utilized in learning the prototypes.

The third category of works [19]–[22], [33] focuses on selecting a subset of instances from positive bags to learn the classifier. For example, EM-DD [21] chooses one instance that is most consistent with the current hypothesis in each positive bag to predict an unknown bag. MI-SVM [12] adopts an iterative framework to learn an SVM classifier. At each iteration, one instance from each positive bag is selected. Together with the instances in negative bags, the selected instances are used to learn the classifier. RW-SVM [20] selects one instance from each positive bag to train the classifier. However, the discriminative ability of these approaches may be restricted. This is because only a subset of instances in positive bags is used to build the classifier, while a large number of ambiguous instances, which may contribute to the prediction, cannot be properly exploited in learning the classifier.

Other methods are also proposed to improve MIL classification accuracy [35]–[38]. For example, MissSVM [35] considers the MIL problem as a semisupervised learning problem, and Citation-kNN [39] extends the K-nearest neighbor method to solve the MIL problem.

In this paper, we propose a similarity-based multiple-instance learning method. Compared to the works in the third category, we explicitly utilize the ambiguous instances, which are not sufficiently considered in those works, to learn the MIL classifier. The incorporation of ambiguous instances makes our classifier more discriminative in classifying the positive and negative bags and meanwhile leads to better robustness.

B. Support Vector Machine (SVM)

Let $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$ be a training set, where $y_i \in \{+1, -1\}$ and n is the number of instances in the training set. To make the instances more linearly separable, a nonlinear mapping function $\phi(\cdot)$ is used to map the data from the input space into a feature space $\mathcal{F}$. The goal of SVM [40] is to find an optimized plane $\mathbf{w}^T\phi(\mathbf{x}) + b = 0$.

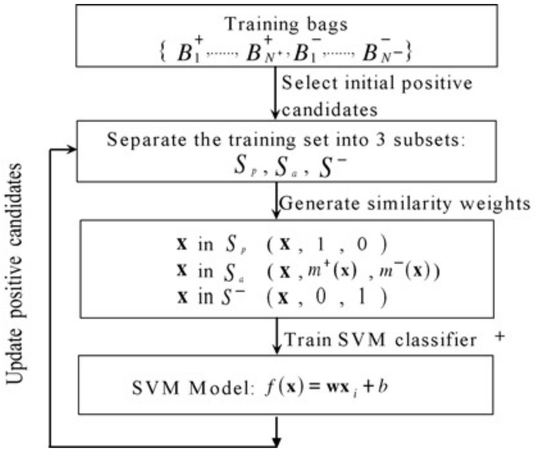
Training bags
$\{ B_1^+ \cdots\cdots B_{N^+}^+, B_1^- \cdots\cdots B_{N^-}^- \}$
Select initial positive candidates
Separate the training set into 3 subsets:
S_p, S_a, S^-
Generate similarity weights
$\mathbf{x}$ in S_p $(\mathbf{x}, 1, 0)$
$\mathbf{x}$ in S_a $(\mathbf{x}, m^+(\mathbf{x}), m^-(\mathbf{x}))$
$\mathbf{x}$ in S^- $(\mathbf{x}, 0, 1)$
Train SVM classifier +
SVM Model: $f(\mathbf{x}) = \mathbf{w}\mathbf{x}_i + b$
Update positive candidates

Fig. 1. Overview of our proposed approach.

To obtain the optimized plane, we need to solve the following objective function:

$$\min \; F(\mathbf{w}, b, \xi) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + c\sum_{i=1}^{n}\xi_i,$$
$$y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n \tag{1}$$

where ξ_i are error terms and c is a regularized parameter. We can obtain $\mathbf{w}$ and b by resolving problem (1). A test instance $\phi(\mathbf{x})$ is classified to the positive class if $\mathbf{w}^T\phi(\mathbf{x})+b \geq 0$ holds. Otherwise, it is predicted as negative.

In standard SVM (1), an instance is considered in only one class. However, in this paper, the ambiguous instance is considered in different classes and thus associated with more than one similarity weight. Standard SVM cannot handle multiple weights for one instance. To solve this problem, we present an extended formulation of SVM, which can effectively incorporate multiple similarity weights of ambiguous instances into the optimization procedure.

III. Similarity-Based Data Model and Algorithm Overview

A. Similarity-Based Data Model

We first introduce notations to describe the MIL problem. Let $D = \{(B_1^+, Y_1^+), \ldots, (B_{N^+}^+, Y_{N^+}^+), (B_1^-, Y_1^-), \ldots, (B_{N^-}^-, Y_{N^-}^-)\}$ denotes a set of training bags, where B_i^+ represents a positive bag with a positive label $Y_i^+ = +1$; B_i^- denotes a negative bag with a negative label $Y_i^- = -1$. N^+ and N^- are the numbers of positive bags and negative bags, respectively. In the following, we will omit the +/− sign when there is no need for disambiguation.

Each bag contains a set of instances. The jth instance in B_i^+ and B_i^- is denoted as B_{ij}^+ and B_{ij}^-, respectively. B_{ij}^+ is associated with a label y_{ij}^+ and B_{ij}^- is with y_{ij}^-. Based on the description of MIL, each positive bag has at least one positive instance and all instances in negative bags are negative. Hence, each positive bag has at least one instance of which the label is $y_{ij}^+ = +1$, and it has $y_{ij}^- = -1$ for all instances in negative bags. The purpose of MIL is to learn a classifier on the training data and use the classifier to predict a new bag.

For the sake of convenience, we line up the instances in all bags together, and re-index the instances as $\{(\mathbf{x}_i, y_i)\}$. Hence, the training set is transformed into $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_l, y_l)\}$ $(i = 1, 2, \ldots, l)$, where l is the total number of instances in the training bags. We then convert each instance $\mathbf{x}_i$ $(i = 1, \ldots, l)$ to a similarity-based data model which is defined as follows:

$$\{\mathbf{x}, m^+(\mathbf{x}), m^-(\mathbf{x})\} \tag{2}$$

where $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$ represent the similarity of $\mathbf{x}$ to the positive class and the negative class, respectively, and it has $0 \leq m^+(\mathbf{x}) \leq 1$ and $0 \leq m^-(\mathbf{x}) \leq 1$.

Using the similarity-based data model presented in (2), we can convert a multi-instance learning problem into a single-instance learning problem, and the supervised learning methods can be extended to solve the MIL problems. More importantly, by introducing the similarity-based data model, the ambiguous instance, which is usually neglected in the training phase due to its ambiguity nature [12], can be modeled and thereafter included in the learning phase, so that the classifier can be refined to be more discriminative.

B. Overview of SMILE Approach

We provide an overview of our proposed approach according to the similarity-based data model. The proposed approach works in four steps, as illustrated in Fig. 1, where each block indicates an object to be operated on and each arrow denotes an operation performed on the object.

Let S^+ and S^- contain the instances from positive bags and negative bags, respectively. The first step is initial positive candidate selection. A subset of instances from positive bags is selected as initial positive candidates. The subset S^+ is thereafter separated into two subsets S_p and S_a, i.e., $S^+ = S_p \cup S_a$. S_p includes the selected positive candidates from positive bags; S_a contains the remaining, unselected instances from positive bags. The second step is similarity weight generation. The similarity weights are assigned to the training instances. For the instances in S^-, $m^-(\mathbf{x}) = 1$ and $m^+(\mathbf{x}) = 0$ are set. In terms of the instances in S_p, we let $m^+(\mathbf{x}) = 1$ and $m^-(\mathbf{x}) = 0$. For the instances in S_a, we assign two similarity weights, i.e., $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$, with $0 < m^+(\mathbf{x}) < 1$ and $0 < m^-(\mathbf{x}) < 1$. The third step is MIL classifier training. An extended formulation of SVM, termed as similarity-based support vector machine (SSVM), is put forward to learn a MIL classifier based on the presented data model. The last step is positive candidate updating. A heuristic framework is proposed to reselect the positive candidates and update the SSVM classifier until the termination criterion is met. In the testing phase, a new bag B is predicted as positive if at least one instance is classified to be positive by the SSVM classifier. Otherwise, it is predicted as negative.

The details of SMILE approach will be discussed in Section IV and its deployment into multiclass classification will be presented in Section V. To simplify the presentation, we let $S^{*-} = S_a \cup S^-$ in the following.

IV. SMILE Approach

A. Initial Positive Candidate Selection

Each positive bag contains at least one positive instance, and it is possible to initially select one instance from the positive bag. This selected instance (called the initial positive candidate) is more likely to be positive, compared to the other instances in the same bag. In this paper, we select the initial positive candidate by modeling the distributions of instances in positive bags and negative bags. By doing this, one instance is selected as the initial positive candidate from each positive bag and is then put into the subset S_p.

Specifically, the selection of initial positive candidates is based on the following definitions:

Definition 1: (Single Set-Based Similarity): Given an instance $\mathbf{x}$ and a subset S, the similarity of $\mathbf{x}$ to S is defined as

$$R(\mathbf{x}, S) = \frac{1}{|S|} \sum_{\mathbf{x}_i \in S} e^{-||\mathbf{x}-\mathbf{x}_i||^2}, \tag{3}$$

where $R(\mathbf{x}, S)$ represents the similarity of $\mathbf{x}$ to the subset S; $|S|$ denotes the subset size of S. It has $||\mathbf{x} - \mathbf{x}_i||^2 = \mathbf{x} \cdot \mathbf{x} - 2\mathbf{x} \cdot \mathbf{x}_i + \mathbf{x}_i \cdot \mathbf{x}_i$, where $\mathbf{x} \cdot \mathbf{x}_i$ represents the inner product of $\mathbf{x}$ and $\mathbf{x}_i$. When a nonlinear mapping function $\phi(\cdot)$ is used to map the instance $\mathbf{x}$ into the feature space, $||\mathbf{x} - \mathbf{x}_i||^2$ in (3) is replaced by $||\phi(\mathbf{x}) - \phi(\mathbf{x}_i)||^2$ and we have $||\phi(\mathbf{x}) - \phi(\mathbf{x}_i)||^2 = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}) - 2\phi(\mathbf{x}) \cdot \phi(\mathbf{x}_i) + \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_i)$.

In Definition 1, the single set-based similarity $R(\mathbf{x}, S)$ is defined based on the exponential decay function, i.e., $e^{-||\mathbf{x}-\mathbf{x}_i||^2}$. By using the exponential decay function, the value of $R(\mathbf{x}, S)$ falls into the range between 0 and 1. When $\mathbf{x}$ lies closer to S, the value of $R(\mathbf{x}, S)$ tends to be larger. When $\mathbf{x}$ is infinitely far away from the instances in S, it has $R(\mathbf{x}, S) \approx 0$. The farther $\mathbf{x}$ is from S, the lower the similarity of $\mathbf{x}$ to S is, which is consistent with our intuitive observations.

Definition 2: (Binary Set-Based Similarity): Given an instance $\mathbf{x}$, subsets S^+ and S^-, the similarity of $\mathbf{x}$ to S^+ is defined as

$$Q(\mathbf{x} \in S^+ | S^+ \cup S^-) = \frac{1}{2}[R(\mathbf{x}, S^+) + 1 - R(\mathbf{x}, S^-)] \tag{4}$$

where S^+ and S^- are subsets containing the training instances from positive bags and negative bags, respectively.

It can be seen that Definition 2 extends Definition 1 to binary class classification in which the positive class and the negative class are available. In Definition 1, we give a general measurement of similarity between an instance and a subset. Based on Definition 1, we propose a binary set-based similarity measurement in Definition 2, where the positive class and the negative class are both considered in measuring the similarity. In (4), $R(\mathbf{x}, S^+)$ is the similarity of $\mathbf{x}$ to S^+. When the value of $R(\mathbf{x}, S^+)$ is larger, it indicates that $\mathbf{x}$ is more similar to S^+. $R(\mathbf{x}, S^-)$ represents the similarity of $\mathbf{x}$ to S^-, and $1 - R(\mathbf{x}, S^-)$ can be considered as the dissimilarity of $\mathbf{x}$ to S^-. When the value of $1 - R(\mathbf{x}, S^-)$ is larger, $\mathbf{x}$ is less similar to S^-. Moreover, $Q(\mathbf{x} \in S^+ | S^+ \cup S^-)$ is the average of $R(\mathbf{x}, S^+)$ and $1 - R(\mathbf{x}, S^-)$. When the instance $\mathbf{x}$ lies closer to S^+ and farther from S^-, the value of $Q(\mathbf{x} \in S^+ | S^-)$ becomes larger. That is to say, $\mathbf{x}$ is more similar to a positive instance and is less likely to be negative. We can also extend Definition 1 to multiclass classification problems where more than two classes are available, as presented in Section V-A.

Definition 3: (Positive Candidate): For the positive bag B_i^+ ($i = 1, \ldots, N^+$), an instance $\mathbf{x}$ is selected as the initial positive candidate, if it satisfies

$$\max_{\mathbf{x} \in B_i^+} Q(\mathbf{x} \in S^+ | S^+ \cup S^-). \tag{5}$$

Similar to MILD [22], one instance is selected out from each positive bag as the positive candidate. Intuitively, for any instance in a positive bag, the instance that is most similar to instances in positive bags and has least similarity to those in negative bags, is more likely to be positive compared to the remaining instances in the same bag. Therefore, the instance with the maximum value of $Q(\mathbf{x} \in S^+ | S^+ \cup S^-)$ (5) is chosen to be the positive candidate. After the positive candidates have been determined, we can further divide S^+ into two subsets S_p and S_a. S_p contains the positive candidates, while S_a includes the remaining, unselected instances, whose labels are relatively ambiguous compared to the positive candidates.

B. Similarity Weight Generation

The main task of this step is to generate similarity weights for each ambiguous instance to the positive class and the negative class, respectively. Based on the similarity-based data model, the training set can be transformed into a pseudo dataset which consists of three parts: $\{\mathbf{x}, 1, 0\}$ for the positive candidates in S_p, $\{\mathbf{x}, 0, 1\}$ for the negative instances in S^-, and $\{\mathbf{x}, m^+(\mathbf{x}), m^-(\mathbf{x})\}$ for the ambiguous instances in S_a, where $0 < m^+(\mathbf{x}) < 1$ and $0 < m^-(\mathbf{x}) < 1$ hold. It can be seen that the similarity weights of instances in S_p to the positive class and the negative class are 1 and 0, respectively. Moreover, the corresponding similarity weights of instances in S^- to the positive class and the negative class are 0 and 1. However, for the instances in S_a, the similarity weights $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$ are unknown. In the following, we show how to generate the similarity weights $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$ for the ambiguous instances in S_a.

The basic idea for computing the similarity weight $m^+(\mathbf{x})$ is to capture the instance's similarity to the positive class and the dissimilarity to the negative class. Likewise, $m^-(\mathbf{x})$ is calculated by considering the instance's dissimilarity to the positive class and the similarity to the negative class. Specifically, for each instance $\mathbf{x}$ in S_a, the corresponding similarity weights to the positive class and the negative class are calculated as follows:

$$\begin{aligned} m^+(\mathbf{x}) &= Q(\mathbf{x} \in S_p | S_p \cup S^-) \\ &= \tfrac{1}{2}[R(\mathbf{x}, S_p) + 1 - R(\mathbf{x}, S^-)], \end{aligned} \tag{6}$$

$$\begin{aligned} m^-(\mathbf{x}) &= Q(\mathbf{x} \in S^- | S_p \cup S^-) \\ &= \tfrac{1}{2}[R(\mathbf{x}, S^-) + 1 - R(\mathbf{x}, S_p)]. \end{aligned} \tag{7}$$

It is seen that $m^+(\mathbf{x})$ equals the similarity of $\mathbf{x}$ to S_p when S_p and S^- are given. Similarly, $m^-(\mathbf{x})$ is equivalent to the similarity of $\mathbf{x}$ to S^-. The generation of $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$ is based on S_p and S^-, where the labels of instances are relatively less ambiguous. Additionally, it has $m^+(\mathbf{x}) + m^-(\mathbf{x}) = 1$ based on (6) and (7).

Similar to our approach, MLBoost assigns instances with label weights. However, the similarity weight in our approach is different from the label weight in MLBoost [30]. MLBoost is based on a boosting framework which usually pays attentions to the misclassified instances. Thus, in MLBoost, the misclassified instance may have a larger label weight (in magnitude) than the correctly classified instance [30]. In contrast, in our approach, the instance is associated with a large similarity weight if its belonging to one class is explicit. From this aspect, the similarity weight in our approach and the label weight in MLBoost may represent quite different information.

C. Similarity-Based Support Vector Machine (SSVM)

In this section, we introduce the SSVM to incorporate the similarity weights into the optimization process. In standard SVM, the training instance is explicitly associated with one class. In SSVM, different from standard SVM, each ambiguous instance is associated with two similarity weights representing the corresponding similarity to the positive class and the negative class. If an ambiguous instance is associated with a larger similarity weight to the positive class, it is more likely to be a positive instance than a negative instance, and vice versa. By considering the similarity to the positive and negative classes, the ambiguous data can be explicitly incorporated in the training process and thereafter the learnt classifier is expected to have better generalization ability.

1) Primal Formulation

In our proposed approach, the similarity of an ambiguous instance to the positive and negative classes is considered, and hence each ambiguous instance is associated with two similarity weights. To incorporate the ambiguous instances, as well as their similarity weights, in the optimization process, SSVM is proposed. The formulation of SSVM is given by

$$\begin{aligned}
\min\ \ & F(\mathbf{w}, b, \xi) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + c_1 \sum_{i:\mathbf{x}_i \in S_p} \xi_i + c_2 \sum_{j:\mathbf{x}_j \in S_a} m^+(\mathbf{x}_j)\xi_j \\
& \quad + c_3 \sum_{k:\mathbf{x}_k \in S_a} m^-(\mathbf{x}_k)\xi_k^* + c_4 \sum_{g:\mathbf{x}_g \in S^-} \xi_g \\
s.t. \quad & \mathbf{w}^T\mathbf{x}_i + b \geq 1 - \xi_i, \qquad \forall\ i : \mathbf{x}_i \in S_p \\
& \mathbf{w}^T\mathbf{x}_j + b \geq 1 - \xi_j, \qquad \forall\ j : \mathbf{x}_j \in S_a \\
& \mathbf{w}^T\mathbf{x}_k + b \leq -1 + \xi_k^*, \quad \forall\ k : \mathbf{x}_k \in S_a \\
& \mathbf{w}^T\mathbf{x}_g + b \leq -1 + \xi_g, \quad \forall\ g : \mathbf{x}_g \in S^- \\
& \xi_i \geq 0,\ \ \xi_j \geq 0,\ \ \xi_k^* \geq 0,\ \ \xi_g \geq 0; \qquad (8)
\end{aligned}$$

where ξ_i, ξ_j, ξ_k^*, and ξ_g are error terms; $m^+(\mathbf{x}_j)\xi_j$ and $m^-(\mathbf{x}_k)\xi_k^*$ can be considered as errors with different weights; c_1, c_2, c_3, and c_4 are regularization parameters which control the tradeoff between the plane margin and the errors.

In the objective function (8), it is clear that the similarity weights to the negative class for instances $\mathbf{x}_i$ in the positive candidate set S_p are 0, and that they are only associated with the positive class, their weighted errors in the objective function being $c_1 \sum_{i:\mathbf{x}_i \in S_p} \xi_i$. The similarity weights to the positive class for instances $\mathbf{x}_g$ in the negative instance set S^- are 0 and the weighted errors are $c_4 \sum_{g:\mathbf{x}_g \in S^-} \xi_g$, since S^- contains the instances from the negative bags and all of them are negative. Instances $\mathbf{x}_j$ and $\mathbf{x}_k$ in the ambiguous instance set S_a are considered to be in both the positive class and the negative class and thus have nonzero similarity weights to both classes. Their weighted errors are given as $c_2 \sum_{j:\mathbf{x}_j \in S_a} \xi_j + c_3 \sum_{k:\mathbf{x}_k \in S_a} \xi_k^*$, where $c_2 \sum_{j:\mathbf{x}_j \in S_a} \xi_j$ is the weighted error of ambiguous instances associated with the positive class and $c_3 \sum_{k:\mathbf{x}_k \in S_a} \xi_k^*$ is with the negative class. From the above analysis, it is easy to see that though the labels of ambiguous instances in S_a are unavailable, they can in fact be incorporated into the supervised learning process by considering their similarity to the positive class and the negative class.

2) Solution to SSVM

The optimization problem in (8) can be converted to the dual form by differentiating the Lagrangian function with the original variables $\mathbf{w}$, b, ξ_i, ξ_j, ξ_k^* and ξ_g. To do this, we introduce the Lagrange multipliers $\alpha_i \geq 0$, $\alpha_j \geq 0$, $\alpha_k^* \geq 0$, $\alpha_g \geq 0$, $\beta_i \geq 0$, $\beta_j \geq 0$, $\beta_k^* \geq 0$, and $\beta_g \geq 0$. Based on the defined Lagrange multipliers, the Lagrangian function of the objective function in (8) can be given as

$$\begin{aligned}
L = & \frac{1}{2}\mathbf{w}^T\mathbf{w} + c_1 \sum_{i:\mathbf{x}_i \in S_p} \xi_i + c_2 \sum_{j:\mathbf{x}_j \in S_a} m^+(\mathbf{x}_j)\xi_j \\
& + c_3 \sum_{k:\mathbf{x}_k \in S_a} m^-(\mathbf{x}_k)\xi_k^* + c_4 \sum_{g:\mathbf{x}_g \in S^-} \xi_g \\
& - \sum_{i:\mathbf{x}_i \in S_p} \alpha_i[\mathbf{w}^T\mathbf{x}_i + b - 1 + \xi_i] - \beta_i\xi_i \\
& - \sum_{j:\mathbf{x}_j \in S_a} \alpha_j[\mathbf{w}^T\mathbf{x}_j + b - 1 + \xi_j] - \beta_j\xi_j \\
& + \sum_{k:\mathbf{x}_k \in S_a} \alpha_k^*[\mathbf{w}^T\mathbf{x}_k + b + 1 - \xi_k^*] - \beta_k^*\xi_k^* \\
& + \sum_{g:\mathbf{x}_g \in S^-} \alpha_g[\mathbf{w}^T\mathbf{x}_g + b + 1 - \xi_g] - \beta_g\xi_g \qquad (9)
\end{aligned}$$

Differentiating the Lagrangian function (9) with $\mathbf{w}$, b, ξ_i, ξ_j, ξ_k^* and ξ_g, the following equations are obtained:

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{w}} &= -\sum_{i:\mathbf{x}_i \in S_p} \alpha_i\ \mathbf{x}_i - \sum_{j:\mathbf{x}_j \in S_a} \alpha_j\ \mathbf{x}_j + \mathbf{w} \\
& \quad + \sum_{k:\mathbf{x}_k \in S_a} \alpha_k^*\ \mathbf{x}_k + \sum_{g:\mathbf{x}_g \in S^-} \alpha_g\ \mathbf{x}_g = 0 \qquad (10) \\
\frac{\partial L}{\partial b} &= -\sum_{i:\mathbf{x}_i \in S_p} \alpha_i - \sum_{j:\mathbf{x}_j \in S_a} \alpha_j \\
& \quad + \sum_{g:\mathbf{x}_g \in S^-} \alpha_g + \sum_{k:\mathbf{x}_k \in S_a} \alpha_k^* = 0 \qquad (11) \\
\frac{\partial L}{\partial \xi_i} &= c_1 - \alpha_i - \beta_i = 0,\ \ i : \mathbf{x}_i \in S_p \qquad (12) \\
\frac{\partial L}{\partial \xi_j} &= c_2 m^+(\mathbf{x}_j) - \alpha_j - \beta_j = 0,\ \ j : \mathbf{x}_j \in S_a \qquad (13) \\
\frac{\partial L}{\partial \xi_k^*} &= c_3 m^-(\mathbf{x}_k) - \alpha_k^* - \beta_k^* = 0,\ \ k : \mathbf{x}_k \in S_a \qquad (14) \\
\frac{\partial L}{\partial \xi_g} &= c_4 - \alpha_g - \beta_g = 0,\ \ g : \mathbf{x}_g \in S^-. \qquad (15)
\end{aligned}$$

If we substitute (10)–(15) into the Lagrangian function (9) straightforwardly, the deviation could become relatively complicated. To simplify the deviation, we let $S^{*-} = S^{-} \cup S_a$ and make the following redefinitions:

$$\alpha_i^{+} = \begin{cases} \alpha_i, & i : \mathbf{x}_i \in S_p \\ \alpha_j, & j : \mathbf{x}_j \in S_a \end{cases} \tag{16}$$

$$\alpha_j^{-} = \begin{cases} \alpha_k^{*}, & k : \mathbf{x}_k \in S_a \\ \alpha_g, & g : \mathbf{x}_g \in S^{-} \end{cases} . \tag{17}$$

Hence, (10) and (11) can be rewritten as

$$\mathbf{w} = \sum_{i:\mathbf{x}_i \in S^{+}} \alpha_i^{+}\mathbf{x}_i - \sum_{j:\mathbf{x}_j \in S^{*-}} \alpha_j^{-}\mathbf{x}_j \tag{18}$$

$$\sum_{i:\mathbf{x}_i \in S^{+}} \alpha_i^{+} = \sum_{j:\mathbf{x}_j \in S^{*-}} \alpha_j^{-}. \tag{19}$$

Moreover, we let

$$c_i^{+} = \begin{cases} c_1, & i : \mathbf{x}_i \in S_p \\ c_2 m^{+}(\mathbf{x}_j), & j : \mathbf{x}_j \in S_a \end{cases} \tag{20}$$

$$c_j^{-} = \begin{cases} c_3 m^{-}(\mathbf{x}_k), & k : \mathbf{x}_k \in S_a \\ c_4, & g : \mathbf{x}_g \in S^{-} \end{cases} \tag{21}$$

$$\beta_i^{+} = \begin{cases} \beta_i, & i : \mathbf{x}_i \in S_p \\ \beta_j, & j : \mathbf{x}_j \in S_a \end{cases} \tag{22}$$

$$\beta_j^{-} = \begin{cases} \beta_k^{*}, & k : \mathbf{x}_k \in S_a \\ \beta_g, & g : \mathbf{x}_g \in S^{-}. \end{cases} \tag{23}$$

Then, (12) and (13) can be represented as (24). Equations (14) and (15) can be rewritten as (25).

$$c_i^{+} = \alpha_i^{+} + \beta_i^{+}, \tag{24}$$

$$c_j^{-} = \alpha_j^{-} + \beta_j^{-}. \tag{25}$$

From the Kuhn–Tucker Theorem, we substitute (18), (19), (24), and (25) into the Lagrangian function (9). The Wolfe dual of (8) can be obtained as

$$\begin{aligned} \min\ & F(\alpha_i^{+}, \alpha_j^{-}) = \\ & \frac{1}{2}\sum_{i:\mathbf{x}_i \in S^{+}}\sum_{j:\mathbf{x}_j \in S^{+}} \alpha_i^{+}\alpha_j^{+}\mathbf{x}_i \cdot \mathbf{x}_j - \sum_{i:\mathbf{x}_i \in S^{+}}\sum_{j:\mathbf{x}_j \in S^{*-}} \alpha_i^{+}\alpha_j^{-}\mathbf{x}_i \cdot \mathbf{x}_j \\ & + \frac{1}{2}\sum_{i:\mathbf{x}_i \in S^{*-}}\sum_{j:\mathbf{x}_j \in S^{*-}} \alpha_i^{-}\alpha_j^{-}\mathbf{x}_i \cdot \mathbf{x}_j - \sum_{i:\mathbf{x}_i \in S^{+}} \alpha_i^{+} - \sum_{j:\mathbf{x}_j \in S^{*-}} \alpha_j^{-}. \\ & s.t. \sum_{i:\mathbf{x}_i \in S^{+}} \alpha_i^{+} = \sum_{j:\mathbf{x}_j \in S^{*-}} \alpha_j^{-}, \\ & 0 \le \alpha_i^{+} \le c_i^{+},\ i : \mathbf{x}_i \in S^{+}, \\ & 0 \le \alpha_j^{-} \le c_j^{-},\ j : \mathbf{x}_j \in S^{*-}. \end{aligned} \tag{26}$$

3) Decision Boundary Determination

After solving the dual form (26), $\mathbf{w}$ and b are obtained. The decision function to predict the instance label is given by

$$y(\mathbf{x}) = \begin{cases} +1, & \mathbf{w}^T\mathbf{x} + b \ge 0, \\ -1, & \mathbf{w}^T\mathbf{x} + b < 0. \end{cases} \tag{27}$$

where $y(\mathbf{x})$ denotes the label of $\mathbf{x}$.

The objective of MIL is to train a classifier on the bag data and utilize the obtained classifier to predict the labels of bags. Based on the instance-level decision function (27), the decision function to predict the bag label is given as follows:

$$Y(B) = \begin{cases} -1, & \sum\limits_{\mathbf{x}_i \in B} y(\mathbf{x}_i) = -|B|, \\ +1, & \text{otherwise} \end{cases} \tag{28}$$

where B is a test bag; $Y(B)$ denotes the predicted label of B; $|B|$ is the number of instances in B. B is predicted as negative only if all instances in B are classified as negative, i.e. $\sum\limits_{\mathbf{x}_i \in B} y(\mathbf{x}_i) = -|B|$. Otherwise, B is classified as positive.

4) Nonlinear Kernel-Based SSVM

We can extend SSVM to nonlinear classification problems. In this case, a nonlinear mapping function $\phi(\cdot)$ is used to map all instances from the input space into a feature space, where both classes are expected to be more linearly separable. The inner products of two vectors in the feature space can be computed using a kernel function $K(\cdot, \cdot)$, as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j). \tag{29}$$

To extend SSVM in the feature space, we need to replace the inner product $\mathbf{x}_i \cdot \mathbf{x}_j$ with $K(\mathbf{x}_i, \mathbf{x}_j)$ in the single set-based similarity (3) and the dual form (26). At the same time, in the decision function (27), $\mathbf{w}^T\mathbf{x}$ is replaced by

$$\sum_{i:\mathbf{x}_i \in S^{+}} \alpha_i^{+} K(\mathbf{x}_i, \mathbf{x}) - \sum_{j:\mathbf{x}_j \in S^{*-}} \alpha_j^{-} K(\mathbf{x}_j, \mathbf{x}).$$

D. Positive Candidate Updating and Heuristic Strategy

To refine the decision boundary, a heuristic strategy that is based on alternating optimization method [41], [42], is proposed to update the positive candidates. The basic idea of the heuristic strategy is that we first initialize the positive candidates, and then repeatedly train the MIL classifier and update the positive candidates until the termination criterion is met. Specifically, the positive candidates are initialized as described in Section IV-A. Then, steps 1) and 2) repeat alternatively until the termination criterion in (30) is satisfied.

1) Fixing the obtained positive candidates, generate similarity weights according to (6) and (7), and then solve the optimization problem (26) to obtain the Lagrange multipliers $\alpha = \{\alpha_i^{+}, \alpha_j^{-}\}$.

2) Fixing the obtained Lagrange multipliers α, update the positive candidates as follows:

$$\begin{aligned} i_1^{(t+1)} &= \arg \min_{\mathbf{x}_g \in B_1^{+}} F\{\alpha^{(t)}, \mathbf{x}_g, \mathbf{x}_{i_2^{(t)}}, \mathbf{x}_{i_3^{(t)}}, \ldots, \mathbf{x}_{i_{N^+}^{(t)}}\}, \\ xi_2^{(t+1)} &= \arg \min_{\mathbf{x}_g \in B_2^{+}} F\{\alpha^{(t)}, \mathbf{x}_{i_1^{(t+1)}}, \mathbf{x}_g, \mathbf{x}_{i_3^{(t)}}, \ldots, \mathbf{x}_{i_{N^+}^{(t)}}\}, \\ & \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ i_k^{(t+1)} &= \arg \min_{\mathbf{x}_g \in B_k^{+}} F\{\alpha^{(t)}, \mathbf{x}_{i_1^{(t+1)}}, \ldots, \mathbf{x}_{i_{k-1}^{(t+1)}}, \\ & \qquad \mathbf{x}_g, \mathbf{x}_{i_{k+1}^{(t)}}, \ldots, \mathbf{x}_{i_{N^+}^{(t)}}\}, \\ & \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ i_{N^+}^{(t+1)} &= \arg \min_{\mathbf{x}_g \in B_{N^+}^{+}} F\{\alpha^{(t)}, \mathbf{x}_{i_1^{(t+1)}}, \mathbf{x}_{i_2^{(t+1)}}, \ldots, \mathbf{x}_{i_{N^+-1}^{(t+1)}}, \mathbf{x}_g\} \end{aligned}$$

where $i_k^{(t)}$ and $i_k^{(t+1)}$ are indexes of positive candidates in B_k^{+} at the tth and $(t+1)$th iterations, respectively. Hence, $\mathbf{x}_{i_k^{(t)}}$ and $\mathbf{x}_{i_k^{(t+1)}}$ are corresponding positive candidates of B_k^{+} at the tth and

$(t+1)$th iterations. $\alpha^{(t)} = \{\alpha_i^{+(t)}, \alpha_j^{-(t)}\}$ are Lagrange multipliers obtained from (26) at the tth iteration.

In the dual form (26), the value of F is determined by α, S^+, S^{*-}, $m^+(\mathbf{x})$, and $m^-(\mathbf{x})$. On one hand, S^+ and S^{*-} can be easily obtained after S_p is determined. We have $S^+ = S_p + S_a$ and $S^{*-} = S_a + S^-$, where S^- remains unchanged throughout the iterations, and $S_a = D - S_p - S^-$. On the other hand, $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$ are decided by S_p and S^- according to (6) and (7). Hence, after α and S_p are determined, the value of F can be obtained. For this reason, we only list α and positive candidates $\mathbf{x}_{i_k^{(t)}}$ in (30).

It is seen from (30) that for each positive bag, we choose one instance that leads to the minimum value of F (26), as the new positive candidate for the next iteration. Let us take B_k^+ as an example to illustrate how to update its positive candidate. First, we select one instance $\mathbf{x}_g$ from B_k^+ and $\mathbf{x}_g$ is assumed to be the positive candidate of B_k^+. Second, S_p is updated by replacing the positive candidate in the tth iteration, i.e., $\mathbf{x}_{i_k^{(t)}}$, with $\mathbf{x}_g$. Third, the value of F is computed by substituting $\alpha^{(t)}$ and the newly updated subset S_p. No quadratic programming (QP) problem is required to compute the F value, since $\alpha^{(t)}$ is given. We traverse all instances in B_k^+, and the instance leading to the minimum value of F is selected as the positive candidate in the $(t+1)$th iteration.

3) Repeat the above two steps until the following stopping criterion is met:

$$F^{(t)} - F^{(t+1)} \leq \epsilon F^{(t)} \tag{30}$$

where $F^{(t)}$ and $F^{(t+1)}$ are the values of F obtained by solving (26) in the tth and $(t+1)$th iterations, respectively; ϵ is a threshold. ϵ is set to be 0.01 in the experiments.

The value of F is determined by the Lagrange multipliers α and the positive candidates in subset S_p. We alternatively optimize the Lagrange multipliers (step 1)] and positive candidates (step 2)] in an EM-like fashion to minimize the value of F. Based on this, we have the following relations:

$$F(\alpha^{(t)}, S_p^{(t)}) \geq F(\alpha^{(t+1)}, S_p^{(t)}) \geq F(\alpha^{(t+1)}, S_p^{(t+1)}) \tag{31}$$

where $F(\alpha^{(t)}, S_p^{(t)}) \geq F(\alpha^{(t+1)}, S_p^{(t)})$ corresponds to step 1) Lagrange multipliers optimizing; $F(\alpha^{(t+1)}, S_p^{(t)}) \geq F(\alpha^{(t+1)}, S_p^{(t+1)})$ is associated with step 2) positive candidate updating. It is seen that the value of F is monotonically decreased during the whole process of optimization.

The SMILE approach is presented in Algorithm 1. Since the value of F is nonnegative and decreases monotonically, Algorithm 1 can converge after a finite number of steps.

V. Extension to Multiclass Classification

In this section, we extend our approach to multiclass classification, where more than two classes are available in the training set. Unlike the traditional classification methods, which decompose the multiclass classification problem into a series of binary class classification problems and then combine the results to yield the final labels, we present a uniform formulation of multiclass classification based on our similarity-based data model and learning framework.

Algorithm 1 SMILE for Multi-Instance Learning Problems.

```
D consists of the instances from all the training bags. $S^-$ contains the instances from the negative bags.
Initialize $\epsilon$; let $S_p \leftarrow \emptyset$, $S_p^{'} \leftarrow \emptyset$, $S_a \leftarrow \emptyset$;
Select one instance from each positive bag as the initial positive candidate, as described in Section IV-A;
Put the initial positive candidates in $S_p$;
Let $t = 0$; $F^{(0)}$ and $MinVal$ be large positive values;
repeat
  t=t+1;
  while t > 1 do
    for (each positive bag $B_k^+$) do
      for (each instance $\mathbf{x}_g$ in $B_k^+$) do
        Let $\mathbf{x}_g$ be the positive candidate of $B_k^+$;
        $S_p^{'} \leftarrow S_p$;
        Update $S_p^{'}$ by replacing $\mathbf{x}_{i_k^{(t-1)}}$ with $\mathbf{x}_g$;
        $S_a \leftarrow D - S_p^{'} - S^-$;
        Calculate the value of $F$ by substituting $\alpha^{(t-1)}$, $S_a$ and $S_p^{'}$ in (26);
        if $F < MinVal$ then
          $MinVal \leftarrow F,\ i_k^{(t)} \leftarrow g$;
        end if
      end for
      Update $S_p$ by replacing $\mathbf{x}_{i_k^{(t-1)}}$ with $\mathbf{x}_{i_k^{(t)}}$;
    end for
  end while
  $S_a \leftarrow D - S_p - S^-$;
  Compute $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$ according to (6) and (7);
  Obtain $\alpha$ and $F$ by solving QP in (26) based on $S^-$, $S_p$, $S_a$, $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$;
  $\alpha^{(t)} \leftarrow \alpha$, $F^{(t)} \leftarrow F$;
until $F^{(t-1)} - F^{(t)} \leq \epsilon F^{(t-1)}$
OUTPUT ($\mathbf{w}$, b)
```

According to [34], the MIL problem in multiclass classification is slightly different from that in binary class classification. In binary class classification, each positive bag contains at least one positive instance and all instances in the negative bags are negative. In contrast, in multiclass classification, each class is a positive one against the remaining classes. That is to say, for each bag in any class, there is at least one positive instance [34]. Based on this problem definition, our method can be adapted to multiclass classification by undertaking minor modifications, presented as follows.

A. Initial Positive Candidate Selection and Similarity Weight Generation

In multiclass classification, we assume that the training set consists of K classes, i.e., $C_1, \ldots, C_K$. Let $\mathcal{Y} = \{1, \ldots, K\}$ contain the indexes of classes in the training set. The goal of MIL multiclass classification is to classify a test bag into one of the K classes. To select the initial positive candidates in multiclass classification, we have the following definitions:

Definition 4 (Multiple Set-Based Similarity): Given an instance $\mathbf{x}$ and K subsets, i.e., $S^1, S^2, \ldots, S^K$, the similarity of $\mathbf{x}$ to S^i $(i = 1, \ldots, K)$ is defined as follows:

$$\begin{aligned} &Q(\mathbf{x} \in S^i | S^1 \cup \cdots \cup S^K) \\ &\quad = \frac{1}{K-1} \sum_{j \in \mathcal{Y}/i} Q(\mathbf{x} \in S^i | S^i \cup S^j) \\ &\quad = \frac{1}{K(K-1)} \sum_{j \in \mathcal{Y}/i} \left(R(\mathbf{x}, S^i) + 1 - R(\mathbf{x}, S^j)\right). \end{aligned} \tag{32}$$

where the subsets $S^1, S^2, \ldots, S^K$ contain the instances of class $C_1, C_2, \ldots, C_K$, respectively. $Q(\mathbf{x} \in S^i | S^1 \cup \cdots \cup S^K)$ denotes the similarity of $\mathbf{x}$ to S^i, when $S^1, \ldots, S^K$ are given. $R(\mathbf{x}, S^i)$ and $R(\mathbf{x}, S^j)$ can be computed according to the single-set based similarity (3). "/" means ruling out, so that $j \in \mathcal{Y}/i$ can also be rewritten as: $j \in \mathcal{Y}$ and $j \neq i$.

In Definition 4, the similarity in multiclass classification can be computed from that of binary class classification. To compute $Q(\mathbf{x} \in S^i | S^1 \cup \cdots \cup S^K)$, we first calculate the similarity of $\mathbf{x}$ to S^i, when S^i and S^j $(j \neq i)$ are given, i.e., $Q(\mathbf{x} \in S^i | S^i \cup S^j)$. Then, $Q(\mathbf{x} \in S^i | S^1 \cup \cdots \cup S^K)$ is obtained by averaging all the similarity. When $K = 2$, the multiple set-based similarity (32) can degrade to the case in binary class classification (4). Based on the similarity measurement given in Definition 4, one positive candidate can be initially selected from each bag, as shown in Remark 1.

Remark 1: For all instances in the subset S^i, one instance is selected as the initial positive candidate, if it satisfies

$$\max_{\mathbf{x} \in S^i} \; Q(\mathbf{x} \in S^i | S^1 \cup \cdots \cup S^K). \tag{33}$$

After the positive candidates are selected, we put the positive candidate of S^i into the subset S_p^i, and the other unselected instances are included in the subset S_a^i. Then, we let $S_p = S_p^1 \cup \cdots \cup S_p^K$, and $S_a = S_a^1 \cup \cdots \cup S_a^K$.

For each instance $\mathbf{x}$ in S_a, the similarity weight $m^i(\mathbf{x})$ of $\mathbf{x}$ to the subset S^i $(i = 1, \ldots, K)$ is computed as

$$\begin{aligned} m^i(\mathbf{x}) &= Q(\mathbf{x} \in S_p^i | S_p) \\ &= \frac{1}{K(K-1)} \sum_{j \in \mathcal{Y}/i} \left(R(\mathbf{x}, S_p^i) + 1 - R(\mathbf{x}, S_p^j). \right) \end{aligned} \tag{34}$$

It is seen that the similarity weights are generated by considering the similarity of $\mathbf{x}$ to the positive candidate subset S_p^i when $S_p^1, \ldots, S_p^K$ are given. It is worth noting that the values of similarity weights $m^i(\mathbf{x})$ fall into the range between 0 and 1. At the same time, for any instance $\mathbf{x}$, the sum of its similarity weights is equal to 1, i.e., $\sum_{i=1}^{K} m^i(\mathbf{x}) = 1$.

B. Similarity-Based Multiclass SVM (SMSVM)

In the following, we present an extension of similarity-based SVM, termed as similarity-based multiclass SVM (SMSVM), to incorporate the similarity weights in a uniform formulation for multiclass classification. To do this, we first line up the instances in S_p as $\mathbf{x}_i$ $(i = 1, \ldots, |S_p|)$ and S_a as $\mathbf{x}_j$ $(|S_p|+1, \ldots, |S_p|+|S_a|)$. $|S_p|$ and $|S_a|$ denote the corresponding subset sizes of S_p and S_a. Then, based on the one-against-all strategy, the formulation of SMSVM is given as

$$\begin{aligned} \min \quad & F(\mathbf{w}_r, b_r, \xi_i^{y_i,r}, \xi_j^{g,r}) = \frac{1}{2} \sum_{r \in \mathcal{Y}} ||\mathbf{w}_r||^2 \\ & + c_1 \sum_{\substack{r \in \mathcal{Y}/y_i \\ i:\mathbf{x}_i \in S_p}} \xi_i^{y_i,r} + c_2 \sum_{g \in \mathcal{Y}} \sum_{\substack{r \in \mathcal{Y}/g \\ j:\mathbf{x}_j \in S_a}} m^g(\phi(\mathbf{x}_j)) \xi_j^{g,r} \\ s.t. \quad & (\mathbf{w}_{y_i}^T \phi(\mathbf{x}_i) + b_{y_i}) - (\mathbf{w}_r^T \phi(\mathbf{x}_i) + b_r) \geq 2 - \xi_i^{y_i,r}, \\ & \qquad i : \mathbf{x}_i \in S_p, \quad r \in \mathcal{Y}/y_i \\ & (\mathbf{w}_g^T \phi(\mathbf{x}_j) + b_g) - (\mathbf{w}_r^T \phi(\mathbf{x}_j) + b_r) \geq 2 - \xi_j^{g,r}, \\ & \qquad j : \mathbf{x}_j \in S_a, \quad g \in \mathcal{Y}, \quad r \in \mathcal{Y}/g \\ & \xi_i^{y_i,r} \geq 0, \quad \xi_j^{g,r} \geq 0 \end{aligned} \tag{35}$$

where $\mathbf{w}_r$ and b_r are the corresponding norm vector and bias of the rth classifier; y_i indicates the instance label of $\mathbf{x}_i$ and is the same as the bag label; $\xi_i^{y_i,r}$ and $\xi_j^{g,r}$ are error terms.

Similar to the traditional multiclass SVM, SMSVM constructs K classifiers, i.e., $f(\mathbf{x}) = \mathbf{w}_r^T \phi(\mathbf{x}) + b_r$ $(r = 1, \ldots, K)$, one for each class. To obtain the rth classifier, a hyperplane is trained between class r and the other $K-1$ classes.

As discussed in Section V-A, the positive candidate $\mathbf{x}_i$ in S_p is more likely to be positive compared to the other instances in the same bag. We let $\mathbf{x}_i$'s label y_i be the same as its bag label. Hence, the similarity of $\mathbf{x}_i$ to class y_i is equal to 1 and that to the other classes is 0, the weighted errors of positive candidates in the objective function being

$$c_1 \sum_{\substack{r \in \mathcal{Y}/y_i \\ i:\mathbf{x}_i \in S_p}} \xi_i^{y_i,r}.$$

Additionally, for the ambiguous instance $\mathbf{x}_j$ in S_a, the instance label is relatively ambiguous and its similarity to all the classes is considered. That is to say, it has $0 < m^g(\phi(\mathbf{x}_j)) < 1$ $(g \in \mathcal{Y})$. Therefore, the weighted errors of ambiguous instances in S_a are

$$c_2 \sum_{g \in \mathcal{Y}} \sum_{\substack{r \in \mathcal{Y}/g \\ j:\mathbf{x}_j \in S_a}} m^g(\phi(\mathbf{x}_j)) \xi_j^{g,r}.$$

Here,

$$\sum_{\substack{r \in \mathcal{Y}/g \\ j:\mathbf{x}_j \in S_a}} m^g(\phi(\mathbf{x}_j)) \xi_j^{g,r}$$

are the weighted errors when the similarity of $\mathbf{x}_j$ to class g is considered.

To solve the objective function (35), we first let $\widetilde{S^i} = S_p^i \cup S_a$ $(i = 1, \ldots, K)$ and $\widetilde{S} = \widetilde{S^1} \cup \ldots \widetilde{S^K}$. The dual form of the objective function (35) can be then given by

$$\begin{aligned} \min \quad & F(\alpha_i^r) = \sum_{i,j:\mathbf{x}_i,\mathbf{x}_j \in \widetilde{S}} \left(\frac{1}{2} u_j^{y_i} A_i A_j - \sum_{r \in \mathcal{Y}} \alpha_i^r \alpha_j^{y_i} \right. \\ & \left. + \frac{1}{2} \sum_{r \in \mathcal{Y}} \alpha_i^r \alpha_j^r \right) K(\mathbf{x}_i, \mathbf{x}_j) - 2 \sum_{\substack{r \in \mathcal{Y} \\ i:\mathbf{x}_i \in \widetilde{S}}} \alpha_i^r \\ s.t. \quad & \sum_{i:\mathbf{x}_i \in \widetilde{S}} \alpha_i^r = \sum_{i:\mathbf{x}_i \in \widetilde{S}} u_i^r A_i, \quad r \in \mathcal{Y} \\ & 0 \leq \alpha_i^r \leq c_i^g, \; \alpha_i^{y_i} = 0, \; r \in \mathcal{Y}/y_i, \; i : \mathbf{x}_i \in \widetilde{S} \\ & A_i = \sum_{r \in \mathcal{Y}} \alpha_i^r, \quad i : \mathbf{x}_i \in \widetilde{S} \\ & u_j^{y_i} = \begin{cases} 1, & y_i = y_j \\ 0, & y_i \neq y_j. \end{cases} \\ & c_i^g = \begin{cases} c_1, & i : \mathbf{x}_i \in S_p \\ c_2 m^g(\phi(\mathbf{x}_i)), & i : \mathbf{x}_i \notin S_p \end{cases} \end{aligned} \tag{36}$$

After solving the dual form in (37), the norm vectors $\mathbf{w}_r$ can be obtained as

$$\mathbf{w}_r = \sum_{i:\mathbf{x}_i \in \widetilde{S}} (u_i^r A_i - \alpha_i^r) \phi(\mathbf{x}_i), \quad r \in \mathcal{Y}. \tag{37}$$

For a test bag B, the decision function is given by (38). It is seen that B is assigned the label corresponding to the largest decision value output by the SMSVM

$$\arg \max_{\mathbf{x}_j\in B} \max_{r\in\mathcal{Y}} \left(\sum_{i:\mathbf{x}_i\in\widetilde{S}} (u_i^r A_i - \alpha_i^r) K(\mathbf{x}_i, \mathbf{x}_j) + b_r. \right). \quad (38)$$

C. Positive Candidate Updating

The positive candidate updating and heuristic strategy in multiclass classification is similar to that described in Section IV-D. By using formulas (32) and (33), the initial positive candidates can be selected out from the bags, and thereafter the subsets $S_p^1, \ldots, S_p^K, S_a^1, \ldots, S_a^K$, as well as S_p and S_a can be obtained. Then, we start the following steps.

1) Fixing the positive candidates in $S_p^1, \ldots, S_p^K$, obtain the Lagrange multipliers α_i^r by resolving the optimization function (37). As $S_p^1, \ldots, S_p^K$ are known, S_p and $\widetilde{S}$ can be easily available. Furthermore, the similarity weights $m^g(\mathbf{x})$ can be computed by (34). By substituting S_p, $\widetilde{S}$, and $m^g(\mathbf{x})$, the dual form (37) can be solved and the Lagrange multipliers α_i can be obtained.

2) Fixing the Lagrange multipliers α_i^r, update the positive candidates in $S_p^1, \ldots, S_p^K$. In a similar manner to the case in binary class classification, we substitute the Lagrange multipliers α_i^r into the dual form (37), and change the positive candidate in each bag to minimize the value of $F(\alpha_i^r)$, similar to that in (30). Then, for each bag, the instance which corresponds to the smallest value of $F(\alpha_i^r)$ is selected as the positive candidate in the next iteration.

The above steps repeat until the criterion (30) is met.

D. Speeding Up Similarity-Based Multiclass SVM

The instance $\mathbf{x}_i$ in S_a is associated with each of the classes by assigning the similarity weights, so that its similarity to the classes can be involved in the objective function to boost the MIL classifier. However, every coin has two sides. Since the similarity of ambiguous instances to all classes is considered, the time complexity of learning the classifier could increase rapidly when the class number K is large. Furthermore, when an instance is far from one class, its similarity to this class may be too small to be considered, compared to the other classes. In this case, the similarity consideration may contribute little to the classifier improvement but may raise the time complexity dramatically. Therefore, we propose a strategy to increase the training efficiency of our approach by selecting out the distinctive similarity weights to train the classifier.

Remark 2: A similarity weight $m^j(\mathbf{x})$ is set to be 0 if it is smaller than the average value of similarity weights associated with class C_j, as follows:

$$m^j(\mathbf{x}) = \begin{cases} m^j(\mathbf{x}), & m^j(\mathbf{x}) \geq \frac{1}{|S_a|}\sum_{i:\mathbf{x}_i\in S_a} m^j(\mathbf{x}_i). \\ 0, & \text{otherwise} \end{cases} \quad (39)$$

In Remark 2, $\frac{1}{|S_a|}\sum_{i:\mathbf{x}_i\in S_a} m^j(\mathbf{x}_i)$ indicates the average value of similarity weights associated with class C_j. For an instance $\mathbf{x}$, if its similarity weight $m^j(\mathbf{x})$ to class C_j is smaller than the average value, it is set to be 0 and thereafter not involved in the optimization function. By doing this, the classifier can be built by using only the distinctive similarity weights. It can greatly improve training efficiency while remaining comparable classification accuracy.

VI. Experiments

We perform experiments on real-world datasets to evaluate the effectiveness of SMILE. All experiments are run on a laptop with 2.8 GHz processor and 3 GB DRAM. The SVM-based algorithms are implemented based on LibSVM [44]. The objectives of our experiments are as follows:

1) to evaluate the effectiveness of SMILE for handling the ambiguity of instance labels in MIL;
2) to evaluate the sensitivity of SMILE to the labeling noise with respect to classification accuracy;
3) to evaluate the performance of SMILE in solving multiclass multi-instance learning problems.

A. Baselines and Experimental Setting

We compare SMILE with the following four baselines.

1) The first is EM-DD [21], which focuses on selecting a subset of instances to predict an unknown bag.
2) The second is mi-SVM [12], which trains the classifier iteratively until each positive bag has at least one instance classified as positive. It is seen that mi-SVM aims to obtain 100% training accuracy of positive bags.
3) The third is MI-SVM [12], which uses a particular instance to replace the whole bag for training the classifier.
4) The fourth is DD-SVM [32], which maps a bag of instances into a "bag-level" vector and uses these vectors to train a bag-level classifier, such that all points in positive bags are able to contribute to the prediction.

We follow the same parameter selecting routine in [2], [12], [22], [45], and [46] to set the parameters. For the SVM-based methods, i.e., mi-SVM, MI-SVM, DD-SVM, and SMILE, RBF kernel is used. The kernel parameter in the RBF kernel is selected from 2^{-5} to 2^5. For the regularization parameters c_1, c_2, c_3 and c_4 in binary class classification (8), we let $c_1 = c_2$ and $c_3 = c_4$, with each of them being selected from 2^{-5} to 2^5. For c_1 and c_2 in multiclass classification (35), we find them from 2^{-5} to 2^5. As in [2], [12], and [45], the averaged results of 10-fold cross-validation with five independent runs are summarized.

B. Performance for Binary Class Classification

We evaluate the performance of SMILE for binary class classification based on several publicly available benchmarks: the Musk dataset[1] for drug activity prediction, the Corel dataset[2] for image retrieval, and the 20 Newsgroup dataset[3] for text categorization. These datasets are originally from the UCI Machine Learning Repository. Due to their problem characteristics, it is appropriate to consider them from a multi-instance perspective. Hence, the researchers converted them

[1] Available at http://www.cs.columbia.edu/Andrews/mil/datasets.html

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

[3] Available at http://people.csail.mit.edu/jrennie/20Newsgroups/

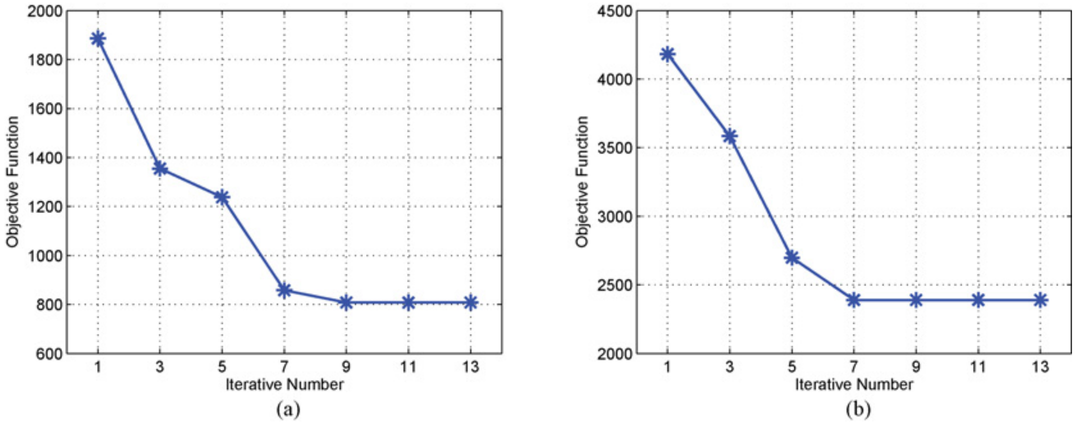

Fig. 2. Converge curve of objective function. (a) Musk1 dataset. (b) Musk2 dataset.

TABLE I
ACCURACY ON MUSK1 AND MUSK2 DATASETS

	Musk1 p-value	**Musk2** p-value
APR [1]	**92.4 ± 0.7** 0.387	89.2 ± 1.5 0.034
EM-DD [21]	84.8 ± 2.2 0.024	84.9 ± 1.7 0.015
mi-SVM [12]	87.4 ± 1.8 0.037	83.6 ± 2.6 0.005
MI-SVM [12]	77.9 ± 2.4 <0.001	84.3 ± 2.2 0.013
DD-SVM [32]	85.8 ± 1.7 0.027	91.3 ± 1.3 0.947
SMILE_R	89.5 ± 1.4 0.043	89.2 ± 1.6 0.038
SMILE	91.3 ± 1.3 N/A	**91.6 ± 1.1** N/A

into MIL datasets and used them to verify the performance of MIL algorithms. At present, they are popularly used in the MIL studies [12], [17], [22], [32], [33], [35], [46].

1) *Musk Dataset:* The Musk dataset consists of the data of molecules. A molecule is considered as a bag and each molecular shape is regarded as an instance. The Musk dataset contains two subsets: Musk1 and Musk2. The Musk1 dataset has 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 dataset has 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 1044. A total of 72 molecules is shared between Musk1 and Musk2. Each instance is represented by a 166-dimensional feature vector.

We first compare SMILE with the baselines—APR [1], EM-DD [21], mi-SVM [12], MI-SVM [12] and DD-SVM [32]. Table I reports the average accuracies, standard deviations, and p-values on the Musk1 and Musk2 datasets. The best accuracy is in bold. The p-values are computed by performing the paired t-test comparing all other classifiers to SMILE under the null hypothesis that there is no difference between the testing accuracy distributions. When the p-value is lower than the confidence level 0.05, there is a significant difference between SMILE and the method compared. On the Musk1 dataset, the average classification accuracy of SMILE is 91.3%, which is statistically better than most baselines, except for APR. As pointed out by [12] and [22], APR has been designed particularly for the drug activity prediction problem. On the Musk2 dataset, SMILE obtains the best average testing accuracy at 91.6% which is statistically better than APR, EM-DD, mi-SVM and MI-SVM as the p-values are lower than 0.05. Though the difference between SMILE and DD-SVM on the Musk2 dataset is not significant, SMILE attains markedly better accuracy than DD-SVM at 5.5% on the Musk1 dataset.

Next, we examine the effectiveness of our strategy on selecting the initial positive candidates. Rather than using the criterion in (5), we randomly select one instance in each positive bag as the initial positive candidate, called SMILE_R. Except for the different schemes on selecting the initial positive candidates, SMILE_R is the same as SMILE. Table I presents the results of SMILE_R. We can find that SMILE is statistically better than SMILE_R, as the p-values for SMILE_R are lower than 0.05 for both Musk1 and Musk2 datasets. The superior performance of SMILE over SMILE_R confirms the effectiveness of our initial positive candidate selection scheme. Since the positive bag may contain negative instances, SMILE_R randomly picks up one instance from the positive bag and a negative instance may be selected as the positive candidate, which could mislead the classifier and decrease the accuracy. Compared to SMILE_R, our scheme is more accurate in identifying the positive instances and hence leads to better classification accuracy.

Last, we investigate the convergence of our proposed approach. Fig. 2(a) and (b) present the converge curves of objective function for the Musk1 and Musk2 subdatasets, respectively. The x-axis denotes the iterative number and the y-axis represents the objective function's value (8). On one hand, we can observe that the objective function's value decreases dramatically as the iterative number goes up. This is because the optimal SSVM classifier is solved by minimizing the values of the objective function (8). When the iterative number increases, the classification accuracy obtains substantial improvements and the objective function's value reduces correspondingly. On the other hand, after a few iterations, the objective function's value remains relatively stable, which implies the convergence of our proposed approach. By alternatively updating the positive candidates and training the SSVM

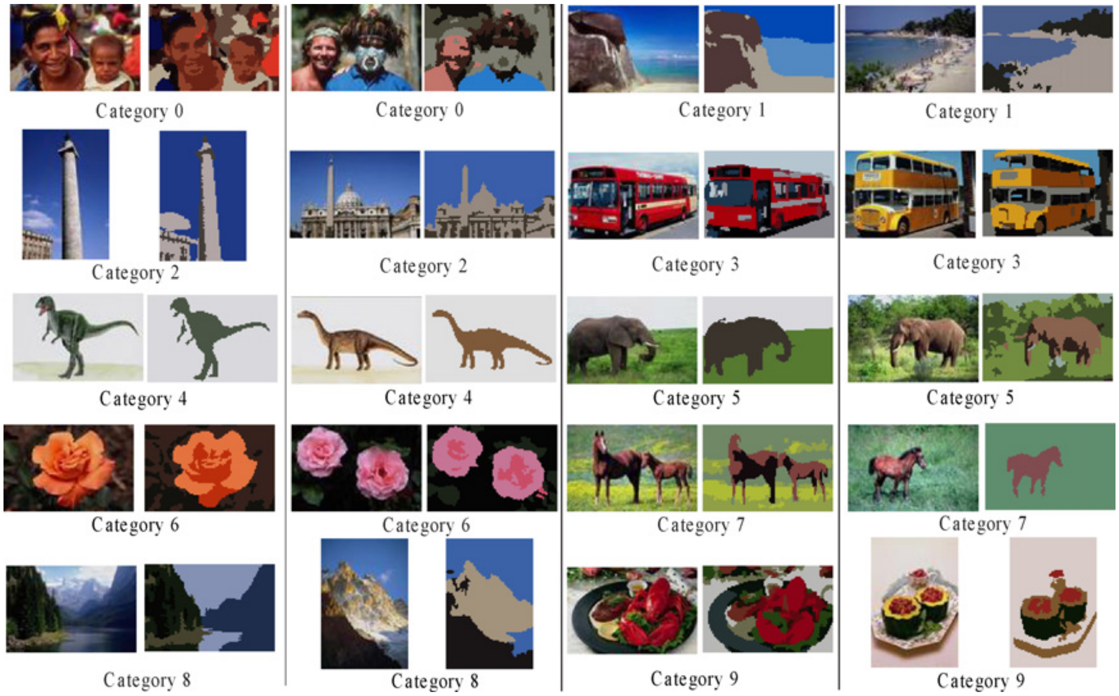

Fig. 3. Images randomly sampled from 10 categories and the corresponding segmentation results. Segmented regions are shown in their representative colors.

TABLE II
IMAGE CATEGORIES AND THE AVERAGE NUMBER OF INSTANCES PER BAG (INST/BAG) FOR EACH CATEGORY

ID	Category name	Inst/bag
0	*African people and villages*	4.84
1	*Beach*	3.54
2	*Historical building*	3.10
3	*Buses*	7.59
4	*Dinosaurs*	2.00
5	*Elephant*	3.02
6	*Flowers*	4.46
7	*Horses*	3.89
8	*Mountains and glaciers*	3.38
9	*Food*	7.24

classifier, our algorithm is able to converge to an optimal solution after only a small number of iterations. By using the termination criterion in (30), our algorithm stops after seven iterations on the Musk1 dataset and six iterations on the Musk2 dataset on average. We have similar findings for the other experimental datasets.

2) *Corel Dataset:* The Corel 1000 dataset is used in the experiments. It contains 10 CD-ROMs published by Corel Corporation. Each Corel CD-ROM of 100 images represents one distinct topic of interest, and the dataset thereafter has 10 thematically diverse image categories, each containing 100 images. Since the Corel dataset is not designed for multi-instance learning, we follow the operations in [12] to generate the MIL dataset for binary class classification. On one hand, we choose one category as the positive class and select 100 images uniformly from the remaining categories as the negative class at each round. As a result, 10 subdatasets are generated and each subdataset consists of 100 positive images and 100 negative images. On the other hand, we segment each image in the 10 subdatasets into several regions, as described in [32]. In the MIL setting, an image corresponds to a bag and a region corresponds to an instance. Table II shows the average number of instances per bag for different categories and Fig. 3 presents images randomly sampled from 10 categories and the corresponding segmentation results.

The results on the Corel subdatasets are reported in Table III, where the Category ID indicates the category which is considered as the positive class. After examining the detailed results, it is impressive to find that SMILE delivers the highest classification accuracy on 9 out of 10 subdatasets. In particular, SMILE is very competitive with EM-DD and MI-SVM, in which one instance is selected to replace the whole bag for constructing the classifier. There is a significant difference in classification accuracy between SMILE and EM-DD on almost all cases except for the “Category4” subdataset. Meanwhile, the classification accuracy of SMILE is statistically better than MI-SVM on 8 out of 10 subdatasets as seen from the p-values. The better performance of SMILE implies that when the number of positive instances is unknown, rather than using one less ambiguous instance to replace the bag, as done in EM-DD and MI-SVM, considerably incorporating the ambiguous instances in learning the classifier can greatly boost the multi-instance learning accuracy.

3) *20 Newsgroup Dataset:* The 20 Newsgroup dataset is popularly used for text categorization and multi-instance learning [46]–[48]. It contains 20 different subcategories. Each subcategory contains about 1000 news items. As in [46], we generate 50 positive bags and 50 negative bags for each of the 20 news categories. Each positive bag contains at least one post randomly drawn from the target category and the other

TABLE III
ACCURACY ON THE COREL DATASET

Category ID	EM-DD	mi-SVM	MI-SVM	DD-SVM	SMILE
	p-value	*p*-value	*p*-value	*p*-value	
Category0	68.7 ± 1.9	71.1 ± 1.8	69.6 ± 2.2	70.9 ± 1.1	**72.4** ± **0.8**
	0.001	0.036	0.024	0.046	
Category1	56.7 ± 3.7	58.7 ± 2.8	56.4 ± 2.5	58.5 ± 3.1	**62.7** ± **2.3**
	<0.001	0.001	<0.001	0.013	
Category2	65.1 ± 2.8	67.9 ± 1.3	66.9 ± 2.2	68.6 ± 1.6	**69.6** ± **1.1**
	0.019	0.046	0.027	0.135	
Category3	85.1 ± 3.2	88.6 ± 1.4	84.9 ± 1.8	85.2 ± 2.7	**90.1** ± **1.3**
	0.007	0.048	0.005	0.004	
Category4	96.2 ± 1.9	94.8 ± 1.5	95.3 ± 2.4	**96.9** ± **1.2**	96.6 ± 1.2
	0.819	0.032	0.178	0.982	
Category5	74.2 ± 3.3	80.4 ± 2.9	74.4 ± 3.7	78.2 ± 2.2	**80.5** ± **1.8**
	<0.001	0.953	0.011	0.038	
Category6	77.9 ± 2.9	82.5 ± 2.4	82.7 ± 2.6	77.9 ± 3.3	**83.3** ± **2.1**
	0.019	0.716	0.736	<0.001	
Category7	91.4 ± 1.9	93.4 ± 1.6	92.1 ± 1.4	94.4 ± 2.5	**94.7** ± **1.2**
	0.037	0.045	0.014	0.853	
Category8	70.9 ± 1.8	72.5 ± 1.4	67.2 ± 2.7	71.8 ± 1.9	**73.8** ± **1.1**
	0.013	0.047	<0.001	0.025	
Category9	80.2 ± 2.1	84.6 ± 1.8	83.4 ± 1.2	84.7 ± 1.4	**84.9** ± **1.5**
	0.018	0.842	0.032	0.924	

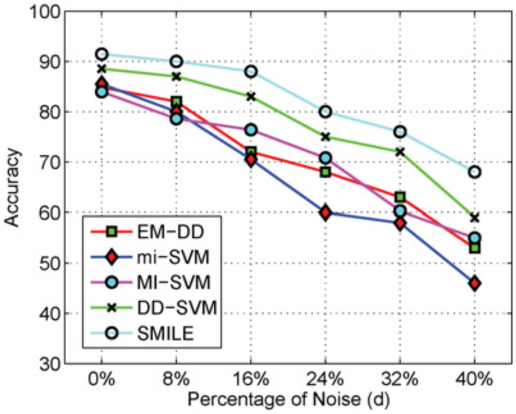

Fig. 4. Sensitivity to labeling noise (Musk dataset).

instances (and all instances in negative bags) are randomly and uniformly drawn from other categories. Each instance is a post represented by the top 200 TFIDF features. Finally, 20 subdatasets are obtained and used in the experiments.

The results on the 20 Newsgroup subdatasets are presented in Table IV, where "Category Name" is the category regarded as the positive class. It can be seen that SMILE shows improved performance over DD-SVM and mi-SVM on most of the subdatasets. Comparing SMILE with mi-SVM, there is a significant difference in classification accuracy on 17 out of 20 subdatasets. Moreover, the classification accuracy of SMILE is statistically better than DD-SVM for all cases except for the "Talk.religion.misc" subdataset. Instead of mapping the instances to "bag-level" data points (DD-SVM) or requiring each bag with one instance classified as positive (mi-SVM), SMILE provides a novel solution to incorporate the ambiguous instances in the learning phase by considering their similarity to the classes, which is shown to be effective in handing the ambiguous instances.

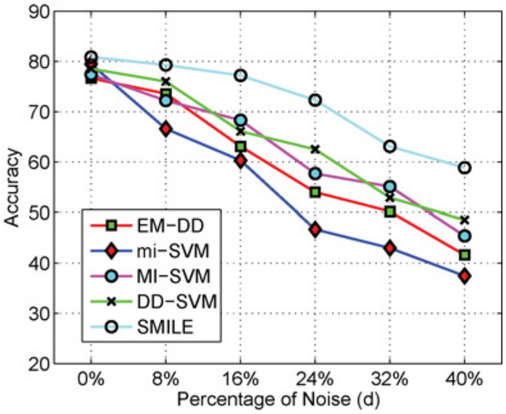

Fig. 5. Sensitivity to labeling noise (Corel dataset).

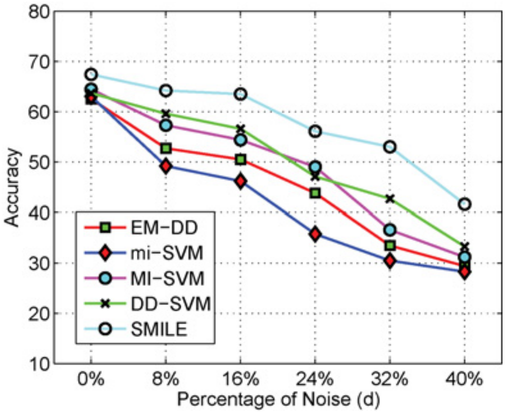

Fig. 6. Sensitivity to labeling noise (20 Newsgroup dataset).

C. Sensitivity to Labeling Noise

We investigate the noise sensitivity of SMILE and the baselines on the benchmarks. We follow the same routine in [22] to generate the labeling noise in the datasets. First, we randomly pick up $\frac{1}{2}d\%$ of positive bags and $\frac{1}{2}d\%$ of negative bags from the training set. Second, we change the labels of the selected positive and negative bags, i.e., relabel the positive bags as negative and the negative bags as positive.

TABLE IV
ACCURACY ON THE 20 NEWSGROUP DATASET

Category Name	EM-DD p-value	mi-SVM p-value	MI-SVM p-value	DD-SVM p-value	SMILE
Alt.atheism	63.6 ± 1.9 0.047	64.2 ± 1.6 0.036	59.6 ± 2.1 0.002	62.7 ± 2.6 0.037	**66.3 ± 1.4**
Comp.graphics	76.2 ± 1.8 0.028	76.7 ± 1.5 0.035	73.9 ± 2.4 0.016	75.9 ± 2.1 0.032	**78.5 ± 1.1**
Comp.os.ms-windows.misc	59.5 ± 1.6 0.043	57.3 ± 2.7 0.022	58.3 ± 2.5 0.037	59.2 ± 2.3 0.045	**61.7 ± 1.9**
Comp.sys.ibm.pc.hardware	58.1 ± 2.9 0.014	54.4 ± 3.4 <0.001	59.1 ± 2.1 0.035	60.2 ± 1.7 0.039	**62.5 ± 1.5**
Comp.sys.mac.harware	55.6 ± 3.5 0.002	57.6 ± 2.3 0.039	55.2 ± 3.3 0.012	57.6 ± 2.6 0.022	**60.8 ± 2.3**
Comp.window.x	67.2 ± 2.3 0.013	65.2 ± 1.8 <0.001	64.9 ± 2.3 <0.001	70.7 ± 1.2 0.048	**72.6 ± 0.9**
Misc.forsale	53.9 ± 1.8 0.031	55.6 ± 1.2 0.048	51.1 ± 2.9 0.008	54.5 ± 1.5 0.034	**56.7 ± 1.1**
Rec.autos	67.4 ± 2.3 0.018	68.5 ± 1.6 0.032	66.8 ± 3.3 0.031	68.6 ± 1.9 0.046	**70.8 ± 1.6**
Rec.motorcycles	55.8 ± 3.8 <0.001	59.9 ± 4.8 0.019	60.7 ± 3.2 0.019	57.8 ± 3.4 0.005	**65.4 ± 2.4**
Rec.sport.baseball	61.9 ± 2.2 0.006	66.1 ± 3.2 0.726	64.5 ± 2.1 0.048	61.6 ± 2.3 0.013	**66.9 ± 1.8**
Rec.sport.hockey	80.5 ± 4.8 <0.001	85.3 ± 4.2 0.025	83.4 ± 3.5 0.023	80.1 ± 3.8 <0.001	**88.9 ± 3.1**
Sci.crypt	63.7 ± 3.8 0.003	68.1 ± 2.4 0.043	**71.2 ± 3.2** 0.973	64.4 ± 3.3 0.002	70.8 ± 2.2
Sci.electronics	78.1 ± 3.5 0.023	79.3 ± 2.6 0.023	78.2 ± 3.3 0.034	80.2 ± 2.4 0.037	**82.6 ± 1.7**
Sci.med	63.5 ± 3.2 0.011	65.4 ± 3.4 0.015	63.6 ± 2.9 0.029	67.3 ± 2.5 0.025	**70.3 ± 2.3**
Sci.space	74.2 ± 3.1 0.024	76.3 ± 2.7 0.028	75.4 ± 3.4 0.006	76.5 ± 2.8 0.037	**80.1 ± 2.1**
Sci.religion.christian	47.9 ± 1.8 0.018	**50.7 ± 2.8** 0.935	48.7 ± 1.4 0.046	46.7 ± 2.7 0.014	50.4 ± 1.2
Talk.politics.guns	48.2 ± 3.5 0.011	50.4 ± 2.7 0.038	49.4 ± 3.2 0.029	48.6 ± 3.9 0.002	**53.9 ± 2.5**
Talk.politics.mideast	63.4 ± 3.7 0.005	67.3 ± 2.3 0.047	66.6 ± 2.5 0.031	65.1 ± 3.4 0.038	**69.8 ± 2.2**
Talk.politics.misc	57.7 ± 2.8 0.015	60.7 ± 1.5 0.024	59.8 ± 1.7 0.037	58.2 ± 2.6 0.009	**62.6 ± 1.4**
Talk.religion.misc	54.8 ± 1.4 0.033	**56.8 ± 2.5** 0.984	54.2 ± 1.6 0.045	57.8 ± 2.3 0.456	56.4 ± 1.2

Finally, all the selected bags are returned to the training set. By doing so, the training set has $d\%$ of bags with noisy labels.

Figs. 4–6 present the corresponding average classification accuracy on the Musk, Corel, and 20 Newsgroup datasets when the percentage of labeling noise $d\%$ varies from 0% to 40%. It is seen that SMILE is more robust than the baselines. When the percentage of labeling noise increases from 0% to 40%, SMILE has the lowest decrease in accuracy on all the datasets. In contrast to SMILE, mi-SVM seems to be the most sensitive method to labeling noise. For example, in Fig. 5, the classification accuracy of mi-SVM declines rapidly with the increase of labeling noise. This may be because mi-SVM focuses on obtaining 100% training accuracy of positive bags. If labeling noise of positive bags exists, the learnt classifier may be severely biased by the noise. Moreover, EM-DD and DD-SVM appear to be more sensitive to the noise than SMILE. In SMILE, the ambiguous instances are included in the learning phase by considering their similarity to the positive and negative classes, which leads to better robustness in classifying the MIL data.

D. Performance for Multiclass Classification

We turn our attentions to the performance of SMILE for solving multiclass classification problems. The baselines, i.e., EM-DD, mi-SVM, MI-SVM, DD-SVM, are originally proposed for binary class classification and we extend them for multiclass classification by performing one-against-all decomposition [49], [50], where the multiclass classification problem is decomposed to a number of binary class classification problems by separating each of the classes from the remaining classes. For SMILE, we implement two variants, called SMILE-B and SMILE-M. For SMILE-B, SMILE is extended to multiclass classification by performing one-against-all decomposition, like the baselines. For SMILE-M, SMILE is applied to multiclass classification by using a uniform multiclass classification formulation, as shown in Section V-B.

The SIVAL dataset[4] is used to evaluate the performance of SMILE and baselines on multiclass classification problems. It contains 1500 images of 25 categories, with 60 images for each category. Category 1 to

[4] Available at http://accio.cse.wustl.edu/sg-accio/SIVAL.html

category 25 are: “AjaxOrange,” “Apple,” “Banana,” “BlueScrunge,” “CandleWithHolder,” “CardboardBox,” “CheckeredScarf,” “CokeCan,” “DataMiningBook,” “DirtyRunningShoe,” “DirtyWorkGloves,” “FabricSoftenerBox,” “FeltFlowerRug,” “GlazedWoodPot,” “GoldMedal,” “GreenTeaBox,” “JuliesPot,” “LargeSpoon,” “RapBook,” “SmileyFaceDoll,” “SpriteCan,” “StripedNoteBook,” “TranslucentBowl,” “WD40Can,” and “WoodRollingPin.” The categories are complex objects photographed against ten highly diverse backgrounds. Six images are taken for each object-background pair. The objects may occur anywhere spatially in the image and also may be photographed at a wide-angle or close up.

The performance on the SIVAL dataset is shown in Fig. 7. By comparing EM-DD, mi-SVM, MI-SVM, DD-SVM, SMILE-B, and SMILE-M, it is found that SMILE-B delivers better classification accuracy than the baselines, which is consistent with the observations in binary class classification, since all these methods are extended to multiclass classification by decomposing the multiclass classification problem into several binary class classification problems. Moreover, SMILE-M performs slightly better than SMILE-B, which indicates that the uniform multiclass classification formulation presented in Section V-B is effective for MIL multiclass classification problems.

In sum, the number of positive instances in positive bags is usually unknown. If the classifier is obtained using only a subset of instances in the training bags, as done in the existing MIL works, e.g., EM-DD and MI-SVM, the discriminative power of the classifier may be limited. Distinguished from these methods, we explicitly incorporate the ambiguous instances in the classifier construction, by considering their similarity to the positive and negative classes. At the same time, SMILE shows better robustness than comparable methods.

VII. Conclusion and Future Work

A. Contribution of This Paper

A significant number of existing multi-instance learning methods proposes the selection of one or several less ambiguous instances to replace the whole bag for training the classifier. Different from these methods, in this paper, we put forward a novel MIL approach, termed SMILE. SMILE explicitly dealt with the ambiguous instances by assigning them different similarity weights to the positive class and the negative class. The ambiguous instances, as well as their similarity weights, were then incorporated into a heuristic learning framework. The characteristic of SMILE is that when we have no prior knowledge of the number of positive and negative instances in a positive bag, it is more appropriate to incorporate the ambiguous instances into the learning phase by considering their similarity to the classes, rather than selecting only a subset of less ambiguous instances, as done in some previous MIL works. Experiments on real-world datasets showed that SMILE delivers consistently better classification accuracy and robustness than comparable methods.

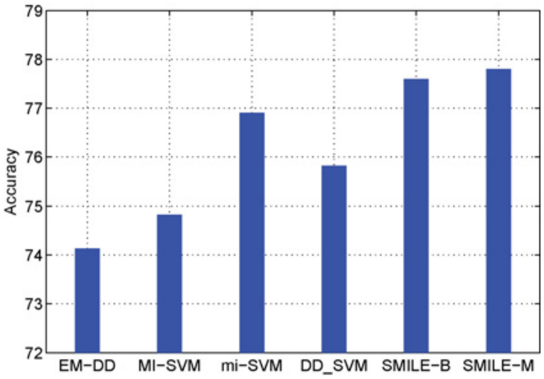

Fig. 7. Results of multiclass classification on the SIVAL dataset.

B. Limitation and Future Work

There are several limitations that may restrict the use of SMILE in applications and we plan to extend this paper in the following directions in the future.

1) *Similarity Weight Generation:* Similarity weight generation is an important step in our proposed approach. A desirable similarity weight generation method can appropriately measure the similarity of ambiguous instances to the classes and hence bring more classification information in the learning phase. Nevertheless, an undesirable similarity weight generation method could result in inaccurate classification information and make the classifier biased. Therefore, we will investigate other methods to generate the similarity weights according to different application problems.

2) *Computational Complexity:* For a training set that consists of n_1 positive bags and n_2 negative bags with each bag containing m instances, the computational complexity of SMILE in one iteration is $O(n_1+n_2*m+2n_1*(m-1))^2$. Here, n_1 is the number of positive candidates whose labels are likely to be positive. n_2*m is the number of instances in negative bags and the labels of these instances are negative. $n_1*(m-1)$ is the number of ambiguous instances of which the similarity to both the positive and negative classes is taken into account. The time complexity $O(n_1+n_2*m+2n_1*(m-1))^2$ can be rewritten as $O(n_1*m+n_2*m+n_1*(m-1))^2$. Compared with $O(n_1*m+n_2*m)^2$ for standard SVM, SMILE needs to solve a larger QP problem with a $n_1*(m-1)$ difference. This is because $n_1*(m-1)$ ambiguous instances are considered not only in the positive class, but also in the negative class.

In each learning iteration, the training efficiency of SMILE is lower than mi-SVM and MI-SVM, since mi-SVM trains a standard SVM, while MI-SVM uses an instance to replace a bag. It takes 37.7 s to train SMILE on the Musk1 dataset that includes 92 bags and 476 instances, compared with mi-SVM (11.8 seconds) and MI-SVM (3.2 seconds). Though SMILE has a higher computational cost, it is able to obtain markedly higher classification accuracy and better robustness than mi-SVM and MI-SVM, which can be observed in Sections VI-B and VI-C. The user can tradeoff the discriminatory power and computational effort. Additionally, many learning methods [51], [52] are proposed to speed up SVM. They can be adopted to accelerate the training efficiency of SMILE. Although the learning efficiency of SMILE is lower than mi-SVM and MI-SVM, it is much faster than EM-DD and DD-SVM. The training time of SMILE on the Musk1 dataset is 37.7 s, which is lower than EM-DD (141.6 s) and DD-SVM (500 min).

Moreover, we use a Gaussian distance to define the single set-based similarity in Definition 1. In the future, we would like to use more distance measurements, such as cosine and Laplacian measurements, and investigate their performances in different real-world applications.

Acknowledgments

The authors would like to thank the anonymous referees for their very useful comments and suggestions.

References

[1] T. Dietterich, R. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial 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 multiple-instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 12, pp. 2084–2098, Dec. 2008.

[3] B. Babenko, M. H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[4] G. C. Moore, 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.

[5] M. Kim and F. D. L. Torre, "Gaussian processes multiple instance learning," in *Proc. ICML*, 2010, pp. 535–542.

[6] T. Deselaers and V. Ferrari, "A conditional random field for multiple-instance learning," in *Proc. ICML*, 2010, pp. 287–294.

[7] Q. Zhang, S.A. Goldman, W. Yu, and J. Fritts, "Content-based image retrieval using multiple-instance learning," in *Proc. ICML*, 2002, pp. 682–689.

[8] D. Zhang, F. Wang, Z.W. Shi, and C.S. Zhang, "Interactive localized content based image retrieval with multiple-instance active learning," *Pattern Recogn.*, vol. 43, no. 2, pp. 478–484, 2010.

[9] Y. Zhang, A.C. Surendran, J.C. Platt, and M. Narasimhan, "Learning from multi-topic web documents for contextual advertisement," in *Proc. ACM SIGKDD Int. Conf. KDD*, 2008, pp. 1051–1059.

[10] O. Maron and A. L. Ratan, "Multiple-instance learning for natural scene classification," in *Proc. ICML*, 1998, pp. 341–349.

[11] Z.-H. Zhou and M.-L. Zhang, "Multi-instance multi-label learning with application to scene classification," in *Proc. Adv. NIPS*, 2006, pp. 1609–1616.

[12] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. Adv. NIPS*, 2003, pp. 561–568.

[13] Z.-H. Zhou, K. Jiang, and M. Li, "Multi-instance learning based web mining," *Applied Intell.*, vol. 22, no. 2, pp. 135–147, 2005.

[14] P. Auer and R. Ortner, "A boosting approach to multiple instance learning," in *Proc. ECML*, 2004, pp. 63–74.

[15] Y. Chevaleyre and J.-D. Zucker, "A framework for learning rules from multiple instance data," in *Proc. ECML*, 2001, pp. 49–60.

[16] J. T. Kwok and P.-M. Cheung, "Marginalized multi-instance kernels," in *Proc. IJCAI*, 2007, pp. 901–906.

[17] H. Blockeel, D. Page, and A. Srinivasan, "Multi-instance tree learning," in *Proc. ICML*, 2005, pp. 57–64.

[18] S. Ray and M. Craven, "Supervised versus multiple instance learning: An empirical comparison," in *Proc. ICML*, 2005, pp. 697–704.

[19] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *Proc. Adv. NIPS*, 1998, pp. 570–576.

[20] D. Wang, J. Li, and B. Zhang, "Multiple-instance learning via random walk," in *Proc. ECML*, 2006, pp. 473–484.

[21] Q. Zhang and S. Goldman, "EM-DD: An improved multiple-instance learning technique," in *Proc. Adv. NIPS*, 2002, pp. 1073–1080.

[22] W. Li and D. Yeung, "MILD: Multiple-instance learning via disambiguation," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 1, pp. 76–89, Jan. 2010.

[23] P. Auer. "On learning from multi-instance examples: Empirical evaluation of a theoretical approach," in *Proc. ICML*, 1997, pp. 21–29.

[24] P.M. Long and L. Tan, "PAC learning axis aligned rectangles with respect to product distributions from multiple-instance examples," *Mach. Learning*, vol. 30, no. 1, pp. 7–21, 1998.

[25] J. Ramon and L. D. Raedt, "Multi-instance neural networks," in *Proc. ICML—Workshop Attribute-Value Relational Learning*, 2000, pp. 53–60.

[26] G. Ruffo, "Learning single and multiple decision trees for security applications," in Ph.D. dissertation, Univ. Turin, Turin, Italy, 2000.

[27] D. R. Dooly, Q. Zhang, S. A. Goldman, and R. A. Amar, "Multiple instance learning of real valued data," *J. Mach. Learning Res.*, vol. 3, pp. 651–678, Mar. 2003.

[28] 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.

[29] J. Bolton, P. Gader, H. Frigui, and P. Torrione, "Random set framework for multiple instance learning," *Inform. Sci.*, vol. 181, no. 11, pp. 2061–2070, 2011.

[30] P. Viola, J. C. Platt, and C. Zhang, "Multiple instance boosting for object detection," in *Proc. Adv. NIPS*, 2006, pp. 1417–1424.

[31] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *Proc. ICML*, 1996, pp. 148–156.

[32] Y. Chen and J. Wang, "Image categorization by learning and reasoning with regions," *J. Mach. Learning Res.*, vol. 5, pp. 913–939, Dec. 2004.

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

[34] 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.

[35] Z. Zhou and J. Xu, "On the relation between multi-instance learning and semi-supervised learning," in *Proc. ICML*, 2007, pp. 1167–1174.

[36] S. Andrews and T. Hofmann, "A cutting-plane algorithm for learning from ambiguous examples," Brown Univ., Tech. Rep. CS-06-06, 2006.

[37] S. Andrews and T. Hofmann. "Multiple-instance learning via disjunctive programming boosting," in *Proc. Adv. NIPS*, 2004, pp. 65–72.

[38] X. Xu and E. Frank, "Logistic regression and boosting for labeled bags of instances," in *Proc. PAKDD*, 2004, pp. 272–281.

[39] J. Wang and J.-D. Zucker, "Solving the multiple instance problem: A lazy learning approach," in *Proc. ICML*, 2000, pp. 1119–1125.

[40] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[41] J. Bezdek and R. Hathaway, "Convergence of alternating optimization," *Neural, Parallel Scientific Comput.*, vol. 11, no. 4, pp. 351–368, 2003.

[42] J. Bi and T. Zhang, "Support vector classification with input data uncertainty," in *Proc. Adv. NIPS*, 2003, pp. 351–368.

[43] S.S. Keerthi and S.K. Shevade, "SMO algorithm for least squares SVM formulations," *Neural Comput.*, vol. 15, no. 2, pp. 487–507, 2003.

[44] C.C. Chang and C.J. Lin (2001), "Libsvm: A library for support vector machines," [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/libsvm, 2001.

[45] F. Murray Joseph, F. Hughes Gordon, and K.-D. Kenneth, "Machine learning methods for predicting failures in hard drives: A multiple-instance application," *J. Mach. Learning Res.*, vol. 6, pp. 783–816, May 2005.

[46] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li, "Multi-instance learning by treating instances as non-i.i.d. samples," in *Proc. ICML*, 2009, pp. 1249–1256.

[47] B. Settles, M. Craven, and S. Ray, "Multiple-instance active learning," in *Proc. Adv. NIPS*, 2008, pp. 1289–1296.

[48] F. Li and C. Sminchisescu, "Convex multiple-instance learning by estimating likelihood ratio," in *Proc. Adv. NIPS*, 2010, pp. 1360–1368.

[49] J. Weston and C. Watkins, "Support vector machines for multi-class pattern recognition," in *Proc. ESANN*, 1999, pp. 219–224.

[50] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.

[51] S. Fine and K. Scheinberg, "Efficient SVM training using low-rank kernel representation," *J. Mach. Learning Res.*, vo. 2, no. 2, pp. 243–264, 2001.

[52] Y. Lee and O.L. Mangasarian, "SSVM: A smooth support vector machine for classification," *Comput. Optim. Appl.*, vol. 20, no. 1, pp. 5–22, 1999.

Yanshan Xiao received the Ph.D. degree in computer science from the Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia, in 2011.

She is currently with the Faculty of Computer, Guangdong University of Technology, Guangzhou, China. Her current research interests include multiple-instance learning, support vector machine, and data mining.

Bo Liu is with the Faculty of Automation, Guangdong University of Technology, Guangzhou, China. His research interests include machine learning and data mining. He has published papers in IEEE TRANSACTIONS ON NEURAL NETWORKS, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, Knowledge and Information Systems, International Joint Conferences on Artificial Intelligence (IJCAI), IEEE International Conference on Data Mining (ICDM), SIAM International Conference on Data Mining (SDM) and ACM International Conference on Information and Knowledge Management (CIKM).

Zhifeng Hao (SM'04) is a Professor with the Faculty of Computer, Guangdong University of Technology, Guangzhou, China. His current research interests include design and analysis of algorithms, mathematical modeling, and combinatorial optimization.

Longbing Cao is a Professor with the Faculty of Information Technology, University of Technology, Sydney, Australia. His current research interests include data mining, multi-agent technology, and agent, and data mining integration.