2011 11th IEEE International Conference on Data Mining

Multi-Instance Metric Learning

Ye Xu
Dartmouth College
Computer Science Department
Hanover, NH
ye@cs.dartmouth.edu

Wei Ping
University of California, Irvine
Department of Computer Science
Irvine, CA
wping@ics.uci.edu

Andrew T. Campbell
Dartmouth College
Computer Science Department
Hanover, NH
campbell@cs.dartmouth.edu

***Abstract*—Multi-instance learning, like other machine learning and data mining tasks, requires distance metrics. Although metric learning methods have been studied for many years, metric learners for multi-instance learning remain almost untouched. In this paper, we propose a framework called *Multi-Instance MEtric Learning (MIMEL)* to learn an appropriate distance under the multi-instance setting. The distance metric between two bags is defined using the *Mahalanobis* distance function. The problem is formulated by minimizing the *KL divergence* between two multivariate Gaussians under the constraints of maximizing the between-class bag distance and minimizing the within-class bag distance. To exploit the mechanism of how instances determine bag labels in multi-instance learning, we design a nonparametric density-estimation-based weighting scheme to assign higher "weights" to the instances that are more likely to be positive in positive bags. The weighting scheme itself has a small workload, which adds little extra computing costs to the proposed framework. Moreover, to further boost the classification accuracy, a kernel version of MIMEL is presented. We evaluate MIMEL, using not only several typical multi-instance tasks, but also two activity recognition datasets. The experimental results demonstrate that MIMEL achieves better classification accuracy than many state-of-the-art distance based algorithms or kernel methods for multi-instance learning.**

***Keywords*-Multi-instance learning; Metric learning; Weighting scheme**

I. Introduction

Multi-instance learning originated from the drug activity prediction problem [11]. In multi-instance learning, training samples are bags that contain many instances. A bag is positive if it contains at least one positive instance; otherwise it is labeled as negative. We know the labels of bags in the training set. However, the exact labels of instances in the bags are unknown. During the past few years, the multi-instance learning framework has attracted much attention in a variety of domains such as object detection [32], image classification [23], [6], [7], visual tracking [2], information retrieval [38], [40], [28], and biomedical informatics [13].

Metric learning [37] is an important tool in machine learning and data mining. It aims at finding an appropriate distance metric over an input space for a given problem. Previous studies [39] indicate that an appropriate distance metric can benefit the accuracy of distance based classifiers compared with using the standard Euclidean distance. Because we only know the labels of bags but cannot access the labels of instances, existing supervised or semi-supervised metric learning methods [27], [10], [3] cannot be trivially extended to solve multi-instance tasks. On the other hand, those unsupervised metric learners [8], [16] that ignore bag label information are not suitable to address this issue, either. Although multi-instance learning problems still involve the corresponding concept of distance [34], to the best of our knowledge, learning a distance metric under the multi-instance setting remains almost untouched.

In this paper, we propose a framework called *Multi-Instance MEtric Learning (MIMEL)* to learn a distance metric in the multi-instance setting. We define the distance metric between two bags by leveraging the instance level *Mahalanobis distance*, which generalizes the standard Euclidean distance via linear scalings and rotations in the input space. The multivariate Gaussian is applied to model the Mahalanobis matrix using a bijection. The *KL Divergence* is used as the regularization term, under the constraints of both maximizing the between-class bag distance and minimizing the within-class bag distance. The *cyclic projection technique* is employed to solve the optimization problem. Another critical concern is the mechanism of the multi-instance learning. Under the multi-instance setting, the bag label is determined by the existence of the positive instances, and thus positive instances have more discriminative abilities [1]. As illustrated in the Fig.1, positive instances in the positive bags obviously play a more important role in distinguishing between positive bags and negative bags. If all the instances in the positive bags are treated equally, the significance of those "key" instances is likely to be undermined. To detect the "key" instances and assign higher weights to them in computing the distance between two bags, a nonparametric scheme is proposed to estimate the probabilistic label of each instance in the positive bags. To achieve a small computing load for this estimation procedure, *Taylor Series* is employed to approximately compute all the probabilistic labels, and an error bound is given.

1550-4786/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDM.2011.106

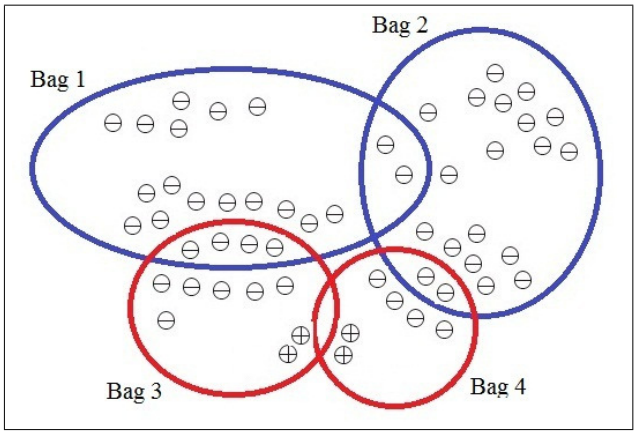

Figure 1. Instances in each bag have different discriminative powers. The top two blue bags are negative bags, and the bottom two red ones are positive bags. The symbol $\oplus$ and $\ominus$ respectively denote positive instances and negative instances. The existence of positive instances implies a close distance between the two positive bags. However, if every instance in a bag is treated with equal weight, some positive bags (e.g., bag 3) and negative bags (e.g., bag 1) are considered to be closer.

In summary, the contributions of our work are as follows:

- We propose a *multi-instance metric learning* method, which learns an appropriate distance function in the multi-instance setting. We also *kernelize* the proposed metric learning method to further boost the classification performance.
- We design a nonparametric *weighting scheme* to assign higher weights to instances that are more likely to be positive in positive bags since positive instances play a more significant role in the multi-instance setting. The designed weighting scheme not only highlights the "key" instances in positive bags, but also exploits the relation between the bag labels and the instance labels, which captures the intrinsic mechanism of multi-instance learning.
- We use two multi-instance learning datasets from *real-world activity recognition tasks* using the GPS sensor. To the best of our knowledge, they are the first activity recognition dataset in the multi-instance setting. The experiments demonstrate that the proposed MIMEL can achieve better performance on not only under classic multi-instance scenarios such as bioinformatics (e.g., Musk1, Musk2) and image annotation problems (e.g., Elephant, Fox, Tiger), but also activity recognition tasks.

The rest of this paper is organized as follows. In section 2, we discuss related work. The detailed MIMEL framework is detailed in section 3. In section 4, we report on experimental results. In Section 5, we present the conclusions.

II. Related Work

Multi-instance learning has been a hot topic during the past decade. Many algorithms have been investigated, such as Diverse Density (DD) [24], Citation-kNNC [34], MI Kernel [14], EM-DD [41], MI SVMs [1], MI Instance Selection [7], mi-Graph Kernel and MI-Graph Kernel [43], and Famer [26].

Metric learning[1], which tries to find a distance to approximate a task-specific similarity, is an effective tool in machine learning and data mining. Various metric learning algorithms have been proposed in the past few years. Depending on whether the label information is needed, these metric learners can be categorized into two classes: supervised metric learning and unsupervised metric learning methods. These supervised metric learners such as [37], [27], [10] cast label information of data into pairwise constraints: the between-class constraints and the within-class constraints. Based on different loss functions, the within-class distance is to be minimized whereas the between-class distance is to be maximized. However, these supervised metric learning algorithms need to avail the label of each instance in the training set, which is impossible to apply in the multi-instance framework. Unsupervised metric learning methods [39], also called manifold learning methods, focus on learning an underlying low-dimensional manifold to preserve the geometric relations between the training data. Employing those typical unsupervised metric learning methods such as Multiple Dimension Scaling (MDS) [8], and Locality Preserving Projections (LPP) [16] to learn a distance for multi-instance tasks would miss the bag label information. Therefore, designing a metric learning algorithm under the multi-instance setting is desirable.

One relevant work is [15], which aims at learning a metric under a different mechanism of how instance labels determines bag labels from the original multi-instance setting introduced in [11]. In [15], the label space is on pairs of bags: if two bags share at least one same label, the pairs of bags is labeled positive, otherwise it is negative. This setting naturally fits the scenario of face recognition. However, our work focuses on the original multi-instance learning setting.

Our work is closely related to metric learners on sets of data [36], [17]. (Although entitled "learning a distance metric from multi-instance multi-label data", [17] is essentially a study of metric learning on sets of data because it ignores the essential mechanism of how instances determine bag labels

[1]Note that although some dimensionality reduction algorithms for multi-instance learning [25], [30] can be deduced as metric learners, they are proposed to address the "curse of dimensionality" problem, and thus often have different methodologies from metric learning. Dimensionality reduction methods aim to preserve discriminative information for classifiers under low dimensionality, whereas multi-instance metric learning algorithms focus on capturing a reasonable distance between bags. In this regard, dimensionality reduction is different from multi-instance metric learning, and is out of the scope of this paper.

in the multi-instance setting. Furthermore, [17] also considers the label space ambiguity, which is out of the scope of this paper.) In the set learning framework, learning examples are sets (bags) that contain many data (instances), and the set metric learner is designed to evaluate the distance between two sets of data. The framework of set metric learning naturally corresponds to the scenario of the *bag of features*, and has various applications, for instance, image annotation [17]. However, there is a fundamental difference between set learning and multi-instance learning. In the framework of set learning, all the data (instances) within a set (bag) jointly make up a complete description and the *label space* is on the set, whereas under the multi-instance learning, the label space is on the instance within the bag. In other words, in multi-instance learning, not only bags but also instances themselves within the bags relate to the physical meaning of the labels, which distinguishes multi-instance learning from the set learning framework. Therefore, it is desirable to capture the discriminative information conveyed by the available instance labels under the multi-instance setting. In our work, the designed weighting scheme tries to use negative instances to estimate the probabilistic labels of instances within the positive bags, which successfully exploits the mechanism of how instances determine bag labels.

Activity recognition [12] is an important task in machine learning and data mining due to its wide application domains such as medical diagnosis [22], human behavior modeling [20] and public facility analysis [42]. Many supervised and semi-supervised learning methods have been employed to address activity recognition tasks. Recently, Stikic et al. [29] first applied multi-instance learning to cope with activity recognition tasks and achieved good performance. However, the datasets used in [29] were collected for traditional supervised activity recognition (i.e., the data were labeled at the instance level). As far as we know, there are few specifically collected activity recognition datasets for multi-instance learning.

III. Framework MIMEL

In this section, we propose the MIMEL framework to learn a metric distance under the multi-instance setting. To capture the essential mechanism of how instances determine bag labels in multi-instance learning, we design a weighting scheme to highlight those "key" instances in positive bags. At last, we show how to kernelize the original MIMEL to further improve the classification performance.

A. Problem Description

Before presenting the multi-instance metric learning problem in detail, we give the formal description of multi-instance learning as follows. Let $\mathcal{X}$ denote the instance space. Given a data set $T = \{(\mathbf{X_1}, L_1), ..., (\mathbf{X_i}, L_i), ..., (\mathbf{X_N}, L_N)\}$, where $\mathbf{X_i} = \{\boldsymbol{x}_{i1}, ..., \boldsymbol{x}_{ij}, ..., \boldsymbol{x}_{i,n_i}\} \subset \mathcal{X}$ is called a bag, $L_i \in \mathcal{L} = \{-1, +1\}$ is the label of $\mathbf{X_i}$ and N is the number of training bags, the goal is to generate a learner to classify unseen bags. Here $x_{ij} \in \mathbf{X}$ is an instance $[\boldsymbol{x}_{ij1}, ..., \boldsymbol{x}_{ijk}, ..., \boldsymbol{x}_{ijD}]^\top$, where x_{ijk} is the value of $\boldsymbol{x}_{ij}$ at the k^{th} attribute, n_i is the number of instances in $\mathbf{X_i}$, and D is the dimensionality of the original space $\mathcal{X}$. If there exists $p \in \{1, ..., n_i\}$ such that $\boldsymbol{x}_{ip}$ is a positive instance, then $\mathbf{X_i}$ is a positive bag and thus $L_i = +1$, but the exact value of the index p is usually unknown; otherwise $L_i = -1$.

Under the multi-instance setting, we seek a metric between two bags so that different classes of bags stay as distant as possible, whereas the same class of bags stays as close as possible. Before formally introducing the criterion of this goal, we need to define a distance metric at the bag level. A reasonable measure is:

$$Dis(\mathbf{X_i}, \mathbf{X_j}) = \frac{1}{n_i n_j} \sum_{a=1}^{n_i} \sum_{b=1}^{n_j} (\boldsymbol{x_{ia}} - \boldsymbol{x_{jb}})^\top (\boldsymbol{x_{ia}} - \boldsymbol{x_{jb}}) \quad (1)$$

This definition means that the distance between bags is measured by the average distance of pairwise instances from different bags. Similar pairwise metric methodology is employed in [14] to setup kernels between multi-instance bags. This pairwise metric methodology has been proven to be effective to measure the distance among bags [43], [25].

Now the problem of metric learning under the multi-instance setting can be formally described as follows: given a data set $T = \{(\mathbf{X_1}, L_1), ..., (\mathbf{X_i}, L_i), ..., (\mathbf{X_N}, L_N)\}$ as above, we aim at learning a positive definite matrix $\mathbf{A}$ that parameterizes the Mahalanobis distance,

$$\begin{aligned} Dis_{\mathbf{A}}(\mathbf{X_i}, \mathbf{X_j}) &= \frac{1}{n_i n_j} \sum_{a=1}^{n_i} \sum_{b=1}^{n_j} Dis_{\mathbf{A}}(\boldsymbol{x_{ia}}, \boldsymbol{x_{jb}}) \\ &= \frac{1}{n_i n_j} \sum_{a=1}^{n_i} \sum_{b=1}^{n_j} (\boldsymbol{x_{ia}} - \boldsymbol{x_{jb}})^\top \mathbf{A} (\boldsymbol{x_{ia}} - \boldsymbol{x_{jb}}) \end{aligned} \quad (2)$$

Under such a metric, the distance between two bags that share the same label is smaller than a given upper bound, i.e., $Dis_{\mathbf{A}}(\mathbf{X_i}, \mathbf{X_j}) \leq u$ for a relatively small value of u. Similarly, the distance between two bags with different labels is larger than a given lower bound, i.e., $Dis_{\mathbf{A}}(\mathbf{X_i}, \mathbf{X_j}) \geq l$ for a large value of l.

Given the bag distance constraints, the problem is to learn a positive definite matrix A that parameterizes the instance level Mahalanobis distance in (2). Under certain conditions, prior information about the Mahalanobis distance function is known. Therefore, we regularize the Mahalanobis matrix A to be close to a given Mahalanobis distance function, parameterized by $\mathbf{A_0}$.[2]

[2]When no prior knowledge is available, setting up $\mathbf{A_0}$ as the identity matrix empirically works well [18].

Next, the KL divergence is employed to measure the "closeness" between $\mathbf{A}$ and $\mathbf{A_0}$. Specifically, we model a multivariate Gaussian for a parameterized matrix $\mathbf{A}$ by the following bijection:

$$p(\boldsymbol{x}; \mathbf{A}) = \frac{1}{C} exp(-0.5(\boldsymbol{x} - \mu)^\top \mathbf{A}(\boldsymbol{x} - \mu)) \tag{3}$$

Therein, C is a normalization constant, μ is the mean of the Gaussian, and $\mathbf{A^{-1}}$ is the covariance of the distribution. Then the "closeness" between two Mahalanobis functions parameterized by $\mathbf{A}$ and $\mathbf{A_0}$ can be evaluated as follows:

$$\mathbf{KL}(p(\boldsymbol{x}; \mathbf{A_0})||p(\boldsymbol{x}; \mathbf{A})) = \int p(\boldsymbol{x}; \mathbf{A_0}) log \frac{p(\boldsymbol{x}; \mathbf{A_0})}{p(\boldsymbol{x}; \mathbf{A})} d\boldsymbol{x} \tag{4}$$

Given the above measure, the multi-instance distance metric learning problem can be formulated as follows,

$$\begin{aligned} & min_{\mathbf{A}} \mathbf{KL}(p(\boldsymbol{x}; \mathbf{A_0})||p(\boldsymbol{x}; \mathbf{A})) \\ \text{subject to} \quad & Dis_{\mathbf{A}}(\mathbf{X_i}, \mathbf{X_j}) \leq u, L_i = L_j; \\ & Dis_{\mathbf{A}}(\mathbf{X_i}, \mathbf{X_j}) \geq l, L_i \neq L_j \end{aligned} \tag{5}$$

B. Optimization

What follows, we describe the procedure to address the optimization problem (5). We rewrite the objective formula in (5) into a type of *Bregman divergence* [4]. We use the Bregman's method to solve the problem.

Assuming the means μ of the Gaussians in (3) to be a constant, [9] indicates that the objective function can be rewritten as follows:

$$\mathbf{KL}(p(\boldsymbol{x}; \mathbf{A_0})||p(\boldsymbol{x}; \mathbf{A})) = \frac{1}{2} D_{ld}(\mathbf{A}, \mathbf{A_0})$$

Therein, $D_{ld}(\mathbf{A}, \mathbf{A_0})$ is the *LogDet divergence* [4] that belongs to one type of Bregman divergence,

$$D_{ld}(\mathbf{A}, \mathbf{A_0}) = tr(\mathbf{A}\mathbf{A_0^{-1}}) - \log \det(\mathbf{A}\mathbf{A_0^{-1}}) - n, \tag{6}$$

where n is the order of matrices $\mathbf{A}$ and $\mathbf{A_0}$.

For the purpose of representation simplicity, we denote

$$\mathbf{M_{ij}} = \frac{1}{n_i n_j} \sum_{a=1}^{n_i} \sum_{b=1}^{n_j} (\boldsymbol{x_{ia}} - \boldsymbol{x_{jb}})(\boldsymbol{x_{ia}} - \boldsymbol{x_{jb}})^\top.$$

Therefore, (5) can be written in the following convex form:

$$\begin{aligned} & min_{\mathbf{A}} D_{ld}(\mathbf{A}, \mathbf{A_0}) \\ \text{subject to} \quad & tr(\mathbf{A}\mathbf{M_{ij}}) \leq u, \text{for } L_i = L_j; \\ & tr(\mathbf{A}\mathbf{M_{ij}}) \geq l, \text{for } L_i \neq L_j. \end{aligned} \tag{7}$$

This optimization problem can be solved using the *cyclic projections method* [4]. Below, we briefly introduce the basic idea of the method. Cyclic projection algorithm uses an iteration scheme to minimize the objective functions under certain constraints. In each iteration, one constraint is chosen. If an equality constraint is chosen, we project the current solution $\mathbf{A_t}$ onto the selected constraint to achieve $\mathbf{A_{t+1}}$ via solving the following equations for α and $\mathbf{A_{t+1}}$:

$$\begin{aligned} \nabla D_{ld}(\mathbf{A_{t+1}}, \mathbf{A_0}) &= \nabla D_{ld}(\mathbf{A_t}, \mathbf{A_0}) + \alpha \mathbf{M_{ij}} \\ tr(\mathbf{A_{t+1}}\mathbf{M_{ij}}) &= p \end{aligned} \tag{8}$$

Therein, $p = 1$ if $L_i = L_j$; $p = -1$ if $L_i \neq L_j$.

If the selected constraint is an inequality, an non-negative *dual variable* λ_{ij} is maintained. After obtaining the solutions of (8), we set $\hat{\alpha} = min(\lambda_{ij}, \alpha)$, and update λ_{ij} by

$$\lambda_{ij} = \lambda_{ij} - \hat{\alpha}.$$

Then $\mathbf{A_{t+1}}$ is updated by:

$$\nabla D_{ld}(\mathbf{A_{t+1}}, \mathbf{A_0}) = \nabla D_{ld}(\mathbf{A_t}, \mathbf{A_0}) + \hat{\alpha} \mathbf{M_{ij}} \tag{9}$$

The key procedure in cyclic projections is to solve the above system of equations. For simplicity, an assumption is made that the rank of $\mathbf{M_{ij}}$ equals one. Hence $\mathbf{M_{ij}}$ can be decomposed as,

$$\mathbf{M_{ij}} = \boldsymbol{z_{ij}} \boldsymbol{z_{ij}^\top} \tag{10}$$

By computing the gradient of objective function (6), the solution of (7) can be achieved using the following iteration:

$$\mathbf{A_{t+1}} = (\mathbf{A_t}^+ - \alpha \mathbf{M_{ij}})^+ \tag{11}$$

Therein, $\mathbf{A_t}^+$ is the pseudoinverse of the positive definite matrix $\mathbf{A_t}$. The detailed algorithm of the cyclic projections can be found in [4].

Then, according to the following *Sherman-Morrison inverse formula*, we get:

$$(\mathbf{A} + \boldsymbol{u}\boldsymbol{v}^\top)^+ = \mathbf{A_t}^+ - \frac{\mathbf{A_t}^+ \boldsymbol{u}\boldsymbol{v}^\top \mathbf{A_t}^+}{\boldsymbol{v}^\top \mathbf{A_t}^+ \boldsymbol{u}} \tag{12}$$

so that (11) can be rewritten as follows:

$$\mathbf{A_{t+1}} = \mathbf{A_t} + \frac{\alpha \mathbf{A_t} \mathbf{M_{ij}} \mathbf{A_t}}{1 - \alpha \boldsymbol{z_i^\top} \mathbf{A_t} \boldsymbol{z_i}} \tag{13}$$

Because $\mathbf{A_{t+1}}$ must satisfy the constraint between bag i and bag j, i.e., $tr(\mathbf{A_{t+1}} \boldsymbol{z_{ij}} \boldsymbol{z_{ij}^\top}) = p$, we can obtain the closed form of α by solving the following equation:

$$tr((\mathbf{A_t} + \frac{\alpha \mathbf{A_t} \mathbf{M_{ij}} \mathbf{A_t}}{1 - \alpha \boldsymbol{z_i^\top} \mathbf{A_t} \boldsymbol{z_i}}) \boldsymbol{z_{ij}} \boldsymbol{z_{ij}^\top}) = p \tag{14}$$

After obtaining

$$\alpha = \frac{1}{tr(\mathbf{A_t} \boldsymbol{z_{ij}} \boldsymbol{z_{ij}^\top})} - \frac{1}{\xi}, \tag{15}$$

we can achieve the update formula by taking α into equation (13).

To cope with the possibility that no feasible solution exists, we employ the *slack variables technique*. Specifically, let ξ be the vector of slack variables, and $c(i, j)$ denote

the index of the constraint between bag i and bag j. The optimization problem in (7) is then rewritten as follows:

$$\begin{aligned} & min_{\mathbf{A}} D_{ld}(\mathbf{A}, \mathbf{A_0}) + \gamma D_{ld}(\xi, \xi_0) \\ \text{subject to} \quad & tr(\mathbf{AM_{ij}}) \leq \xi_{c(i,j)}, \text{for } L_i = L_j; \\ & tr(\mathbf{AM_{ij}}) \geq \xi_{c(i,j)}, \text{for } L_i \neq L_j. \end{aligned} \tag{16}$$

Therein, γ is the slack factor which controls the tradeoff between satisfying the constraints and minimizing the divergence between $\mathbf{A}$ and $\mathbf{A_0}$. The component of ξ_0 is initialized as u for the same label constraints and l for the different label constraints. The solution of (16) is quite similar to (7). We give the detailed procedure in Algorithm 1. A single iteration has the time complexity $O(n^2)$, where n is the average number of instances in each bag. [9], [10] indicate that the solution is not sensitive to the parameters and the constraint pairs chosen in each iteration.

Input: Data set $\{(\mathbf{X_1}, L_1), ..., (\mathbf{X_N}, L_N)\}$, input Mahalanobis matrix $\mathbf{A_0}$, slack variable γ, and distance threshold u and l.

1: Initialize $\xi_{c(i,j)} = u$ if $L_i = L_j$; otherwise l.
2: Initialize $t = 0$, and $\lambda_{i,j} = 0$, $\forall i, j$.
3: **Repeat** step 4 to step 10 several times.
4: Randomly pick up two bags $\mathbf{X_i}$ and $\mathbf{X_j}$.
5: Compute $p = \frac{1}{n_i n_j} \sum_{a=1}^{n_i} \sum_{b=1}^{n_j} (\boldsymbol{x_{ia}} - \boldsymbol{x_{jb}})^\top \mathbf{A_t} (\boldsymbol{x_{ia}} - \boldsymbol{x_{jb}})$.
6: Set up $\delta = 1$ if $L_i = L_j$; $\delta = -1$ otherwise.
7: Compute $\alpha = \min(\lambda_{i,j}, \frac{\delta}{2} - (\frac{1}{p} - \frac{\gamma}{\xi_{c(i,j)}}))$.
8: Update $\xi_{c(i,j)} \leftarrow \frac{\gamma \xi_{c(i,j)}}{\gamma + \delta\alpha\xi_{c(i,j)}}$.
9: Update $\lambda_{i,j} \leftarrow \lambda_{i,j} - \alpha$.
10: Update $\mathbf{A_{t+1}} \leftarrow \mathbf{A_t} + \frac{\delta\alpha}{1-\delta\alpha p} \mathbf{A_t M_{ij} A_t}$.

Output: The Mahalanobis Matrix $\mathbf{A}^*$.

Algorithm 1: Multi-Instance Metric Learning

C. Weighting Scheme

According to the mechanism of multi-instance learning, the bag label is determined by the existence of the positive instances. Therefore, positive instances have more discriminative abilities and are more critical [1]. As illustrated in the Fig.1, if all instances are treated equally for multi-instance learning, bag 3 and bag 1 are thought to have smaller distance than bag 3 and bag 4 because bag 3 and bag 1 have more overlapping instances. Thus, it is desirable to assign higher weights to the probably positive instances so that the discriminative abilities of positive instances can be taken advantage of in evaluating the bag distance.

Some studies try to assign different weights to instances within a bag based on their discriminative abilities. For example, [7] employs the Lasso process [31], where $\mathcal{L}_1$ Norm SVM is usually used, to learn the weights for instances. The DD scheme [24], which aims at estimating the probabilistic labels of instances, is applied by [19] to assign weights to instances. However, most of these weighting schemes suffer from heavy workloads. To address this problem, we design an efficient weighting scheme based on a *nonparametric* probability density estimation method.

Under the multi-instance setting, we know that all instances in negative bags are negative. However, as for positive bags, we don't know which instances are positive and which are negative. In our work, first we model the probability distribution of negative instances based on all the data in negative bags. Then the learned model is used to estimate the density of each instance in positive bags. The instances in positive bags which fit the distribution model well, i.e., have a high density value, are more likely to be negative instances. On the contrary, the instances that have a low density value are more likely to be positive instances and should be highlighted. Parametric distribution models such as the Gaussian Model have been widely used in modeling the distribution of data. However, it may not work well when the data are multimodal. Gaussian Mixed Model (GMM) is an effective way to cope with the multimodal problem. Unfortunately, how to predetermine the number of the mixture components without prior knowledge in GMM is difficult. Therefore, we apply the *nonparametric kernel estimator* to model the distributions with all negative instances in negative bags.

Formally, we model the probability distribution of negative instances with the following equation:

$$f(\boldsymbol{x}) = \frac{1}{h^D N^-} \sum_{i=1}^{N^-} K(\frac{\boldsymbol{x} - \boldsymbol{x_i^-}}{h}) \tag{17}$$

Therein, $\boldsymbol{x_i^-} \in \mathcal{R}^{\mathcal{D}}, i = 1, 2, ..., N^-$ are all negative instances in negative bags, and h is the bandwidth parameter. The Gaussian Kernel $K(\boldsymbol{x}) = \frac{1}{\sqrt{2\pi}} exp(-\frac{\|\boldsymbol{x}\|^2}{2})$ is commonly used for nonparametric kernel estimators. For every instance within a positive bag, we apply (17) to infer the probability that the instance is negative. For each positive bag, we achieve the weight vector as below,

$$w(\mathbf{X_i}) = [1 - f_{nor}(\boldsymbol{x_{i1}}), ..., 1 - f_{nor}(\boldsymbol{x_{i,n_i}})] \tag{18}$$

Here, $f_{nor}(\boldsymbol{x_{ij}})$ is the normalized value (i.e., between 0 and 1) of the probability density $f(\boldsymbol{x_{ij}})$ for the purpose of convenient computation. In this way, the probable positive instances in positive bags are assigned higher weights.

A disadvantage of using (17) to estimate the probability is that the time complexity for inferring each instance in positive bags is linear with the total number of negative instances. Thus, the complexity of the whole weighting scheme would be quadratic with the instance number in all training bags, which adds heavy computing burden on the proposed framework. To cope with this problem, we propose an efficient way to compute $f(\boldsymbol{x})$.

Note that (17) can be rewritten as follows:

$$f(\boldsymbol{x}) = \frac{1}{h^D N^-} \cdot \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{N^-} exp(-\frac{\|\boldsymbol{x}-\boldsymbol{x_i^-}\|^2}{2h^2})$$
$$= \frac{1}{\sqrt{2\pi}h^D N^-} \sum_{i=1}^{N^-} exp(-\sum_{j=1}^{D} \frac{(x_j - x_{ij}^-)^2}{2h^2})$$
$$= \frac{1}{\sqrt{2\pi}h^D N^-} exp(-\frac{\sum_{j=1}^{D} {x_j}^2}{2h^2})$$
$$\cdot \sum_{i=1}^{N^-} exp(-\frac{\sum_{j=1}^{D} {x_{ij}^-}^2}{2h^2}) exp(\frac{\sum_{j=1}^{D} x_j x_{ij}^-}{h^2}) \quad (19)$$

For the term $exp(\frac{\sum_{j=1}^{D} x_j x_{ij}^-}{h^2})$, we apply *Taylor Series* to achieve the following approximation of (19):

$$\hat{f}(\boldsymbol{x}) = \frac{1}{\sqrt{2\pi}h^D N^-} exp(-\frac{\sum_{j=1}^{D} {x_j}^2}{2h^2})$$
$$\cdot \sum_{i=1}^{N^-} exp(-\frac{\sum_{j=1}^{D} {x_{ij}^-}^2}{2h^2})(1 + \frac{1}{h^2} \sum_{j=1}^{D} x_j x_{ij}^-)$$
$$= \frac{1}{\sqrt{2\pi}h^D N^-} exp(-\frac{\|\boldsymbol{x}\|^2}{2h^2}) \sum_{i=1}^{N^-} exp(-\frac{\|\boldsymbol{x_i^-}\|^2}{2h^2}) +$$
$$\frac{exp(-\frac{\|\boldsymbol{x}\|^2}{2h^2})}{\sqrt{2\pi}h^{(D+2)} N^-} \sum_{j=1}^{D} x_j (\sum_{i=1}^{N^-} x_{ij}^- exp(-\frac{\|\boldsymbol{x_i^-}\|^2}{2h^2}))$$

Therefore, when using the above formula to estimate the probability density value of instances in positive bags, we can first scan all the negative bags once, and compute the values of the items $\sum_{i=1}^{N^-} exp(-\frac{\|\boldsymbol{x_i^-}\|^2}{2h^2})$ and $\sum_{i=1}^{N^-} x_{ij}^- exp(-\frac{\|\boldsymbol{x_i^-}\|^2}{2h^2})$. During the density estimation stage, we use the stored values of the items. Thus, the time complexity for inferring each instance is $O(1)$, and the whole complexity of the weighting scheme is linear with the total number of instances in all bags, which adds little extra computing burden to the MIMEL framework.

The following Lemma indicates the error bound between $f(\boldsymbol{x})$ and $\hat{f}(\boldsymbol{x})$.

Lemma 1: $|\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \frac{2}{\sqrt{2\pi}e^2 h^D}$.

Proof: According to the Lagrange remainder of Taylor Series, we obtain the following inequality:

$$|\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \frac{1}{\sqrt{2\pi}h^D N^-} exp(-\frac{\sum_{j=1}^{D} {x_j}^2}{2h^2})$$
$$\cdot \sum_{i=1}^{N^-} \frac{(\sum_{j=1}^{D} x_j x_{ij}^-)^2}{2h^4} exp(-\frac{\sum_{j=1}^{D} {x_{ij}^-}^2}{2h^2})$$
$$= \frac{1}{\sqrt{2\pi}h^D N^-} \sum_{i=1}^{N^-} \frac{\|\boldsymbol{x_i^-}^\top \boldsymbol{x}\|^2}{2h^4} exp(-\frac{\|\boldsymbol{x_i^-}\|^2 + \|\boldsymbol{x}\|^2}{2h^2})$$
$$\leq \frac{1}{\sqrt{2\pi}2h^D N^-} \sum_{i=1}^{N^-} \frac{(\|\boldsymbol{x_i^-}\|^2 + \|\boldsymbol{x}\|^2)^2}{4h^4} exp(-\frac{\|\boldsymbol{x_i^-}\|^2 + \|\boldsymbol{x}\|^2}{2h^2})$$

Because function $g(y) = y^2 exp(-y)$ achieves the maximum at $y = 2$ on $[0, +\infty)$, we obtain the following result,

$$|\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \frac{2}{\sqrt{2\pi}e^2 h^D} \quad (20)$$

which completes the proof. **END**

In all our experiments, the bandwidth h is larger than 1. Therefore, we can conclude that the item in the right of inequality sign of (20) is small, and $\hat{f}(\boldsymbol{x})$ can be used to approximate $f(\boldsymbol{x})$ at the risk of a small error.

Let $\hat{w}(\mathbf{X_i})$ be the normalized weight vector learned by (18), the sum of whose components equals one. Then, we use the following equation instead of (2) to represent the distance between two bags:

$$Dis_{\mathbf{A}}(\mathbf{X_i}, \mathbf{X_j}) =$$
$$\sum_{a=1}^{n_i} \sum_{b=1}^{n_j} \hat{w}(\boldsymbol{x_{ia}}) \hat{w}(\boldsymbol{x_{jb}}) (\boldsymbol{x_{ia}} - \boldsymbol{x_{jb}})^\top \mathbf{A} (\boldsymbol{x_{ia}} - \boldsymbol{x_{jb}})$$

In this way, the pairs of positive instances between two bags are highlighted when the distance between bags are computed.

By the weighting scheme, the negative instances are exploited to infer the probabilistic labels of instances in positive bags in the proposed MIMEL framework. In other words, the mechanism of how the instance labels determine bag labels in multi-instance learning is made use of, which distinguishes MIMEL from set metric learners.

D. Kernelizing the Method

Linear method could work well in linear separable problems. However, for nonlinear tasks, the performance might be affected. In what follows, we show how to kernelize the MIMEL framework to address this issue.

In the kernel version of the framework, each instance $\boldsymbol{x} \in \mathcal{X}$ is transformed into a $\mathcal{H}$-dimensional kernel space via a nonlinear map ϕ. The original problem (16) in kernel space can be formalized in this way as follows:

$$min_{\mathbf{A}} D_{ld}(\mathbf{A}, \mathbf{A_0}) + \gamma D_{ld}(\xi, \xi_0)$$
$$\text{subject to}$$
$$\sum_{a=1}^{n_i} \sum_{b=1}^{n_j} \hat{w}(\boldsymbol{x_{ia}}) \hat{w}(\boldsymbol{x_{jb}}) Dis_{\mathbf{A}}(\phi(\boldsymbol{x_{ia}}), \phi(\boldsymbol{x_{jb}}))$$
$$\leq \xi_{c(i,j)}, \text{for } L_i = L_j;$$
$$\sum_{a=1}^{n_i} \sum_{b=1}^{n_j} \hat{w}(\boldsymbol{x_{ia}}) \hat{w}(\boldsymbol{x_{jb}}) Dis_{\mathbf{A}}(\phi(\boldsymbol{x_{ia}}), \phi(\boldsymbol{x_{jb}}))$$
$$\geq \xi_{c(i,j)}, \text{for } L_i \neq L_j. \quad (21)$$

Therein, $\mathbf{A}, \mathbf{A_0} \in \mathcal{R}^{\mathcal{H}^2}$ are operators in the kernel space. Here we assume that the weights of instances within each bag remain constant after the kernel mapping.

Thus, the key issue to kernelize MIMEL is to compute the distance between two instances that are mapped in the kernel space,

$$Dis_{\mathbf{A}}(\phi(\boldsymbol{x}), \phi(\boldsymbol{y})) = (\phi(\boldsymbol{x}) - \phi(\boldsymbol{y}))^\top \mathbf{A} (\phi(\boldsymbol{x}) - \phi(\boldsymbol{y}))$$
$$= \phi(\boldsymbol{x})^\top \mathbf{A} \phi(\boldsymbol{x}) + \phi(\boldsymbol{y})^\top \mathbf{A} \phi(\boldsymbol{y}) - 2\phi(\boldsymbol{x})^\top \mathbf{A} \phi(\boldsymbol{y})$$

To address this problem, we define another kernel function

$$\hat{K}(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^\top \mathbf{A} \phi(\boldsymbol{y}). \quad (22)$$

In this way, the problem (21) can be rewritten as follows,

$$\begin{gathered} min_{\hat{\mathbf{K}}} D_{ld}(\hat{\mathbf{K}}, \hat{\mathbf{K}}_0) + \gamma D_{ld}(\xi, \xi_{\mathbf{0}}) \\ \text{subject to} \\ \sum_{a=1}^{n_i}\sum_{b=1}^{n_j} \hat{w}(\boldsymbol{x_{ia}})\hat{w}(\boldsymbol{x_{jb}})(\hat{\mathbf{K}}_{ia,ia} + \hat{\mathbf{K}}_{jb,jb} - 2\hat{\mathbf{K}}_{ia,jb}) \\ \leq \xi_{c(i,j)}, \text{for } L_i = L_j; \\ \sum_{a=1}^{n_i}\sum_{b=1}^{n_j} \hat{w}(\boldsymbol{x_{ia}})\hat{w}(\boldsymbol{x_{jb}})(\hat{\mathbf{K}}_{ia,ia} + \hat{\mathbf{K}}_{jb,jb} - 2\hat{\mathbf{K}}_{ia,jb}) \\ \geq \xi_{c(i,j)}, \text{for } L_i \neq L_j. \end{gathered} \tag{23}$$

Therein,

$$\hat{\mathbf{K}}_{ia,jb} = \hat{\mathbf{K}}(\boldsymbol{x_{ia}}, \boldsymbol{x_{jb}}).$$

$\hat{\mathbf{K}}$ and $\hat{\mathbf{K}}_0$ are T-order kernel matrices, where $T = \sum_{i=1}^{N} n_i$ is the total number of instances in all training bags.

According to (22), we can achieve the relation between the solutions of problem (21) and (23) as follows:

$$\hat{\mathbf{K}}^* - \hat{\mathbf{K}}_0 = \phi(\mathbf{X})^\top(\mathbf{A}^* - \mathbf{A_0})\phi(\mathbf{X})$$

where $\phi(\mathbf{X}) = [\phi(\boldsymbol{x_{11}}), ..., \phi(\boldsymbol{x_{1n_1}}), ..., \phi(\boldsymbol{x_{N1}}), ..., \phi(\boldsymbol{x_{Nn_N}})]$. Let $\mathbf{S} = \hat{\mathbf{K}}_0^+(\hat{\mathbf{K}}^* - \hat{\mathbf{K}}_0)\hat{\mathbf{K}}_0^+$, so that we can obtain the solution of problem (21) in the iteration form,

$$\mathbf{A}^* = \mathbf{A_0} + \phi(\mathbf{X})\mathbf{S}\phi(\mathbf{X})^\top = \mathbf{A_0} + \sum_{i,j=1}^{T} \phi(\boldsymbol{x_i}) S_{ij} \phi(\boldsymbol{x_j})^\top$$

where $\phi(\boldsymbol{x_i})$ is the i^{th} column of matrix $\phi(\mathbf{X})$ and S_{ij} is the $(i,j)^{th}$ entry of matrix $\mathbf{S}$.

Note that $\mathbf{A}^*$ is a $\mathcal{H}\times\mathcal{H}$ matrix, which cannot be explicitly expressed. Thus, we implicitly update $\mathbf{A}^*$ using the kernel trick:

$$\begin{gathered} \hat{\mathbf{K}}^*(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^\top \mathbf{A}^* \phi(\boldsymbol{y}) = \\ \hat{\mathbf{K}}_0(\boldsymbol{x}, \boldsymbol{y}) + \sum_{i,j=1}^{T} S_{ij} \mathbf{K}(\boldsymbol{x}, \boldsymbol{x_i}) \mathbf{K}(\boldsymbol{y}, \boldsymbol{x_j}). \end{gathered}$$

Therein, $\mathbf{K}(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^\top \phi(\boldsymbol{y})$ is the original kernel function. As discussed in section 3.1, $\mathbf{A_0}$ is empirically set up as the identity matrix, so that $\hat{\mathbf{K}}_0(\boldsymbol{x}, \boldsymbol{y}) = \mathbf{K}(\boldsymbol{x}, \boldsymbol{y})$. Now the problem (21) can be reduced to solve the $\hat{\mathbf{K}}^*$ in (23). The following Lemma discloses the relations between $\hat{\mathbf{K}}^*$ and the solution of the original problem (16).

Lemma 2: Suppose $\hat{\mathbf{A}}^*$ is the solution of the original problem (16) and $\hat{\mathbf{K}}^*$ is the solution of (23). Then we can conclude that $\hat{\mathbf{K}}^* = \bar{\mathbf{X}}^\top \hat{\mathbf{A}}^* \bar{\mathbf{X}}$. Therein, $\bar{\mathbf{X}} = [\boldsymbol{x_1}, ..., \boldsymbol{x_T}]$.

Proof: According to the update formula (13) for problem (16), we have

$$\hat{\mathbf{A}}_{\mathbf{t+1}} = \hat{\mathbf{A}}_{\mathbf{t}} + \beta \hat{\mathbf{A}}_{\mathbf{t}} \mathbf{M}_{\mathbf{ij}} \hat{\mathbf{A}}_{\mathbf{t}}. \tag{24}$$

Thus we can obtain its counterpart for the problem (23) as follows,

$$\hat{\mathbf{K}}_{t+1} = \hat{\mathbf{K}}_t + \beta' \hat{\mathbf{K}}_t \mathbf{E}_{\mathbf{ij}} \hat{\mathbf{K}}_t, \tag{25}$$

where

$$\mathbf{E_{ij}} = \frac{1}{n_i n_j} \sum_{a=1}^{n_i}\sum_{b=1}^{n_j} (\boldsymbol{e_{ia}} - \boldsymbol{e_{jb}})(\boldsymbol{e_{ia}} - \boldsymbol{e_{jb}})^\top. \tag{26}$$

Obviously $\beta = \beta'$. Then we use the mathematical induction to validate that at each iteration, update $\mathbf{A_t}$ and $\hat{\mathbf{K}}_t$ satisfies

$$\hat{\mathbf{K}}_t = \bar{\mathbf{X}}^\top \hat{\mathbf{A}}_t \bar{\mathbf{X}}.$$

Initially, we have

$$\hat{\mathbf{K}}_0 = \bar{\mathbf{X}}^\top \hat{\mathbf{A}}_0 \bar{\mathbf{X}},$$

so that the base case holds. Assume that

$$\hat{\mathbf{K}}_t = \bar{\mathbf{X}}^\top \hat{\mathbf{A}}_t \bar{\mathbf{X}},$$

thus we have

$$\begin{aligned} \hat{\mathbf{K}}_{t+1} &= \bar{\mathbf{X}}^\top \hat{\mathbf{A}}_t \bar{\mathbf{X}} + \beta \bar{\mathbf{X}}^\top \hat{\mathbf{A}}_t \bar{\mathbf{X}} \mathbf{E_{ij}} \bar{\mathbf{X}}^\top \hat{\mathbf{A}}_t \bar{\mathbf{X}} \\ &= \bar{\mathbf{X}}^\top (\hat{\mathbf{A}}_t + \beta \hat{\mathbf{A}}_t \mathbf{M_{ij}} \hat{\mathbf{A}}_t) \bar{\mathbf{X}}, \end{aligned}$$

which completes the proof. **END**

Given Lemma 2, we can obtain $\hat{\mathbf{K}}^*$ by solving the original optimization problem discussed in Section 3.1. Thus, the solution of the kernelized MIMEL can be achieved.

IV. Experiments

In this section, we presents the experimental results using several types of datasets. First, we employ MIMEL to address two drug activity prediction problems and three image annotation tasks, which are widely used benchmarks in multi-instance learning. We also evaluate MIMEL using two activity recognition datasets, which are collected using the GPS sensor. The detailed information about each dataset is listed in Table I.

For each dataset, we learn a Mahalanobis matrix via MIMEL, and use 1-Nearest Neighbor Classifier (1NNC) for classification, where the distance between two bags is defined as (2). To validate the effectiveness of the weighting scheme, we test the accuracy of MIMEL without using the weighting scheme (denoted by MIMEL w.o. WS). Then another Mahalanobis matrix is computed via the kernelized MIMEL (denoted by Ker-MIMEL). For the matrix learned by Ker-MIMEL, we use the SVM to classify bags in the test sets. Note, that when using SVM for classification, we indeed construct a *non-parametric kernel* between training bags and test bags via the following equations,

$$\mathbf{K}(\mathbf{X_i}, \mathbf{X_j}) = \sum_{a=1}^{n_i}\sum_{b=1}^{n_j} \hat{w}(\boldsymbol{x_{ia}})\hat{w}(\boldsymbol{x_{jb}})\hat{K}(\boldsymbol{x_{ia}}, \boldsymbol{x_{jb}})$$

where $\hat{K}(\boldsymbol{x_{ia}}, \boldsymbol{x_{jb}})$ is computed using (22). Citation-kNNC [34], a distance based multi-instance classification method is used to compare with MIMEL. The results of set distance metric work [36] are also listed for comparison. As

Table I
THE DETAILED DESCRIPTION OF THE DATASETS FOR EVALUATION

Dataset	Musk1	Musk2	Elephant	Fox	Tiger	Bike	Vehicle
Number of bags	92	102	200	200	200	120	120
Number of positive bags	47	39	100	100	100	60	60
Number of negative bags	45	63	100	100	100	60	60
Average number of instances per bag	5.2	64.7	6.96	6.6	6.1	7	7
Dimensionality of instance space	166	166	230	230	230	7	7

a baseline, we evaluate the classification performance of 1NNC (denoted by 1NNC) with three different kinds of bag distances: *Average Distance*, *Sum of Minimum Distance* and *Hausdorff Distance*. Some typical *parametric kernels* for multi-instance learning such as MI Kernel [14], MI-Graph Kernel, mi-Graph Kernel [43] and PPMM Kernel [33] are compared with Ker-MIMEL.

A. Drug Activity Recognition

In what follows, we evaluate MIMEL using the well-known databases Musk1 and Musk2. Under the two drug activity recognition datasets, every instance is represented by a 166-dimensional vector. Musk1 contains 92 bags while Musk2 contains 102 bags. More details about the two datasets can be found in [11] and Table I.

Table II
THE CLASSIFICATION ACCURACY (%) OF MIMEL, MIMEL W.O. WS, CITATION-KNNC, 1NNC, SET METRIC LEARNER, KER-MIMEL, MI KERNEL, MI-GRAPH KERNEL, MI-GRAPH KERNEL AND PPMM KERNEL UNDER THE MUSK1 AND MUSK2. SOME OTHER TYPICAL CLASSIFICATION ALGORITHMS FOR MULTI-INSTANCE LEARNING ARE ALSO LISTED AT THE BOTTOM OF THE TABLE FOR COMPARISON.

Algorithm	Musk1	Musk2
MIMEL	86.1 ± 1.2	83.4 ± 1.1
MIMEL w.o. WS	84.9 ± 1.1	82.6 ± 1.6
Citation-kNNC [34]	90.3	83.3
1NNC[36]	83.3	78.0
Set Metric Learner [36]	83.5	79.5
Ker-MIMEL	92.4 ± 0.7	$\mathbf{90.3 \pm 0.6}$
MI-Kernel [14]	88.0	89.3
MI-Graph Kernel [43]	90.0	90.0
mi-Graph Kernel [43]	88.9	**90.3**
PPMM Kernel [33]	**95.6**	81.2
MI-SVM [1]	77.9	84.3
mi-SVM [1]	87.4	83.6
MissSVM [44]	87.6	80.0
DD [24]	88.0	84.0
EM-DD [41]	84.8	84.9
APR [11]	92.4	89.2

We use MIMEL to compute a Mahalanobis matrix. The 1NNC is used to compute the classification accuracy. The typical distance based classifier Citation-kNNC and the baseline 1NNC are used for comparison. For 1NNC, we report the **best** results of the three bag distances under each dataset. The best classification accuracy of the set metric learner [36] is also used to compare with MIMEL. Note that [36] learns a set distance metric based on five kinds of set distances, respectively: *Single Linkage*, *Complete Linkage*, *Average Distance*, *Sum of Minimum Distances*, and *Hausdorff Distance*. Here we only report the **best** accuracy result among the five distances under each dataset. As for the Ker-MIMEL, we construct a non-parametric kernel and apply SVM for classification. MI Kernel, MI-Graph Kernel, mi-Graph Kernel, and PPMM Kernel are computed for comparison, where RBF kernel is used. For all the kernel methods, LibSVM [5] is applied for SVM training and prediction. For all the methods, we employ the 10-fold cross validation scheme to achieve the classification accuracy and list them in Table II. To further validate the effectiveness of our framework, we list the results of some other state-of-the-art multi-instance classification algorithms at the bottom of Table II. The best accuracy results are highlighted with figures in bold typeface. Table II shows that the classification accuracy of MIMEL outperforms the the set metric learner under almost all conditions; Ker-MIMEL outperforms most kernel methods for multi-instance learning. This is because MIMEL can obtain an appropriate metric under the multi-instance tasks. Compared with other typical classification methods for multi-instance learning, the MIMEL methods are better under most conditions. Note that the MIMEL framework is better than the MIMEL w.o. WS, which indicates the validity of the weighting scheme from another aspect. Although PPMM Kernel can achieve a little better accuracy than Ker-MIMEL on Musk1, the proposed MIMEL framework is very promising. (The results of PPMM Kernel are obtained by an exhaustive search which is impractical [33].)

B. Image Annotation

In what follows, we validate MIMEL via three image annotation datasets: Elephant, Fox, and Tiger. In each of the three tasks, we aim at detecting one corresponding category of animals, i.e., elephant, fox, and tiger respectively. The three datasets introduced in [1] are well known benchmarks in multi-instance learning. In each of the three datasets, an image (bag) is composed of a set of segments (instances), and each segment is characterized by color, texture and shape descriptors. Each of the three datasets contains 100 positive bags and 100 negative bags, and every bag contains about 7 instances on average. More information about the three datasets can be found in [1] and Table I.

Similar to last subsection, we still use the distance based

Table III
CLASSIFICATION ACCURACY (%) OF MIMEL, MIMEL W.O. WS, CITATION-KNNC, 1NNC, SET METRIC LEARNER, KER-MIMEL, MI KERNEL, MI-GRAPH KERNEL, MI-GRAPH KERNEL AND PPMM KERNEL UNDER THE ELEPHANT, FOX AND TIGER. SOME OTHER TYPICAL CLASSIFICATION ALGORITHMS FOR MULTI-INSTANCE LEARNING ARE ALSO LISTED AT THE BOTTOM OF THE TABLE FOR COMPARISON.

Algorithm	Elephant	Fox	Tiger
MIMEL	$\mathbf{86.5 \pm 0.4}$	63.0 ± 1.7	83.0 ± 0.8
MIMEL w.o. WS	85.0 ± 0.2	61.0 ± 1.5	82.5 ± 0.2
Citation-kNNC	83.0	58.5	81.5
1NNC	79.5	60.0	77.5
Set Metric Learner	85.5	**65.5**	**85.5**
Ker-MIMEL	$\mathbf{86.5 \pm 0.4}$	$\mathbf{66.5 \pm 0.4}$	$\mathbf{85.5 \pm 0.7}$
MI-Kernel	84.3	60.3	84.2
MIGraph Kernel	85.1	61.2	81.9
miGraph Kernel	**86.8**	61.6	**86.0**
PPMM Kernel	82.4	60.3	80.2
MI-SVM	81.4	59.4	84.0
mi-SVM	82.0	58.2	78.9
EM-DD	78.3	56.1	72.1

Table IV
CLASSIFICATION ACCURACY (%) OF MIMEL, CITATION-KNNC, 1NNC, SET METRIC LEARNER, KER-MIMEL, MI KERNEL, MI-GRAPH KERNEL, AND MI-GRAPH KERNEL UNDER THE TRANSPORTATION MODE.

Algorithm	Bike	Vehicle
MIMEL	90.0 ± 0.7	88.3 ± 1.9
Citation-kNNC	75.7 ± 2.4	87.3 ± 2.1
1NNC	71.7 ± 2.2	82.5 ± 1.6
Ker-MIMEL	$\mathbf{92.5 \pm 0.7}$	$\mathbf{92.5 \pm 1.2}$
MI-Kernel	90.0 ± 3.1	89.2 ± 1.7
MI-Graph Kernel	89.3 ± 2.8	88.7 ± 2.0
mi-Graph Kernel	86.7 ± 1.7	85.8 ± 2.1

methods Citation-kNNC to compare with MIMEL which is classified via 1NNC, and employ several typical multi-instance kernels to compare with Ker-MIMEL which is classified via SVM. The best results of 1NNC and the set metric learner are also reported. We still use 10-fold cross validation to achieve the average classification performance and list them in Table III. Some other multi-instance classification methods are listed at the bottom of the Table III for comparison. The results dictate that MIMEL is able to achieve the best performance under almost all the datasets compared with not only their counterparts but also other state-of-the-art multi-instance classifiers. Moreover, MIMEL is better than the MIMEL w.o. WS, which demonstrates again that the weighting scheme can capture the "key" instances within each bag and improve the classification accuracy. Compared with the set metric learner, we can conclude that the weighting scheme improves the performance of the learned metric.

C. Activity Recognition

In this section, we use two activity recognition datasets [20] for multi-instance learning collected by the GPS sensor to evaluate the proposed method.

Transportation Mode [21], [42], [20] is a commonly studied problem within activity recognition. A temporal sequence of GPS sensor-data is used to infer the transportation mode of people in daily life. The experiment is performed with 12 participants over a week who each carry a GPS device. The raw GPS log data record the longitude, latitude, and the timestamp information every 6 seconds. We split each participant's data into many traces (bags) based on a predetermined time window, so that each trace contains temporal consecutive data collected within the same length of time. We segment each trace into several sub-traces (instances) using the change point based segmentation method [42]. For each sub-trace (instance), we extract seven features: the mean velocity, the variance velocity, the maximum velocity, the minimum velocity, the mean acceleration, the variance acceleration, and the maximum acceleration. Participants are asked to label their own data at the trace (bag) level, i.e., whether a trace of data contains the following transportation activities or not: $\{bike, vehicle\}$. Thus, the task is to detect the corresponding category of transportation mode from the background mode. Each of the two tasks contains 60 positive bags and 60 negative bags.

Under the two datasets, we use MIMEL and Ker-MIMEL to compute a Mahalanobis matrix, respectively. Then 1NNC and SVM are used to compute the classification accuracy. Similar to the former experiments, Citation-kNNC and the best 1NNC are used to compare with MIMEL, while MI Kernel, MI-Graph Kernel, and mi-Graph Kernel are compared with Ker-MIMEL. We employ the 10-fold cross validation and list the average results in Table IV. The results indicate that the MIMEL framework can solve the two kinds of activity recognition problems with good accuracy. Compared with other distance based methods or kernel methods for multi-instance learning, MIMEL is superior under almost all conditions.

V. CONCLUSION

In this paper, we propose MIMEL to learn a metric distance for multi-instance learning. Mahalanobis distance function is used to define the bag distance. By employing Gaussian Model, the parameterized Mahalanobis matrix is modeled. We minimize the KL divergence between two Gaussians under certain constraints. A weighting scheme is designed to highlight "key" instances within positive bags. Moreover, we kernelize MIMEL to further enhance the classification accuracy.

In experiments, we compare the proposed MIMEL framework with Citation-kNNC, set metric learner, and a few kernel methods for multi-instance learning under some typical multi-instance learning tasks. Some other state-of-the-art classification methods for multi-instance learning are also compared with MIMEL. The results indicate that the

proposed MIMEL framework achieves better classification performance for multi-instance learning.

VI. Acknowledgement

We would like to thank Shengqi Yang from UCSB and the anonymous reviewers for their invaluable input.

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS'03*, pages 561–568, 2003.

[2] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online Multiple Instance Learning. In *CVPR'09*, pages 983–990, 2009.

[3] M. Bilenko, S. Basu and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *ICML'04*, pages 81–88, 2004.

[4] Y. Censor and S. A. Zenios. Parallel optimization: Theory, algorithm, and applications. *Oxford Uniersity Press*, 1997.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: a Library for Support Vector Machines. ACM Trans. Intelligent Systems and Technology*,2(27):1–27, 2011.

[6] Y. Chen and J. Z. Wang. Image categorization by learning and reasoning with regions. *JMLR*, 5:913–939, 2004.

[7] Y. Chen, J. Bi and J. Z. Wang. Miles: Multiple-instance learning via embedded instance selection. *TPAMI*, 28(12):1931–1947, 2006.

[8] T. Cox and M. Cox. Multidimensional Scaling. *Chapman and Hall*, 1994.

[9] J. V. Davis and I. Dhillon. Differential entropic clustering of multivariate gaussians. In *NIPS'06*, pages 337–344, 2006.

[10] J. V. Davis, B. Kulis, P. Jain, S. Sra and I. S. Dhillon. Information-theoretic metric learning. In *ICML'07*, pages 209–216, 2007.

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

[12] T. Duong, D. Phung, H. Bui and S. Venkatesh. Efficient duration and hierarchical modeling for human activity recognition. Artificial Intelligence, 173(7):830–856, 2009.

[13] G. Fung, M. Dundar, B. Krishnappuram, and R. B. Rao. Multiple instance learning for computer aided diagnosis. In *NIPS'07*, pages 425–432, 2007.

[14] T. Gartner, P. A. Flach, A. Kowalczyk, and A. Smola. Multi-instance kernels. In *ICML'02*, pages 179–186, 2002.

[15] M. Guillaumin, J. Verbeek, and C. Schmid. Multiple Instance Metric Learning from Automatically Labeled Bags of Faces. In *ECCV'10*, pages 634–647, 2010.

[16] X. He and P. Niyogi. Locality preserving projections. In *NIPS'03*, 2003.

[17] R. Jin, S. Wang and Z.-H. Zhou. Learning a distance metric from multi-instance multi-label data. In *CVPR'09*, pages 896–902, 2009.

[18] B. Kulis, M. Sustik and I. Dhillon. Learning low-rank kernel matrices. In *ICML'06*, pages 505–512, 2006.

[19] J. T. Kwok and P.-M. Cheung. Marginalized multi-instance kernels. In *IJCAI'07*, pages 901–906, 2007.

[20] N. D. Lane, Y. Xu, H. Lu, S. Hu, T. Choudhury, A. T. Campbell, and F. Zhao. Enabling Large-scale Human Activity Inference on Smartphones using Community Similarity Networks (CSN). In *Ubicomp'11*, pages 355–364, 2011.

[21] L. Liao, D. J. Patterson, D. Fox and H. Kautz. Learning and inferring transportation routines. In *Artificial Intelligence*, 171(5):311–331, 2007.

[22] B. Longstaff, S. Reddy, and D. Estrin. Improving Activity Classification for Health Applications on Mobile Devices using Active and Semi-Supervised Learning. In *Pervasive Health'10*, pages 1–7, 2010.

[23] O. Maron and A. Ratan. Multiple-instance learning for natural scene classification. In *ICML'98*, pages 341–349, 1998.

[24] O. Maron, and T. Lozano-Perez. A framework for multiple-instance learning. In *NIPS'98*, pages 570–576, 1998.

[25] W. Ping, Y. Xu, K. Ren, C.-H. Chi and F. Shen. Non-i.i.d. multi-instance dimensionality reduction by learning a maximum bag margin subspace. In *AAAI'10*, pages 551–556, 2010.

[26] W. Ping, Y. Xu, J. Wang and X.-S. Hua. FAMER: Making Multi-Instance Learning Better and Faster. In *SDM'11*, pages 594–605, 2011.

[27] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *NIPS'03*, 2003.

[28] B. Settles, M. Craven and S. Ray. Multiple-instance active learning. In *NIPS'08*, pages 1289–1296, 2008.

[29] M. Stikic and B. Schiele. Activity recognition from sparsely labeled data using multi-instance learning. In *LoCA'09, Workshop of Pervasive09*, pages 156–173, 2009.

[30] Y.-Y. Sun, M. Ng, and Z.-H. Zhou. Multi-instance dimensionality reduction. In *AAAI'10*, pages 587–592, 2010.

[31] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal Statistical Society*, 1996.

[32] P. Viola, J. C. Platt and C. Zhang. Multiple instance boosting for object detection. In *NIPS'06*, pages 1417–1424, 2006.

[33] H.-Y. Wang, Q. Yang, and H. Zha. Adaptive p-posterior mixture-model kernels for multiple instance learning. In *ICML'08*, pages 1136–1143, 2008.

[34] J. Wang and J.-D. Zucker. Solving the multiple-instance problem: a lazy learning approach. In *ICML'00*, pages 1119–1125, 2000.

[35] K. Q. Weinberger, J. Blitzer and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 10:207–244, 2009.

[36] A. Woznica and A. Kalousis. Adaptive distances on sets of vectors. In *ICDM'10*, pages 579–588, 2010.

[37] E. P. Xing, A. Y. Ng, M. I. Jordan and S. J. Russell. Distance metric learning with application to clustering with side-information. In *NIPS'02*, pages 505–512, 2002.

[38] C. Yang and T. Lozano-Perez. Image database retrieval with multiple-instance learning techniques. In *ICDE'00*, pages 233–243, 2000.

[39] L. Yang and R. Jin. Contents distance metric learning: a comprehensive survey. *Technical Report, Michigan State University*, 2006.

[40] Q. Zhang and S. A. Goldman, W. Yu, and J. E. Fritts. Content-Based Image Retrieval Using Multiple-Instance Learning. In *ICML'02*, pages 682–689, 2002.

[41] Q. Zhang and S. A. Goldman. EM-DD: an improved multiple-instance learning technique. In *NIPS'01*, pages 1073–1080, 2001.

[42] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from GPS trajectories. In *WWW'09*, pages 791–800, 2009.

[43] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li. Multi-instance learning by treating instances as non-i.i.d. samples. In *ICML'09*, pages 1249–1256, 2009.

[44] Z.-H. Zhou and J.-M. Xu. On the relation between multi-instance learning and semi-supervised learning. In *ICML'07*, pages 1167–1174, 2007.