2010 IEEE International Conference on Data Mining

SMILE: A Similarity-Based Approach for Multiple Instance Learning

Yanshan Xiao*, Bo Liu*, Longbing Cao*, Jie Yin† and Xindong Wu‡§
*QCIS Center, Faculty of Engineering and IT, University of Technology, Sydney, NSW 2007, Australia
† Information Engineering Laboratory, CSIRO ICT Centre, Australia
‡ Department of Computer Science, University of Vermont, Burlington, VT 05401, USA
§ School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China
syxiao@it.uts.edu.au; csbliu@it.uts.edu.au; longbing.cao-1@uts.edu.au; jie.yin@csiro.au; xwu@cems.uvm.edu

***Abstract*—Multiple instance learning (MIL) is a generalization of supervised learning which attempts to learn useful information from bags of instances. In MIL, the true labels of the instances in positive bags are not always available for training. This leads to a critical challenge, namely, handling the ambiguity of instance labels in positive bags. To address this issue, this paper proposes a novel MIL method named SMILE (Similarity-based Multiple Instance LEarning). It introduces a similarity weight to each instance in positive bag, which represents the instance similarity towards the positive and negative classes. The instances in positive bags, together with their similarity weights, are thereafter incorporated into the learning phase to build an extended SVM-based predictive classifier. Experiments on three real-world datasets consisting of 12 subsets show that SMILE achieves markedly better classification accuracy than state-of-the-art MIL methods.**

***Keywords*-Multiple Instance Learning;**

I. Introduction

Multiple instance learning (MIL) [1] is a new paradigm in data mining and machine learning that addresses the classification of bags. In MIL, the labels in the training set are associated with sets of instances, which are called bags. A bag is labelled positive if at least one of its instances is positive; otherwise, the bag is labelled negative. The task of MIL is to classify unknown bags by using the information of labelled bags. Many real-world applications involving ambiguous concepts can be elegantly solved by applying multi-instance learning. For instance, in the text categorization domain, since a document often contains more than one topic, it is appropriate to consider this task from a multiple instance view. The document is classified as positive if it contains at least one block which is related to the subject of interest.

In contrast to the standard supervised learning, the key challenge of MIL is that the label of any single instance in a positive bag can be unavailable. From the description of MIL, it can easily be seen that all instances in negative bags are negative. However, it does not require all the instances in a positive bag to be positive. This means that a positive bag may contain some 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 [2]. 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 [3], [4], [5], [6], [7] focuses on selecting a subset of instances from positive bags to learn the classifier. In the training phase, except for the selected instances, the remaining instances in positive bags are excluded from training. For example, RW-SVM [7] designs an instance selection mechanism to select one instance from each positive bag. Together with the negative instances from negative bags, these selected instances in positive bags are used to build the classifier. However, the discriminative ability of these approaches may be restricted. This is because only a subset of the instances is used to learn the classifier, while quite a number of remaining instances in positive bags, which may contribute to the construction of classifier, are excluded from the classifier learning.

In this paper, we propose a novel multiple instance learning method, termed SMILE (Similarity-based Multiple Instance LEarning). It appropriately utilizes the ambiguous instances in positive bags to improve MIL accuracy. Instead of excluding a number of ambiguous instances in positive bags from training, SMILE explicitly deals with ambiguous instances by considering their similarity to both the positive and negative classes. Specifically, we assign two similarity weights to each ambiguous instance towards the positive and negative classes. Then, we incorporate these ambiguous instances, together with their similarity weights, into an extended formulation of support vector machines (SVM). Based on a heuristic learning framework (see Section IV-E),

1550-4786/10 $26.00 © 2010 IEEE
DOI 10.1109/ICDM.2010.126

the selection of positive instances and similarity weights can be updated to refine the classification boundary.

The main contributions of our work can be viewed from the following aspects.

- We propose a novel approach to make the ambiguous instances, which are neglected by some previous MIL works [3]-[7], contribute to the learning of classifiers. Compared to the related MIL works [3]-[7], the incorporation of ambiguous instances enables a more powerful classifier with better discriminative ability.
- We use a similarity-based data model to assign similarity weights to each ambiguous instance in positive bags, so that the similarity of ambiguous instances to the positive and negative classes can be evaluated. Such information is thereafter incorporated into the classifier construction.
- We present an extended formulation of SVM to build the classifier. The extended SVM is capable of incorporating the similarity information of an ambiguous instance to the positive and negative classes in the optimization procedure.
- We evaluate our proposed approach on real-world datasets, and experimental results show that our proposed approach achieves markedly better accuracy than comparable MIL methods in various applications.

The rest of the paper is organized as follows. Section II reviews the works related to our study. The similarity-based data model and symbol definitions are presented in Section III. We detail the similarity-based approach for the MIL problem in Section IV. Experiments are conducted in Section V, and Section VI concludes the paper and offers possible future work.

II. Related Work

Since the proposed method SMILE is a SVM-based MIL approach, we first review the previous work on MIL, and then introduce the basic principle of the standard SVM.

A. Multiple Instance Learning Methods

Over the last decade, multiple instance learning has attracted much attention in solving many real-world applications, ranging from drug activity prediction [1], image categorization [8], [9], and text categorization [3], [2] to image retrieval [10]. The initial MIL algorithms are presented in [1], [11], [12], which are based on hypothesis classes consisting of axis-aligned rectangles. Following these works, many MIL methods from different perspectives have been proposed. In the following, we will briefly review some of the works in multiple instance learning.

The first category of works set the instance labels in positive bags as positive, and then the standard supervised learning method or an iterative framework is adopted to train the classifier. For example, Ray and Craven [2] label all the instances in positive bags as positive and the standard SVM is used to train the classifier straightforwardly. However, this method relies on the positive bags being fairly rich in positive instances. Rather than learning the classifier at one time, mi-SVM [3] and MILBoost [13] train the classifier iteratively to refine the classification boundary. In mi-SVM [3], the labels of instances in positive bags are initialized as positive, and then the classifier is trained repeatedly until each positive bag has at least one instance which is classified as positive by the classifier. Obviously, mi-SVM focuses on obtaining 100% training accuracy of positive bags. However, if labelling noise of the positive bags exists, the accuracy of mi-SVM may be greatly reduced. In MILBoost [13], all instances initially get the same label as the bag label for training the first classifier and each instance is assigned a weight indicating its label. At each round, the instance weights are updated by a line search to maximize the log likelihood function and subsequent classifiers are trained. However, the MILBoost is based on the framework of Boosting and may sometimes be less robust [14]. In the experiments, we explicitly compare the robustness of our proposed method and MILBoost.

The second category of works [8], [9] design mechanisms to map a bag of instances into a "bag-level" training vector, and each instance serves as a dimension in the new feature space. Typical examples include MILES [9] and DD-SVM [8]. In MILES, the bag is embedded into a new feature space. The 1-norm SVM is used to select the important features (instances) for prediction. In DD-SVM, it learns a collection of instance prototypes according to a Diverse Density (DD) function and the instance prototypes are used to map a bag into a point in a new feature space. Then, the standard SVM is used to build the classifier for separating the "bag-level" points. However, DD-SVM and MILES may transform the MIL into a high dimensionality problem. This is because the dimension of the "bag-level" training vector is equal to the total number of instances in the training set. If the number of instances is large, the "bag-level" training vector may turn out to be extremely high dimensional. Moreover, both DD-SVM and MILES include the computation of DD function, which is "very sensitive to noise", as pointed out by [6].

The third category of works [4], [5], [6], [7], [9] focus on selecting a subset of instances from positive bags to learn the classifier. For example, the DD method [4] aims to find one data point (target concept), which is nearest to the instances in positive bags and is farthest from the negative bags. A test bag is classified as positive if the distance between the selected data point and any of its instances is below a threshold. EM (Expectation-Maximization)-DD [5] chooses one instance which is most consistent with the current hypothesis in each positive bag to predict an unknown instance. MI-SVM [3] adopts an iterative framework to learn the classifier. At each iteration, only one instance from each positive bag is selected. Together with instances in negative bags, the selected instances are used to learn the classifier.

RW-SVM [7] select one instance from each positive bag. These selected instances and instances in negative bags are used to learn 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 amount of ambiguous instances, which may contribute to the construction of classifier, is neglected.

Moreover, some other methods are also employed to improve MIL classification accuracy [15], [16], [17], [18]. For example, MissSVM which is proposed by [15], considers the MIL as a semi-supervised learning problem. Citation-kNN [19] extends K nearest neighbors 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 neglected in those works, to learn the classifier. The incorporation of ambiguous instances makes our classifier more discriminative in classifying the positive and negative bags. Furthermore, we incorporate the ambiguous instances in the training by measuring its similarity to the positive and negative classes, and then the similarity weighted large margin learning method is used to learn the classifier, which is expected to gain stronger robustness than the DD-based MIL method and those works which focus on reducing the false positive rate. The experiments show that our proposed approach achieves markedly better classification accuracy and robustness than the comparable methods in the three categories.

B. Support Vector Machines

Support vector machines (SVM) [20] have been proven to be a powerful classification tool in data mining and machine learning areas. SVM seeks an optimal separating hyperplane which maximizes the margin between two classes after mapping the data into a feature space. In the following, we briefly review the principle of SVM in the standard supervised learning.

Let $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_{|S|}, y_{|S|})\}$ be a training set, where $\mathbf{x}_i \in R^m$, $y_i \in \{+1, -1\}$ and $|S|$ is the number of instances in S. In the SVM [20], a nonlinear mapping function $\phi(.)$ is used to map the dataset from the input space (R^m) into a feature space F, where both classes are expected to be much more linearly separable. The goal of SVM is to find $\mathbf{w}$ and b which lead to an optimized hyperplane [20]:

$$D_{w,b}(\mathbf{x}) = w^T \phi(\mathbf{x}) + b = 0. \tag{1}$$

The decision function (1) satisfies the following condition

$$\begin{aligned} &y_i(w^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, |S|, \\ &\xi_i \geq 0, \quad for \quad i = 1, \ldots, |S|. \end{aligned} \tag{2}$$

where ξ_i are introduced to relax the margin constraints. To obtain the SVM classifier, we need to solve the following problem:

$$\min F(w, b, \xi) = \frac{1}{2} w^T w + C \sum_{i:\mathbf{x}_i \in S} \xi_i, \tag{3}$$

subject to constraints (2), where C is a parameter which balances the margin and classification errors. By introducing the Lagrange function [20], w and b are solved, and the decision classifier (Equation (1)) is then obtained.

For a test instance $\mathbf{x}$, if $D_{w,b}(\mathbf{x}) > 0$, it is classified into the positive class; otherwise, it belongs to the negative class.

In our paper, each ambiguous instance is associated with two similarity weights. However, the standard SVM in Equation (3) can not handle two weights for one instance. To solve this problem, we present an extended formulation of the standard SVM. It can effectively incorporate both similarity weights into the optimization procedure.

III. Similarity-Based Data Model

Let $\{(B_1^+, Y_1^+), \ldots, (B_{N^+}^+, Y_{N^+}^+), (B_1^-, Y_1^-), \ldots, (B_{N^-}^-, Y_{N^-}^-)\}$ denote a set of training bags, where B_i^+ represents a positive bag with positive label $Y_i^+ = +1$; B_i^- denotes a negative bag with negative label $Y_i^- = -1$. N^+ and N^- are the numbers of positive and negative bags, respectively.

Each bag contains a set of instances. The j^{th} instance in B_i^+ and B_i^- is denoted as B_{ij}^+ and B_{ij}^-, respectively. n_i^+ and n_i^- represent the corresponding instance numbers in B_i^+ and B_i^-. 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 $D = \{(\mathbf{x}_i, y_i)\}$, $i = 1, 2, \ldots, l$, where l is the total number of training instances.

For an instance $\mathbf{x}$, we convert it to a similarity-based data model defined as follows:

$$\{\mathbf{x}, m^+(\mathbf{x}), m^-(\mathbf{x})\}, \tag{4}$$

where $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$ represent the different similarity of $\mathbf{x}$ towards the positive and negative classes, respectively. We have $0 \leq m^+(\mathbf{x}) \leq 1$ and $0 \leq m^-(\mathbf{x}) \leq 1$. $\{\mathbf{x}, 1, 0\}$ means that $\mathbf{x}$ belongs to the positive class, while $\{\mathbf{x}, 0, 1\}$ indicates that $\mathbf{x}$ is negative. For $\{\mathbf{x}, m^+(\mathbf{x}), m^-(\mathbf{x})\}$, where $0 < m^+(\mathbf{x}) < 1$ and $0 < m^-(\mathbf{x}) < 1$, it implies that the similarity of $\mathbf{x}$ towards the positive and negative classes are both considered.

Using the above similarity-based data model, we can convert a multiple instance learning problem into a single instance learning problem. This makes it possible for the supervised learning methods adapted to solve the MIL problem.

IV. The SMILE Approach

A. SMILE framework

Given a set of training instances, the objective of SMILE is to build a classifier using both the positive and negative bags. The classifier is thereafter applied to classify the

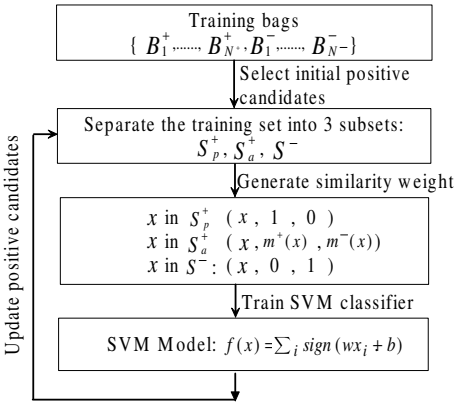


Figure 1. The overview of our proposed approach

coming test bag. Figure 1 illustrates the SMILE's working process, which operates as follows.

Initially, some instances from positive bags are selected as positive candidates. The original dataset is thereafter separated into three subsets S_p^+, S_a^+ and S^-, where S_p^+ consists of all the selected positive candidates; S_a^+ contains the rest unselected instances in positive bags; S^- consists of the instances belonging to negative bags. Secondly, the similarity weights are assigned to instances in subsets S_p^+, S_a^+ and S^-. For 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 each instance in S_a^+, it is assigned similarity weights $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$ towards the positive and negative classes, respectively. Thirdly, the extended formulation of standard SVM is put forward to learn a MIL classifier based on the presented data model. Finally, positive candidates are reselected and the SVM model is updated until the termination criterion is met.

In order to simplify the presentation, we first let $S^+ = S_p^+ + S_a^+$ and $S^{*-} = S_a^+ + S^-$. The SMILE algorithm consists of four aspects as discussed in Sections B to E.

B. Initial Positive Candidates Selection

From the description of MIL problem, each positive bag must contain at least one positive instance. Therefore, it is possible to initially choose one positive candidate from each positive bag [6]. In our proposed approach, an instance, which is most likely to be a positive instance, will be selected out from each positive bag and are put into subset S_p^+. The selection of such positive candidates relies on the following definitions: *Single Set-Based Similarity* and *Similarity*.

Definition 1: (Single Set-Based Similarity) Given an instance $\mathbf{x}$ and a subset S, the single set-based similarity between $\mathbf{x}$ and S is defined in Equation (5).

$$R(\mathbf{x}, S) = \frac{1}{|S|} \sum_{\mathbf{x}_i \in S} e^{-||\phi(\mathbf{x}) - \phi(\mathbf{x}_i)||^2}. \quad (5)$$

where $|S|$ denotes the sample size of S; $\phi(\mathbf{x}_i)$ is the image of instance $\mathbf{x}_i$ in the feature space and

$$||\phi(\mathbf{x}) - \phi(\mathbf{x}_i)||^2 = K(\mathbf{x}, \mathbf{x}) + K(\mathbf{x}_i, \mathbf{x}_i) - 2K(\mathbf{x}, \mathbf{x}_i),$$

where $K(.,.)$ is a mercer kernel function.

It can be observed that, if an instance $\mathbf{x}$ is close to S, the single set-based similarity between $\mathbf{x}$ and S is naturally high. By contrast, its single set-based similarity with S becomes low. It is easy to see that the value of similarity $R(\cdot, \cdot)$ falls into the range of $[0, 1]$.

Definition 2: (Similarity) Given instance $\mathbf{x}$, subsets S^+ and S^-, the similarity Q of instance $\mathbf{x}$ towards S^+ is defined as follows:

$$Q(\mathbf{x} \in S^+ | S^+ \cup S^-) = \frac{1}{2}[R(\mathbf{x}, S^+) + 1 - R(\mathbf{x}, S^-)]. \quad (6)$$

where S^+ and S^- are subsets containing the training instances from positive bags and negative bags, respectively. Since the similarity of $\mathbf{x}$ is related to the relative location of S^+ and S^-, its similarity with S^+ and S^- are both considered in Equation (6). In Equation (6), $R(\mathbf{x}, S^+)$ is the single set-based similarity of $\mathbf{x}$ and S^+. If the value of $R(\mathbf{x}, S^+)$ is larger, it indicates that $\mathbf{x}$ more likely belongs to S^+. $R(\mathbf{x}, S^-)$ represents the similarity between $\mathbf{x}$ and S^-, and hence $1 - R(\mathbf{x}, S^-)$ can be considered as the dissimilarity of $\mathbf{x}$ with S^-.

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

$$\arg \max_{\mathbf{x} \in B_i^+} Q(\mathbf{x} \in S^+ | S^+ \cup S^-). \quad (7)$$

Similar to MILD [6], we select one instance from each positive bag as the positive candidate. Intuitively, the instance, which is most similar to instances in the other positive bags and has least similarity to those in the negative bags, is more likely to be a positive candidate, compared with the rest instances in the same bag. Therefore, the instance with maximum value in (7) is chosen to be the positive candidate.

After 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 rest of unselected positive bag instances, whose labels are relatively ambiguous compared to the positive candidates.

C. Similarity Weight Generation

This section introduces the method to generate similarity weights for training instances. First, according to the MIL data model presented in Section III, the instances in subset S^- are assigned with $m^+(\mathbf{x}) = 0$ and $m^-(\mathbf{x}) = 1$, while those in S_p^+ have $m^+(\mathbf{x}) = 1$ and $m^-(\mathbf{x}) = 0$.

For each instance $\mathbf{x}$ in S_a^+, its corresponding similarity weights towards the positive and negative classes are calculated as follows:

$$\begin{aligned} m^+(\mathbf{x}) &= Q(\mathbf{x} \in S_p^+ | S_p^+ \cup S^-) \\ &= \frac{1}{2}[R(\mathbf{x}, S_p^+) + 1 - R(\mathbf{x}, S^-)], \quad (8) \\ m^-(\mathbf{x}) &= Q(\mathbf{x} \in S^- | S_p^+ \cup S^-) \\ &= \frac{1}{2}[R(\mathbf{x}, S^-) + 1 - R(\mathbf{x}, S_p^+)]. \quad (9) \end{aligned}$$

For each instance $\mathbf{x}$ in S_a^+, we compute its similarity weights towards the positive class in Equation (8) and towards the negative class in Equation (9), which are obtained by considering its similarity with S_p^+ and S^-. Additionally, we have $m^+(\mathbf{x}) + m^-(\mathbf{x}) = 1$ based on Equations (8) and (9).

D. Extended Formulation of Standard SVM

In our MIL data model, training instances are assigned similarity weights towards the positive and negative classes, respectively. To incorporate such similarity weights into the learning process, an extended formulation of the standard SVM is given as follows:

$$\begin{aligned} \min \quad & F(w, b, \xi) \\ & = \frac{1}{2} w^T w + C_1 \sum_{i:\mathbf{x}_i \in S_p^+} \xi_i + C_2 \sum_{j:\mathbf{x}_j \in S_a^+} m^+(\mathbf{x}_j)\xi_j \\ & + C_3 \sum_{k:\mathbf{x}_k \in S_a^+} m^-(\mathbf{x}_k)\xi_k^* + C_4 \sum_{g:\mathbf{x}_g \in S^-} \xi_g \\ s.t. \quad & w^T \phi(\mathbf{x}_i) + b \geq 1 - \xi_i, \\ & w^T \phi(\mathbf{x}_j) + b \geq 1 - \xi_j, \\ & w^T \phi(\mathbf{x}_k) + b \leq -1 + \xi_k^*, \\ & w^T \phi(\mathbf{x}_g) + b \leq -1 + \xi_g, \\ & \xi_i \geq 0,\ \xi_j \geq 0,\ \xi_k^* \geq 0,\ \xi_g \geq 0. \quad (10) \end{aligned}$$

where ξ_i, ξ_j, ξ_k^* and ξ_g are the measure of errors; $m^+(\mathbf{x}_j)\xi_j$ 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 penalty factors which control the tradeoff between the hyperplane margin and the errors. In addition, for instances in S_p^+, since their similarity weights towards the negative class are zero, the weighted errors are $C_1 \sum_{i:\mathbf{x}_i \in S_p^+} \xi_i$. For instances in S^-, since their similarity weights towards the positive class are zero, the weighted errors are $C_4 \sum_{g:\mathbf{x}_g \in S^-} \xi_g$. For instances in S_a^+, they have non-zero similarity weights towards the positive and negative classes, and hence the weighted errors are given as $C_2 \sum_{j:\mathbf{x}_j \in S_a^+} \xi_j + C_3 \sum_{k:\mathbf{x}_k \in S_a^+} \xi_k^*$.

Assume α_i, α_j, α_k^* and α_g are Lagrange multipliers. In order to simplify the presentation, we redefine some notations in the following:

$$\alpha_i^+ = \begin{cases} \alpha_i, & i : \mathbf{x}_i \in S_p^+ \\ \alpha_j, & j : \mathbf{x}_j \in S_a^+ \end{cases} \quad (11)$$

$$\alpha_j^- = \begin{cases} \alpha_k^*, & k : \mathbf{x}_k \in S_a^+ \\ \alpha_g, & g : \mathbf{x}_g \in S^- \end{cases} \quad (12)$$

$$C_i^+ = \begin{cases} C_1, & i : \mathbf{x}_i \in S_p^+ \\ C_2 m^+(\mathbf{x}_j), & j : \mathbf{x}_j \in S_a^+ \end{cases} \quad (13)$$

$$C_j^- = \begin{cases} C_2 m^-(\mathbf{x}_k), & k : \mathbf{x}_k \in S_a^+ \\ C_3, & g : \mathbf{x}_g \in S^- \end{cases} \quad (14)$$

Based on the above definitions, the Wolfe dual of (10) can be obtained as follows:

$$\begin{aligned} \min \ F(\alpha) = \quad & \frac{1}{2} \sum_{i:\mathbf{x}_i \in S^+} \sum_{j:\mathbf{x}_j \in S^{*-}} (\alpha_i^+ - \alpha_j^+)^2 K(\mathbf{x}_i, \mathbf{x}_j) \\ & - \sum_{i:\mathbf{x}_i \in S^+} \alpha_i^+ - \sum_{j:\mathbf{x}_j \in S^{*-}} \alpha_j^+ \\ s.t. \quad & \sum_{i:\mathbf{x}_i \in S^+} \alpha_i^+ = \sum_{j:\mathbf{x}_j \in S^{*-}} \alpha_j^-, \\ & 0 \leq \alpha_i^+ \leq C_i^+,\ i : \mathbf{x}_i \in S^+, \\ & 0 \leq \alpha_j^- \leq C_j^-,\ j : \mathbf{x}_j \in S^{*-}. \quad (15) \end{aligned}$$

Moreover, w can be obtained by differentiating the Lagrangian function with w, b and ξ, as shown in the following:

$$w = \sum_{i:\mathbf{x}_i \in S^+} \alpha_i^+ \phi(\mathbf{x}_i) - \sum_{j:\mathbf{x}_j \in S^{*-}} \alpha_j^- \phi(\mathbf{x}_j). \quad (16)$$

After solving the dual form (15) and substituting w into the classification problem, the classifier for classifying the instances can be obtained in (17).

$$L(\mathbf{x}_i) = \begin{cases} +1, & w^T \phi(\mathbf{x}_i) + b \geq 0, \\ -1, & w^T \phi(\mathbf{x}_i) + b < 0. \end{cases} \quad (17)$$

where $\phi(\mathbf{x}_i)$ is a test instance; $L(\mathbf{x}_i)$ represents the predicted label of $\phi(\mathbf{x}_i)$.

The objective of MIL is to classify the bags. Based on the description of the MIL problem, the classifier for classifying the bags is obtained as follows:

$$L(B) = \begin{cases} +1, & \sum_{x_i \in B} L(\mathbf{x}_i) > -|B|, \\ -1, & \sum_{x_i \in B} L(\mathbf{x}_i) = -|B|. \end{cases} \quad (18)$$

where B is a test bag; $L(B)$ denotes the predicted label of B; $|B|$ is the number of instances in B. From (18), it can be seen that only if all the instances in the bag are predicted negative, i.e. $L(\mathbf{x}_i) = -|B|$, bag B is predicted as negative. Otherwise, if not all the instances are negative, i.e. $L(\mathbf{x}_i) > -|B|$, bag B will be classified as positive.

E. Positive Candidate Update and Heuristic Strategy

Similar to mi-SVM [3], a heuristic strategy is adopted to update the positive candidates and to refine the decision boundary. The heuristic strategy is based on the alternating optimization method [21], [22].

Following the heuristic strategy of the alternating optimization method, we first obtain the initial positive candidates, as described in Section IV-B, and then the following steps repeat until a termination criterion is met:

1) Fixing the obtained positive candidates, generate similarity weights according to Equations (8) and (9), and then solve the optimization problem (15) to obtain Lagrangian multipliers $\alpha = \{\alpha_i^+, \alpha_j^-\}$.
2) Fixing the obtained Lagrangian multipliers α, update the positive candidates as follows:

$$
\begin{aligned}
i_1^{(t+1)} = \quad & \arg \min_{g=1}^{n_1^+} F\{\alpha^{(t)}, \mathbf{x}_g, \mathbf{x}_{i_2^{(t)}}, \mathbf{x}_{i_3^{(t)}}, \ldots, \\
& \mathbf{x}_{i_{N^+}^{(t)}}\}, \ \mathbf{x}_g \in B_1^+, \\
i_2^{(t+1)} = \quad & \arg \min_{g=1}^{n_2^+} F\{\alpha^{(t)}, \mathbf{x}_{i_1^{(t+1)}}, \mathbf{x}_g, \mathbf{x}_{i_3^{(t)}}, \ldots, \\
& \mathbf{x}_{i_{N^+}^{(t)}}\}, \ \mathbf{x}_g \in B_2^+, \\
& \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
i_k^{(t+1)} = \quad & \arg \min_{g=1}^{n_k^+} F\{\alpha^{(t)}, \mathbf{x}_{i_1^{(t+1)}}, \ldots, \mathbf{x}_{i_{k-1}^{(t+1)}}, \\
& \mathbf{x}_g, \mathbf{x}_{i_{k+1}^{(t)}}, \ldots, \mathbf{x}_{i_{N^+}^{(t)}}\}, \mathbf{x}_g \in B_k^+, \\
& \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
i_{N^+}^{(t+1)} = \quad & \arg \min_{g=1}^{n_{N^+}^+} F\{\alpha^{(t)}, \mathbf{x}_{i_1^{(t+1)}}, \mathbf{x}_{i_2^{(t+1)}}, \ldots, \\
& \mathbf{x}_{i_{N^+-1}^{(t+1)}}, \mathbf{x}_g\}, \ \mathbf{x}_g \in B_{N^+}^+. \qquad (19)
\end{aligned}
$$

where $i_k^{(t)}$ and $i_k^{(t+1)}$ are indexes of positive candidates in bag B_k^+ at the t^{th} and $(t+1)^{th}$ iterations, respectively. Hence, $\mathbf{x}_{i_k^{(t)}}$ and $\mathbf{x}_{i_k^{(t+1)}}$ are positive candidates of bag B_k^+ at the corresponding t^{th} and $(t+1)^{th}$ iterations. $\alpha^{(t)} = \{\alpha_i^{+(t)}, \alpha_j^{-(t)}\}$ are Lagrangian multipliers obtained from dual form (15) at the t^{th} iteration.

After substituting (11) and (13) into dual form (15), it can be seen that the value of F is determined by α, S^+, S^{*-}, $m^+(x)$ and $m^-(x)$. Moreover, S^+ and S^{*-} can be easily obtained if S_p^+ is determined, because $S^+ = S_p^+ + S_a^+$ and $S^{*-} = S_a^+ + S^-$, where S^- will keep unchanged throughout the iterations, and $S_a^+ = D - S_p^+ - S^-$. $m^+(x)$ and $m^-(x)$ are decided by S_p^+ and S^- based on (8) and (9). Therefore, after α and S_p^+ are determined, the F value can be figured out. For this reason, we only list α and positive candidates $\mathbf{x}_{i_k^{(t)}}$ in (19).

To update the positive candidate in a positive bag, we will select one instance, which leads to minimum F value in (15), as a new positive candidate. To illustrate how the positive instance is updated, let us take B_k^+ as an example. For any instance $\mathbf{x}_g$ in B_k^+, firstly, it is assumed to be the positive instance of B_k^+. Secondly, similarity weights $m^+(\mathbf{x})$ in (8) and $m^-(\mathbf{x})$ in (9) are recalculated based on $\mathbf{x}_g$ and the positive instances in other positive bags. Thirdly, the F value is computed by substituting $\alpha^{(t)}$, newly updated $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$ in (15). It is noted that no quadratic programming (QP) is required to compute the F value in this step, since $\alpha^{(t)}$ is given. Finally, for all instances in B_k^+, the instance with minimum F value is selected as the positive candidate of B_k^+.

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

$$F^{(t)} - F^{(t-1)} \leq \epsilon F^{(t)} \qquad (20)$$

where $F^{(t)}$ and $F^{(t-1)}$ are the values of F at the $(t+1)^{th}$ and t^{th} iterations, respectively; ϵ is a threshold. Since the value of F is nonnegative [23], with the decreasing of F, $(F^{(t)} - F^{(t-1)})/F^{(t)}$ will be smaller than a threshold ϵ. ϵ is set to be 0.01 in our experiments.

The SMILE approach is presented in Algorithm 1.

V. Experiments and Evaluation

Extensive experiments have been conducted on three benchmark datasets: MUSK, image retrieval and text categorization datasets [1]. The three datasets in total include 12 subsets, which have been commonly used in previous MIL works [3], [7].

A. Baseline Methods

In our experiments, SMILE is compared with four baseline approaches: EM-DD [5], DD-SVM [8], mi-SVM [3] and MILBoost [13].

The first one is EM-DD [5] which focuses on selecting one instance from each bag to predict an unknown instance. The second one is DD-SVM [8] which maps a bag of instances into a "bag-level" vector and constructs a "bag-level" classifier, so that all points in positive bags contribute to the prediction. Chen et al. [8] have shown that DD-SVM obtains much better accuracy than MI-SVM [3] and we mainly discuss DD-SVM in our experiments. The third one is mi-SVM [3] which aims at achieving 100% training accuracy of positive bags. The fourth one is MILBoost [13] in which each instance is assigned a weight and the instance weights are updated iteratively to train the classifier. Following the setting in [24], Naive Bayes is used as the base classifier in MILBoost, since "it is not straightforward to incorporate the instance weights (of MILBoost) into an SVM solver" [24]. These baseline methods are used to test the ability of SMILE on several real-world MIL datasets.

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

Algorithm 1 SMILE Method for Multi-Instance Problem.

Let D contain instances from all training bags; S^- contains instances from all negative bags; let S_p^+ and S_a^+ be empty;
Initialize one positive candidate from each positive bag as described in Section IV-B;
Put the initialized positive candidates in S_p^+;
t=0;
repeat
  t=t+1;
  Set $S_a^+ = D - S_p^+ - S^-$;
  Compute $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$ according to Equations (8) and (9);
  Obtain α by solving QP in (15) based on S^-, S_p^+ and S_a^+ with $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$;
  Let $\alpha^{(t)} = \alpha$;
  for (each positive bag B_g^+) **do**
    for (each instance $\mathbf{x}_j$ in B_g^+) **do**
      Let $\mathbf{x}_j$ be the positive candidate of B_g^+;
      Update S_p^+ with $\mathbf{x}_j$;
      Recompute $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$ based on the updated S_p^+;
      Calculate the value of F, denoted as $F(\mathbf{x}_j)$, by substituting $\alpha^{(t)}$, $m^+(\mathbf{x})$ and $m^-(\mathbf{x})$ in (15);
    end for
    $i_g^{(t+1)}$=arg $\min F(\mathbf{x}_j)$, $\mathbf{x}_j \in B_g^+$;
    Update S_p^+ with $\mathbf{x}_{i_g^{(t+1)}}$;
  end for
  Set $F^{(t)}$=$F\{\alpha^{(t)}, \mathbf{x}_{i_1^{(t+1)}}, \mathbf{x}_{i_2^{(t+1)}}, \ldots, \mathbf{x}_{i_{N^+}^{(t+1)}}\}$;
until $F^{(t)} - F^{(t-1)} \leq \epsilon F^{(t)}$
OUTPUT (w,b)

B. Experimental Settings

We follow the experimental settings in [3] to design our experiments. Firstly, RBF kernel (21) is used for SVM related methods on MUSK, image retrieval and text categorization datasets. Secondly, since Poly kernel (22) is generally known to work well for image and text categorization, the results on image retrieval and text categorization datasets with Poly kernel are also reported. Thirdly, ten-fold cross-validation is applied and the average accuracy is presented.

$$K(\mathbf{x}_i, \mathbf{x}_j) = exp\{-||\mathbf{x}_i - \mathbf{x}_j||^2/\sigma\}, \quad (21)$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + r)^d. \quad (22)$$

In RBF and Poly kernel functions, σ, γ, r and d are kernel parameters. Following the same parameter selection routine in [8] and [15], we let γ and σ selected from 2^{-5} to 2^5. r is chosen from 0 to 9. d is selected from 1 to 10. [2] As for parameters C_1, C_2, C_3 and C_4 in the objective function (10), we let $C_1 = C_2$ and $C_3 = C_4$, and each of them is selected from 2^{-5} to 2^5. Following the experimental setting in [24], MILBoost is run through 30 boosting iterations which end up with an ensemble of 30 classifiers.

[2]For Poly kernel (22), if $\gamma = 1$, $r = 0$ and $d = 1$, Poly kernel degenerates to linear kernel, i.e., $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$; hence we implicitly include linear kernel in the experiments.

C. Musk Dataset

The Musk dataset contains two subsets: Musk1 and Musk2. Musk1 has 47 positive bags and 45 negative bags with about 5 instances per bag. Musk2 has 39 positive bags and 63 negative bags. The number of instances in each bag is much larger, ranging from 1 to 1,044 (about 64 instances per bag on average). Each instance is represented by a 166-dimensional feature vector.

Table I
AVERAGE ACCURACY ON MUSK1 AND MUSK2 DATASETS.

	Musk1	**Musk2**
APR [1]	**92.4**	89.2
EM-DD [5]	84.8	84.9
MILBoost [13]	82.3	85.7
mi-SVM [3]	87.4	83.6
DD-SVM [8]	85.8	91.3
SMILE	91.3	**91.6**

As shown in Table I, the average accuracy on Musk1 and Musk2 datasets utilizing 10-fold cross-validation is given. From Table I, the classification accuracy of SMILE on Musk 1 is 91.3%, which outperforms most methods, except for the APR method. The reason APR has the highest accuracy is that the APR method has been designed particularly for the drug activity prediction problem [3], [6]. On the Musk2 dataset, SMILE obtains the best classification accuracy at 91.6% which is 5.9%, 8% higher than MILBoost and mi-SVM, respectively. Though DD-SVM has similar accuracy as SMILE on Musk2, SMILE significantly outperforms DD-SVM at 5.5% on Musk1.

In addition, it is seen that the accuracy of EM-DD on Musk1 and Musk2 datasets is 84.8% and 84.9% respectively, which is lower than SMILE by around 6.7% on average. This is because EM-DD only focuses on selecting one instance from each bag for prediction, while a large number of remaining instances in bags, which can be used to boost MIL accuracy, is neglected. In contrast to EM-DD, SMILE explicitly utilizes these ambiguous instances and achieves markedly better classification accuracy than EM-DD.

D. Image Retrieval Dataset

The image retrieval dataset consists of 3 subsets: Elephant, Tiger and Fox. In each subset, it contains 100 positive bags and 100 negative bags of about 1300 instances. Each bag represents an image and the instances in a bag represent different segmented blobs of the corresponding image.

The experimental results on Tiger, Elephant and Fox datasets by using the RBF and Poly kernels are reported in Table II. As mentioned earlier, the results with Poly kernel in our experiments have already included those of linear

Table II
AVERAGE ACCURACY ON ELEPHANT, TIGER AND FOX DATASETS.

		Tiger	Elephant	Fox
EM-DD		72.1	78.3	56.1
MILBoost		75.3	80.7	55.2
mi-SVM	RBF	78.9	80	57.9
	Poly	78.4	82.2	58.2
DD-SVM	RBF	75.8	81.6	55.7
	Poly	77.2	83.5	56.6
SMILE	RBF	**86.5**	84.3	65.9
	Poly	86.2	**85.8**	**67.7**

kernel. From Table II, the corresponding accuracy of SMILE on Tiger, Elephant and Fox datasets is 86.5%, 84.3% and 65.9% with RBF kernel, while those with Poly kernel are 86.2%, 85.8% and 67.7%, respectively. It can be seen that SMILE achieves better classification results than the other MIL methods.

In particular, the performance of SMILE is significantly better than that of EM-DD, in which only one instance from each bag is selected for prediction. For example, the accuracy of EM-DD on Fox dataset is 56.1%, while that of SMILE on Poly kernel is 67.7%. SMILE outperforms EM-DD by 11.6%. This once again confirms that SMILE is very effective in MIL. Rather than selecting only one instance from each bag for prediction, as EM-DD does, considerably involving the ambiguous instances in the learning process can significantly boost MIL accuracy.

E. Text Categorization Dataset

We use the text categorization datasets from [3] to evaluate the performance of SMILE. These datasets are extremely sparse and high-dimensional, which makes them more challenging. The text datasets are randomly sampled from TREC9. The sampled documents are then split into passages using overlapping windows of a maximum of 50 words each. In the MIL setting, a document is considered as a bag and an instance is usually a paragraph or a short passage [3]. Finally, the text categorization dataset consists of 7 subsets, each containing 200 positive and 200 negative bags with about 3330 instances.

The experimental results on text categorization datasets are presented in Table III, from which it can be seen that the accuracy of SMILE on the TSTs from 1 to 10 datasets are 95.7%, 85.3%, 90.5%, 82.5%, 83.5%, 70.9% and 83.4% with RBF kernel. Those with Poly kernel are 96.7%, 83.8%, 91.7%, 87.6%, 84.7%, 72.7% and 85.7%, respectively. After examining the detailed results of other comparable MIL methods, it is impressive to find that SMILE achieves the highest classification accuracy over all 7 datasets. Taking the TST3 dataset as an example, SMILE with the Poly kernel performs much better than EM-DD with around 23% difference in classification accuracy.

Compared to MILBoost, DD-SVM and mi-SVM in which all instances are included to learn the classifier as SMILE does, it is clearly seen that SMILE achieves markedly better classification accuracy than all of them. Let us take the TST2 subset as an example. On the TST2 subset, MILBoost, DD-SVM (with RBF kernel) and mi-SVM (with RBF kernel) obtain the three lowest accuracy at 77.5%, 73.7% and 74.3%, respectively, while SMILE (with RBF kernel) reaches as high as 85.3%, which is around 10% higher than MILBoost, DD-SVM and mi-SVM on average. The better performance of SMILE over MILBoost, DD-SVM and mi-SVM further implies the effectiveness of SMILE in utilizing the ambiguous instances.

F. Sensitivity to Labelling Noise

We investigate the noise sensitivity of SMILE and the baseline methods on 12 benchmark subsets. Following the similar settings in [6], we generate the labelling noise in the datasets. Firstly, we randomly pick up $d\%$ positive bags and $d\%$ negative bags from the training set. Secondly, we change the labels of the selected positive and negative bags, i.e. relabel the positive bags as negative and label the negative bag as positive. Lastly, all the selected bags are put back to the training set. By doing so, the training set has $2 * d\%$ bags with noisy labels (called *noisy bags*).

After the method of generating labelling noise is determined, we follow the operation in [6] to generate the noisy datasets. Specifically, for each of the 12 subsets, 20 noisy datasets are randomly generated under a particular percentage of labelling noise. Then, each noisy dataset is split into training and test sets of equal size. Due to the space limitation, the average classification accuracy of all the noisy datasets in Musk, image retrieval and text categorization datasets is presented, as shown in Figure 2 (a)-(c). The results of mi-SVM, DD-SVM and SMILE are reported under the RBF kernel.

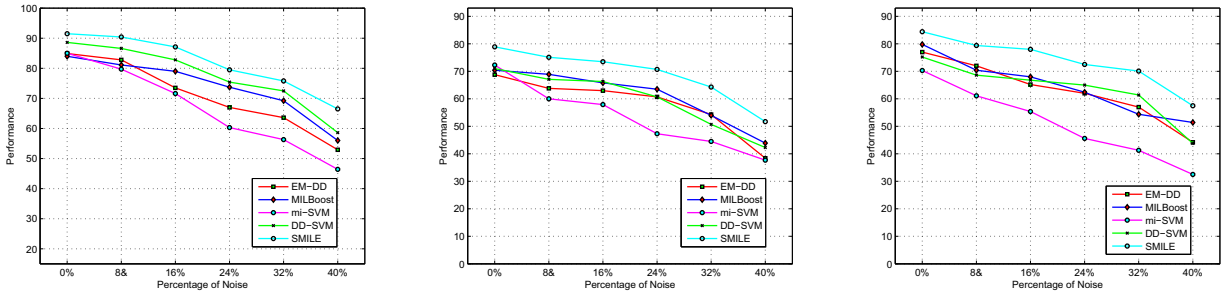
In Figure 2 (a)-(c), it can be seen that SMILE is more robust than all the baseline approaches (EM-DD, MILBoost, mi-SVM and DD-SVM). When the percentage of labelling noise increases from 0% to 40%, SMILE has the lowest decrease of classification accuracy on all three datasets. In contrast to SMILE, mi-SVM seems to be the most sensitive to labelling noise. For example, in Figure 2 (c), the classification accuracy of mi-SVM declines rapidly with the increase of noise level. This may be because mi-SVM focuses on obtaining 100% training accuracy of positive bags. If labelling noise of positive bags exists, the classifier which is learnt for obtaining 100% training accuracy of positive bags, may be severely biased by the noise. Moreover, EM-DD, DD-SVM and MILBoost appear to be more sensitive to the noise than SMILE. This is because EM-DD and DD-SVM involve the computation of DD, while it is pointed out by [6] that DD is "very sensitive to noise".

G. Running Time Analysis

The average training time of EM-DD, MILBoost, mi-SVM, DD-SVM and SMILE on the sub-datasets of Musk,

Table III
AVERAGE ACCURACY ON TEXT CATEGORIZATION DATASETS.

		TST1	TST2	TST3	TST4	TST7	TST9	TST10
EM-DD		85.8	84.0	69.0	80.5	75.4	65.5	78.5
MILBoost		88.3	77.5	85.8	79.5	82.6	66.4	76.8
mi-SVM	RBF	90.4	74.3	69.0	69.6	81.3	55.2	52.6
	Poly	93.6	78.2	87.0	82.8	81.3	67.5	79.6
DD-SVM	RBF	82.6	73.7	82.4	66.3	80.6	62.8	77.2
	Poly	84.8	74.5	87.9	69.4	81.5	66.3	78.6
SMILE	RBF	95.7	**85.3**	90.5	82.5	83.5	70.9	83.4
	Poly	**96.7**	83.8	**91.7**	**87.6**	**84.7**	**72.7**	**85.7**



(a) Musk dataset (b) Image retrieval dataset (c) Text categorization dataset

Figure 2. Sensitivity to labelling noise.

image retrieval and text categorization datasets is presented in Figure 3 (a) to (c). All the experiments are run in a Matlab environment on a laptop with 2.8 GHz processor and 3GB DRAM. All SVM related algorithms are performed by extending LibSVM [25].

From Figure 3 (a) to (c), it is seen that MILBoost is the most efficient method on all the benchmark datasets. For MILBoost, it is based on the Boosting framework and is generally faster than the algorithms training on all data at one time. However, the MIL accuracy of SMILE is significantly better than MILBoost for all 12 benchmark subsets. For example, on the Fox subset, the accuracy of SMILE with Poly kernel is 67.7%, which is much higher than MILBoost (55.2%) at 12.5%. mi-SVM is the second most efficient method. However, it simply focuses on obtaining the 100% training accuracy of positive bags, which may lead to severe accuracy decline if labelling noise of positive bags exists. Moreover, EM-DD and DD-SVM are the least efficient algorithms since they require the computation of DD, which is relatively computationally expensive.

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 by some previous works, the discriminative power of the classifier may be limited. Distinguished from these methods, we explicitly incorporate the ambiguous instances in the construction of classifier, by considering their similarity towards the positive and negative classes. At the same time, SMILE shows better robustness than the DD-based MIL approaches and those works which focus on reducing the false positive rate.

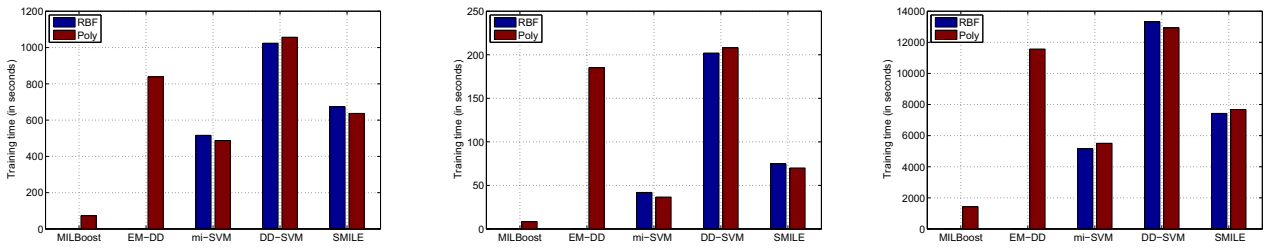
VI. CONCLUSION AND FUTURE WORK

Quite a few existing multiple instance learning methods exclude ambiguous instances in positive bags from the construction of a classifier. In this paper, we have proposed a novel MIL method - SMILE (Similarity-based Multiple Instance LEarning). SMILE explicitly deals with ambiguous instances by assigning different similarity weights towards the positive and negative classes. It further incorporates such ambiguous information into a heuristic classification framework. Experiments on three real-world datasets, consisting of 12 subsets in total, have shown that SMILE achieves markedly better classification accuracy than comparable MIL methods on most of the datasets.

In the future, we plan to extend this work in several directions. Firstly, we would like to develop more efficient SVM learning techniques to further improve training efficiency. Similar to other SVM-based techniques, we need to solve a non-convex optimization problem. Therefore, in this work, we employ an iterative learning framework to obtain the solution. We will look into other possibilities of utilizing more efficient learning techniques to enhance the scalability of our proposed method. Secondly, we will investigate other methods for generating instance similarity weights in different application problems. A desirable method should incorporate some domain knowledge into boosting the performance.

ACKNOWLEDGMENTS

This work is supported by QCIS Center of UTS, Australian Research Council (DP1096218, DP0988016, LP100200774 and LP0989721), the US National Science



(a) Musk dataset (b) Image retrieval dataset (c) Text categorization dataset

Figure 3. Average training time for the subsets in Musk, image retrieval and text categorization datasets (The training time of mi-SVM, DD-SVM and SMILE is reported with RBF and Poly kernel, respectively, while EM-DD and MILBoost are not SVM-based methods).

Foundation (CCF-0905337), the 973 Program of China (2009CB326203), and the National Natural Science Foundation of China (60828005).

REFERENCES

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

[2] S. Ray and M. Craven. Supervised versus multiple instance learning: An empirical comparison. *In Proceeding of ICML*, pages 697–704, 2005.

[3] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. *In Proceeding of NIPS*, 15:561–568, 2003.

[4] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. *In Proceeding of NIPS*, 10:570–576, 1998.

[5] Q. Zhang and S. Goldman. Em-dd: An improved multiple-instance learning technique. *In Proceeding of NIPS*, 15:1073–1080, 2002.

[6] W. Li and D. Yeung. Mild: Multiple-instance learning via disambiguation. *IEEE Transactions on Knowledge and Data Engineering*, 22:76–89, 2010.

[7] D. Wang, J. Li, and B. Zhang. Multiple-instance learning via random walk. *In Proceeding of ECML*, 2006.

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

[9] Y. Chen, J. Bi, and J. Wang. Mile: Multiple instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1931–1947, 2006.

[10] Q. Zhang, S.A. Goldman, W. Yu, and J. Fritts. Content-based image retrieval using multiple-instance learning. *In Proceeding of ICML*, pages 682–689, 2002.

[11] P. Auer. On learning from multi-instance examples: Empirical evaluation of a theoretical approach. *In Proceeding of ICML*, pages 21–29, 1997.

[12] P.M. Long and L. Tan. Pac learning axis aligned rectangles with respect to product distributions from multiple-instance examples. *In Proceeding of CLT*, 1996.

[13] P. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for object detection. *In Proceeding of NIPS*, 18:1417–1424, 2006.

[14] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. *In Proceeding of ICML*, pages 148–156, 1996.

[15] Z. Zhou and J. Xu. On the relation between multi-instance learning and semi-supervised learning. *In Proceeding of ICML*, pages 1167–1174, 2007.

[16] S. Andrews and T. Hofmann. A cutting-plane algorithm for learning from ambiguous examples. *Technical report, Brown University*, 2006.

[17] S. Andrews and T. Hofmann. Disjunctive programming boosting. *In Proceeding of NIPS*, 16, 2004.

[18] X. Xu and E. Frank. Logistic regression and boosting for labeled bags of instances. *In Proceeding of PAKDD*, pages 272–281, 2004.

[19] J. Wang and J.-D. Zucker. Solving the multiple instance problem: A lazy learning approach. *In Proceeding of ICML*, pages 1119–1125, 2000.

[20] V.N. Vapnik. Statistical learning theory. *John Wiley and Sons, New York*, 1998.

[21] J. Bezdek and R. Hathaway. Convergence of alternating optimization. *Neural, Parallel and Scientific Computations*, 11:351–368, 2003.

[22] J. Bi and T. Zhang. Support vector classification with input data uncertainty. *In Proceeding of NIPS*, 11:351–368, 2003.

[23] S.S. Keerthi and S.K. Shevade. Smo algorithm for least squares svm formulations. *Neural Computation*, 15:487–507, 2003.

[24] Y. Zhang, A.C. Surendran, J.C. Platt, and M. Narasimhan. Learning from multi-topic web documents for contextual advertisement. *In Proceeding of SIGKDD*, pages 1051–1059, 2008.

[25] C.C. Chang and C.J. Lin. Libsvm: A library for support vector machines. *Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm*, 2001.