Metric learning for multi-instance classification with collapsed bags

Dewei Li*†, Dongkuan Xu*†, Jingjing Tang*†, Yingjie Tian†‡§
*School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
†Research Center on Fictitious Economy & Data Science, Chinese Academy of Sciences, Beijing 100190, China
‡Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China
§ Email:Yingjie Tian(tyj@ucas.ac.cn, corresponding author)

***Abstract*—As a kind of popular problem in machine learning, multi-instance task has been researched by means of many classical methods, such as kNN, SVM, etc. For kNN classification, its performance on traditional task can be boosted by metric learning, which seeks for a data-dependent metric to make similar examples closer and separate dissimilar examples by a margin. It is a challenge to define distance between bags in multi-instance problem, let alone learning appropriate metric for the problem. In this paper, we propose a new approach for multi-instance classification, with the idea of metric learning embedded. A new kind of distance is used to measure the similarity between bags. To weaken redundant information from bags and reduce computation complexity, $k-$means method is implemented to get collapsed bags by replacing each instance with its corresponding cluster centroid. The aim of metric learning is to expand inter-class bag distance and shrink intra-class bag distance, leading to the construction of an optimization problem with maximal relative distance. Kernel function can be introduced into the model to extract nonlinear information from the inputs. Gradient descent is utilized to solve the problem effectively. Numerical experiments on both artificial datasets and benchmark datasets demonstrated that the method can obtain competitive performance comparative to kNN and the state-of-the-art method in multi-instance classification.**

***Index Terms*—Metric learning; Multi-instance; Clustering; Kernel**

I. Introduction

The concept of 'multi-instance' first appeared in predicting drug activity. A drug molecule contains several conformers, if one or more conformers can cause the molecule smell musky, the molecule is labeled as positive, otherwise negative. In practical, only the label of the molecule is known, we do not know which conformer plays the role. The task is to predict an unlabeled molecule via the information of conformers. In machine learning, the problem can be regarded as a multi-instance task, a molecule is a bag with multiple instances, each corresponds to a conformer. Multi-instance problem is a kind of semi-supervised problem, in which we cannot extract the label information of any instance, but only the information of feature vectors. Many real world applications can be modeled as multi-instance task, such as biomedical informatics, image classification, object detection. Multi-instance problem has attracted much attention in recent years with the raise of many effective methods. The methods can be classified into two categories: one is to represent bags precisely, including MILD[15], MILES[6], MI-kernel[10], etc.; the other is to measure bag similarity accurately via distance, including DD[17], EMDD[34], Citation-kNN[23], Clust[21], MInD[7], etc. All the methods have obtained competitive performance in multi-instance problem.

In the above distance related methods, where Euclidean distance is used to compute the distance between instances. The technique of metric learning can be introduced to learn an instance-dependent metric to measure the distance between bags more accurately. Metric learning is an effective method in improving the performance of distance related algorithms, such as kNN, where the decision rule depends heavily on appropriate distance. It can extract the structure information of original feature space and adjust relative position of each point, shrink intra-class distance and expand inter-class distance. It has been studied extensively[8, 24, 29] and applied in various applications[4, 16]. And experiments have validated the efficacy of metric learning.

In this paper, we propose a novel approach named MIML-CB(metric learning for multi-instance with collapsed bags), to learn an instance-dependent metric from collapsed bags. First, a new bag distance is introduced, more accurate than minimal Hausdorff distance and the averaged pairwise instance distance[14]. $k-$means method is used to divide all the instances from all the bags into K clusters and compress bags by replacing each instance with its corresponding cluster centroid. Inspired by the idea of large margin learning, we construct an optimization problem on neighborhood level with the principle of maximal relative distance, which make the inter-class bag distance large than the intra-class bag distance as much as possible. The problem can be converted into an unconstrained programming with respect to the transformation decomposed from the positive semidefinite metric. To further extract nonlinear information from the instances, a kernel version(ker-MIMLCB) is developed to learn a linear transformation in a high dimensional Hilbert space. Numerical experiments validate the effectiveness of MIMLCB and ker-MIMLCB in improving the performance of 1NN classification for bags.

The contributions of the paper are summarized in the following paragraphs.

- A new framework with metric learning is proposed for multi-instance classification, aims to find an instance-dependent metric by maximizing the relative distance on neighborhood level. The constructed optimization problem can be solved by gradient descent effectively.

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

- Our model can be kernelized by introducing kernel function to extract nonlinear features from the training set. Experiments verified that kernelized model performs better on most of the datasets.
- In consideration of instance contribution, similar instances should cluster together. All the instances from all the bags are partitioned into several clusters. Then the primary bags can be compressed by substituting each cluster centroid for all of its intra-cluster instances. The information of the original instances is fused into the centroids.

The structure of the paper is organized as follows. In Section II, related works for multi-instance learning and metric learning are introduced. Our linear and kernel version models are proposed in Section III, the detailed procedure for optimization will also be provided. Experiments and conclusion remarks are given in Section IV and V respectively.

II. Related works

In this section, we will give a brief introduction for related works, including multi-instance learning and metric learning.

A. Multi-instance learning

Multi-instance Learning (MIL) is generated from detection of drug activity[9], and is applied for situations where the label information is only available for the bag, a set of instances. MIL is to classify a bag, as either positive or negative. Typically, bag is considered as positive if the bag contains at least one positive instance, otherwise it is negative. MIL's ability to deal with instance label ambiguity has been used to various applications[27, 28, 31, 33]. APR[9], the first MIL method, constructs a rectangle parallel to the axis to encompass instances from positive bags but not the ones from negative bags. mi-SVM[1] tries to maximize the soft-margin over hidden instance label assignations and the discriminant function. MILD[15] discriminates the true positive instances from positive bags and provides two schemes to represent a bag. MILES[6] transforms bags into an instance-based feature vectors, then the feature selection and classification are conducted simultaneously. MI-Kernel[10] considers all instances in positive bags as positive and takes the normalized sum of a bag's instances to represent the bag. The methods based on distance are also proposed. DD[17] and EM-DD[34] label a bag based on the diverse density, using pairwise distance information. Citation-kNN[23] takes the minimum Hausdorff distance as two bags distance, and employs kNN method. MInD[7] provides several ways to measure the distance between bags, showing flexibility for various scenarios.

B. Metric learning

In distance-related methods, such as kNN, $k-$means, distance information is critical in deciding the label of pattern, which measures the similarity of two patterns. In traditional classification, Euclidean distance(squared) $d(x_i, x_j) = (x_i - x_j)^\top (x_i - x_j)$ is the most commonly used measurement, where x_i, x_j are two column vectors. However, it ignores structure information of the inputs that can be exploited to measure the similarities more accurately. Distance metric learning aims to learn a Mahalanobis metric M to measure the relationship between two points in a new way,

$$d_M(x_i, x_j) = (x_i - x_j)^\top M(x_i - x_j) \tag{1}$$

where M is a data-dependent positive semidefinite matrix. In the new measurement with M, similar points are closer to each other, and dissimilar points are separated well. To be a complete metric, M should has the property of distinguishability, non-negativity, symmetry and triangular inequality[18]. Recently, a great number of methods have been presented to demonstrate the effectiveness of data-dependent metric in improving the performance of kNN. To our knowledge, Eric. Xing proposed the earliest method in metric learning[29], which is constructed to minimize the distances between similar inputs with enforcing that dissimilar points should be separated by one unit margin. The similarity side-information is helpful for kNN classification. Goldberger presented a probability framework called NCA(neighborhood component analysis)[12] which directly maximizes the probability that every point has the same label with its nearest neighbor. Then LMNN(large margin nearest neighbor)[26] is introduced on the neighborhood level, similar as NCA. For each point and its corresponding neighbors, an ideal metric should help to pull the neighbors with the same label closer and push different labeled inputs away. Several methods have been proposed to improve the performance of LMNN[22, 25]. ITML(information-theoretic metric learning)[8] constructs Bregman optimization problem, enforcing the probability distribution of the original training sets to be proximal to a predefined Gaussian distribution. MCML[11] seeks for a proper metric to compress all the points in the same class into a single point. Extensive studies have been made on metric learning[19, 32].

In multi-instance classification, metric learning has also been applied to boost the performance. Similar as ITML, MIMEL[30] minimizes the KL divergence of two Gaussian distribution. The assumption of distribution is established on the bag level. But the influential parameter ξ_c is hard to select in MIMEL, may affect the performance seriously. With the hypothesis that two bags share at least one label are in positive pair, otherwise negative, MildML[13] learns an effective metric from multi-instance problem. However, the performance of MildML is limited since it cannot extract nonlinear information. What's more, the distance designed for bags is not accurate in MIMEL and MildML, which can be improved for better performance[7]. The two methods have not considered the information of instance distribution. In our paper, we will fuse the information of similar instances with the consideration of instance distribution. A new framework is proposed with only one trade-off parameter and it can be kernelized to extract nonlinear features.

III. Metric learning for multi-instance classification

In this section, a metric learning framework is proposed to learn an instance-dependent metric for multi-instance classification. The $k-$means technique is implemented to

get collapsed bags. Then a new kind of distance is used to measure the distance between bags more accurately. To shrink the distance of similar bags and expand the distance of dissimilar bags, relative distance margin of each bag triplet is to be maximized. The model can be further kernelized as nonlinear version for improved classification performance.

A. Multi-instance problem

Given a training set with c patterns,

$$T = \{(B'_1, Y_1), (B'_2, Y_2), \cdots, (B'_c, Y_c)\} \tag{2}$$

where $B'_i = \{x'_{i1}, x'_{i2}, \cdots, x'_{im'_i}\} \in R^{n \times m'_i}, i = 1, \cdots, c$. All the patterns are in the form of bags, which contains multiple instances. For the bag $B'_i, i = 1, \cdots, c$, the label of each instance is unknown, but the bag label $Y_i \in \{+1, -1\}$ is given. It is a kind of semi-supervised learning problem. In the original multi-instance problem, there only exist two classes of bags and a bag is positive if it contains positive instance, otherwise negative. But in multi-class setting, a bag belongs to a given class only when it contains one or several key instances. The basic goal of multi-instance classification is to learn a classifier to discriminate unlabeled bag based on the information of its instances.

B. Distance on bag level

Unlike the distance defined for instances, which has been studied throughly based on substantial theory, distance between bags is hard to be defined and there is no universally accepted measurement. Several frequently used distances includes maximal hausdorff distance, minimal horsdorff distance, average instance distance. Inspired by the average minimal instance distance[7],

$$d_{meanmin}(B'_i, B'_j) = \frac{1}{m'_i} \sum_{k=1}^{m'_i} \min_l d(x'_{ik}, x'_{jl})$$

a new bag distance is defined as following since $d_{meanmin}$ is asymmetrical, which has been proposed and studied in [14],

$$\begin{aligned} D_{am}(B'_i, B'_j; M) \quad = \quad & \frac{1}{m'_i} \sum_{p=1}^{m'_i} \min_{x'_{jl} \in B'_j} d_M(x'_{ip}, x'_{jl}) + \\ & \frac{1}{m'_j} \sum_{q=1}^{m'_j} \min_{x'_{ih} \in B'_i} d_M(x'_{jq}, x'_{ih}) \end{aligned} \tag{3}$$

For two bags B'_i, B'_j and each instance x'_{ip} in B'_i, we compute the distance between x'_{ip} and each sample in B'_j. Then the smallest distance is selected and recorded. The distance between B'_i and B'_j is the mean of these distances. Though the distance on bag level is defined, learning a proper distance metric for instances is also very important since it can be used to adjust the distance between instances, also bags, with certain targets.

C. Collapsed bag

In tradition classification, distance is computed between single feature vectors directly. But for the multi-instance problem, distance for bags is computed to reveal the relationship of different patterns, based on the calculation of pairwise distance of instances. Suppose that each bag contains m instances, the computation complexity for bag distance is m^2 times than distance between single instances. It will bring much burden on computer. Besides, there exists much redundant information in pairwise distance, which can weaken the ability of desired metric. Based on the observation that similar instances gather together on neighborhood, no matter they are from the same bag or not, we can make clustering for all the instances and each bag can be compressed by replacing every instance with its corresponds cluster centroid. The original training set (2) can be updated as

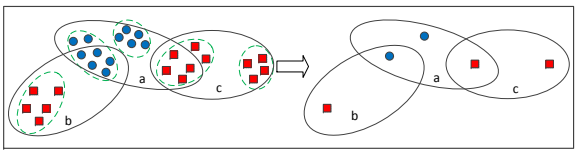


Fig. 1. The collapsed bags obtained from k-means for multi-instance problem.

$$T_{new} = \{(B_1, Y_1), (B_2, Y_2), \cdots, (B_c, Y_c)\} \tag{4}$$

where $B_i = \{x_{i1}, x_{i2}, \cdots, x_{im_i}\} \in R^{n \times m_i}, i = 1, \cdots, m$. The basic idea is shown in Figure 1. There are three bags **a, b, c** , the blue circles denote positive instances and the red squares denote negative instances. So **a** and **b** are positive bags and **c** is a ngative bag. All the instances can be clustered into five clusters, and the collapsed bags are shown in the right part of the figure, where each instance denotes a cluster centroid.

D. Model

In metric learning view, a desired metric pulls similar bags closer and pushes different labeled bags further, improving the performance of metric related methods in multi-instance classification. Based on the idea of large margin learning, for each bag B_i and its neighborhood on bag level, p_i triplets are constructed, each is in the form of (B_i, B_j, B_k), where $Y_j = Y_i, Y_k \neq Y_i$. The total number of triplets is $P = \sum_{i=1}^{m} p_i$. The relative bag distance for every triplet is defined as $r = D_{am}(B_i, B_k) - D_{am}(B_i, B_j)$. Then we expect to maximize the sum of all the relative bag distances to enlarge between-class distance and narrow within-class distance. The following optimization problem can be constructed with the regularized term to avoid overfitting,

$$\min_M \qquad J(M) = \log \sum_{s=1}^{P} \exp(-r_s) + \frac{\lambda}{2} \|M\|^2 \tag{5}$$

$$s.t. \qquad M \succcurlyeq 0 \tag{6}$$

where λ is a trade-off to balance the relative distance margin and the regularized term. Since the constraint of positive semidefinite is difficult to be handled, we consider to decompose M as $L^\top L$, where L is a matrix with appropriate dimension. Then the distance function (3) can be rewritten

as

$$D_{am}(B_i, B_j; L) = \frac{1}{m_i}\sum_{p=1}^{m_i}\min_{x_{jl}\in X_j} d(Lx_{ip}, Lx_{jl}) + \quad (7)$$

$$\frac{1}{m_j}\sum_{q=1}^{m_j}\min_{x_{ih}\in X_i} d(Lx_{jq}, Lx_{ih}) \quad (8)$$

The unconstrained optimization problem will be formulated with respect to L,

$$\min_{L} \quad J(L) = \log\sum_{s=1}^{P}\exp(-r_s) + \frac{\lambda}{2}\|L\|^2 \quad (9)$$

The target transformation L can be obtained by gradient descent algorithm, namely,

$$L = L - \eta\frac{\partial J}{\partial L} \quad (10)$$

where η is the learning rate and

$$\frac{\partial J}{\partial L} = -\frac{\sum_{s=1}^{P}\exp(-r_s)\frac{\partial r_s}{\partial L}}{\sum_{s=1}^{P}\exp(-r_s)} + \lambda L \quad (11)$$

Since $\partial r_s/\partial L = \partial D_{am}(B_i, B_k)/\partial L - \partial D_{am}(B_i, B_j)/\partial L$, we only give the computation for $\partial D_{am}(B_i, B_j)/\partial L$ as follows:

$$\begin{aligned}\frac{\partial D_{am}(B_i, B_j)}{\partial L} &= \frac{1}{m_i}\sum_{p=1}^{m_i}\frac{\partial}{\partial L}\min_{x_{jl}\in X_j} d(Lx_{ip}, Lx_{jl}) \\ &\quad + \frac{1}{m_j}\sum_{q=1}^{m_j}\frac{\partial}{\partial L}\min_{x_{ih}\in X_i} d(Lx_{jq}, Lx_{ih}) \\ &= \frac{2}{m_i}\sum_{p=1}^{m_i} L(x_{ip} - x_{jl^*})(x_{ip} - x_{jl^*})^\top \\ &\quad + \frac{2}{m_j}\sum_{q=1}^{m_j} L(x_{jq} - x_{ih^*})(x_{jq} - x_{ih^*})^\top \quad (12)\end{aligned}$$

For each $p = 1, \cdots, m_i$,

$$l^* = \arg\min_{x_{jl}\in X_j} d(Lx_{ip}, Lx_{jl}) \quad (13)$$

and for each $q = 1, \cdots, m_j$,

$$h^* = \arg\min_{x_{ih}\in X_i} d(Lx_{jq}, Lx_{ih}) \quad (14)$$

It should be noted that l^*, h^* is computed based on L, which is unknown before solving the optimization problem. In gradient descent, L will be first initialized as identity matrix to compute the derivative of J, and then updated by gradient descent iteratively. The detailed procedure for MIMLCB is given in Algorithm 1. The objective function in the primary problem (9) is a convex function and its convergence in mathematical can be ensured[3].

E. Kernel version

To extract nonlinear features from the inputs, we will introduce kernel into the previous model. The inputs will be mapped into a high dimensional Hilbert space by a nonlinear mapping: $\phi : R^n \longrightarrow H$, in which a linear transformation will be learned. The equation (15) can be rewritten as

$$\begin{aligned}&\frac{\partial D_{am}(B_i, B_j)}{\partial L} \\ &= \frac{2}{m_i}\sum_{p=1}^{m_i} L(\phi(x_{ip}) - \phi(x_{jl^*}))(\phi(x_{ip}) - \phi(x_{jl^*}))^\top \\ &\quad + \frac{2}{m_j}\sum_{q=1}^{m_j} L(\phi(x_{jq}) - \phi(x_{ih^*}))(\phi(x_{jq}) - \phi(x_{ih^*}))^\top \quad (15)\end{aligned}$$

Similar as [22], the transformation L can be parameterized by $L = \Omega\Phi$, where

$$\Phi = [\phi(x_{11}), \phi(x_{12}), \cdots, \phi(x_{1m_1}), \phi(x_{21}), \cdots, \phi(x_{cm_c})]^\top \quad (16)$$

and Ω is a matrix to combine the mapped feature points linearly. Let

$$\mathrm{k}_{ij} = \Phi\phi(x_{ij}) = [\mathrm{k}(x_{i1}, x_{ij}), \mathrm{k}(x_{i2}, x_{ij}), \cdots, \mathrm{k}(x_{cm_c}, x_{ij})]^\top \quad (17)$$

where $\mathrm{k}(a, b) = (\phi(a)\cdot\phi(b)$ is the kernel function, the gradient (15) can be transformed as

$$\begin{aligned}&\frac{\partial D_{am}(B_i, B_j)}{\partial L} \\ &= \frac{2\Omega}{m_i}\sum_{p=1}^{m_i}(\mathrm{k}_{ip} - \mathrm{k}_{jl^*})(\phi(x_{ip}) - \phi(x_{jl^*}))^\top \\ &\quad + \frac{2\Omega}{m_j}\sum_{q=1}^{m_j}(\mathrm{k}_{jq} - \mathrm{k}_{ih^*})(\phi(x_{jq}) - \phi(x_{ih^*}))^\top \\ &= \frac{2\Omega}{m_i}\sum_{p=1}^{m_i}(E_{ip}^{\mathrm{k}_{ip}-\mathrm{k}_{jl^*}} - E_{jl^*}^{\mathrm{k}_{ip}-\mathrm{k}_{jl^*}})\Phi \\ &\quad + \frac{2\Omega}{m_j}\sum_{q=1}^{m_j}(E_{jq}^{\mathrm{k}_{jq}-\mathrm{k}_{ih^*}} - E_{ih^*}^{\mathrm{k}_{jq}-\mathrm{k}_{ih^*}})\Phi \quad (18)\end{aligned}$$

where E_i^v is the $N \times N$ (N is the total number of instances) matrix with vector v in the $i-$th column and all 0 in other

---

Algorithm 1 Metric learning for multi-instance task with collapsed bags(MIMLCB)

Input: The training set T; The trade-off parameter λ, cluster number K, learning rate η, maximum of iterations τ.

Output: The target transformation L;

Procedure:

1. Let $t = 1$ and initialize L as identity matrix;
2. Partition all the instances in T into K clusters by $k-$means, compress each bag into collapsed bags and get the new training set T_{new}.
3. Input T_{new} into the problem (9), and update L by gradient descent
$$L^{t+1} = L^t - \eta\frac{\partial J}{\partial L}$$
4. Let $t = t + 1$, go to step 3 if $t > \tau$, otherwise obtain the solution $L^* = L^{t+1}$.

Algorithm 2 kernelized Metric learning for multi-instance task with collapsed bags(ker-MIMLCB)

Input: The training set T; The trade-off parameter λ, cluster number K, learning rate η, maximum of iterations τ.
Output: The target transformation Ω;
Procedure:
1. Let $t = 1$ and initialize Ω as identity matrix;
2. Partition all the instances in T into K clusters by $k-$means, compress each bag into collapsed bags and get the new training set T_{new}.
3. Input T_{new} into the problem (9), and update Ω by gradient descent

$$\Omega^{t+1} = \Omega^t - \eta\Gamma$$

4. Let $t = t + 1$, go to step 3 if $t > \tau$, otherwise obtain the solution $\Omega^* = \Omega^{t+1}$.

columns. Let

$$\Lambda_{ij}^s = \frac{2\Omega}{m_i}\sum_{p=1}^{m_i}(E_{ip}^{\mathrm{k}_{ip}-\mathrm{k}_{jl^*}} - E_{jl^*}^{\mathrm{k}_{ip}-\mathrm{k}_{jl^*}}) + \frac{2\Omega}{m_j}\sum_{q=1}^{m_j}(E_{jq}^{\mathrm{k}_{jq}-\mathrm{k}_{ih^*}} - E_{ih^*}^{\mathrm{k}_{jq}-\mathrm{k}_{ih^*}}) \tag{19}$$

then

$$\frac{\partial r_s}{\partial L} = (\Lambda_{ik}^s - \Lambda_{ij}^s)\Phi \tag{20}$$

We have

$$\begin{aligned}\frac{\partial J}{\partial L} &= -\frac{\sum_{s=1}^{P}\exp(-r_s)}{\sum_{s=1}^{P}\exp(-r_s)}(\Lambda_{ik}^s - \Lambda_{ij}^s)\Phi + \lambda\Omega\Phi \\ &= \Gamma\Phi\end{aligned} \tag{21}$$

where

$$\Gamma = -\frac{\sum_{s=1}^{P}\exp(-r_s)}{\sum_{s=1}^{P}\exp(-r_s)}(\Lambda_{ik}^s - \Lambda_{ij}^s) + \lambda\Omega \tag{22}$$

In the updating for gradient descent,

$$L^{new} = L^{old} - \eta\frac{\partial J}{\partial L} = (\Omega^{old} - \eta\Gamma)\Phi = \Omega^{new}\Phi \tag{23}$$

In the kernel version of our model, we can update and optimize Ω instead of L. After getting the solution for Ω, the points in the space H can be transformed by $L\phi(x_{ij}) = \Omega\mathrm{k}_{ij}$. The procedure for the kernel version of our model is given in Algorithm 2.

IV. Experiments

In this section, numerical experiments will be made on two kinds of datasets: artificial datasets and benchmark datasets, to show the effectiveness of our method in multi-instance classification. All the experiments were made on Matlab 2015a(PC, 8GB RAM).

A. Datasets

We will first describe our datasets since they are different in the number of instances, attributes, bags, etc. And the related applications in real world are also diverse.

Artificial datasets: **Rhombus** is a synthetic MIL dataset that has 100 positive and 100 negative bags. There are 10 instances in each bag. Negative instances and positive instances are generated from uniform distributions and normal distributions respectively. In detail, we created 6 Rhombus datasets, changing the witness rate[5], i.e., the ratio of true positive instance in each positive bag, from 10% to 60% in steps of 10%.

Benchmark datasets: This category contains the datasets of drug activity prediction, image annotation. **Musk1** and **Musk2** are two famous datasets in drug activity prediction, which are provided with the presentation of the concept of 'multi-instance'. Both are used to predicted whether a molecule has a musky smell or not. Each bag denotes a molecule and each instance corresponds to a conformer. A molecule is positive if one or more conformers can cause its corresponding molecule smell musky, otherwise negative. **Atom**, **bond** and **chain** are the problems related with predicting the mutagenicity of the molecules, whether a molecule is mutagenic or non-mutagenic. **Elephant**, **fox** and **tiger** belongs to image annotation problem. Each bag is a image and only positive ones contain the animal. Every instance in a bag denotes an image segment. **Westeast** is a task of predicting whether a train is eastbound or westbound. A train (bag) contains a variable number of cars (instances) that have different shapes and carry different loads (instance-level attributes).

The statistical information for all the datasets are shown in Table I. The column inst/bag means the number of instances per bag. The last column denotes the number of instances per collapsed bag. It can be seen that the total number of instances reduces drastically after $k-$means++, which lessen much computation complexity.

B. Baseline methods and experimental design

Our proposed model makes classification on multi-instance problem based on distance. 1NN method with the D_{am} distance instead of Euclidean distance will be used to classify bags. The new distance metric learned by our models will then be applied in 1NN. To make comparisons, the original method APR for multi-instance and distance related methods were selected as baseline methods. The information is listed as follows:

- APR[9]: It is the earliest work in solving multi-instance problem. The algorithm extends a rectangle from a seed point and the smallest rectangle that contains at least one instance of each positive bag and no instance of any negative bags is its goal. A bag can be classified by the decision boundary of the learned rectangle.
- Citation-kNN[23]: The method is an extension of kNN on multi-instance problem, also a lazy learning algorithm. A unlabeled bag can be predicted by its R nearest neighbors and C nearest citers. Hausdorff distance is used, instead of Euclidean distance, to compute the distance between different bags.

- Clust[21]: The idea of clustering is also utilized in this method. The instances form all the positive bags are divided into K clusters.
- MinD[7]: The method compares several kinds of distance for bags and analyzes the shortcomings and advantages of each measurement. The *meanmean*(MInD-m) and *minimal Hausdorff*(MInD-h) distance are used in our experiments.

The experiments on APR, Citation-kNN, Clust were implemented in the MIL tools[20]. The default settings for the parameters were implemented in experiments. In our model, the learning rate η was set to be 0.1. The best value for cluster K in $k-$means++ was selected from the set $\{2, 5, 10, 15, 20\}$. For the trade-off λ, it was selected from {2,4,8,16}. In ker-MIMLCB, the Radial basis function $K(x, z) = \exp(-\gamma\|x-z\|^2)$ was used as the kernel function and γ was chosen from $\{0.1, 1\}$. Error rate was selected as the index to evaluate the performance of our methods. The error rates of atom, bond and chain were all obtained from the mean of 3 times 10-fold cross validation. And 5 times 3-fold cross validation was used for the rest of the datasets.

C. Results and analysis

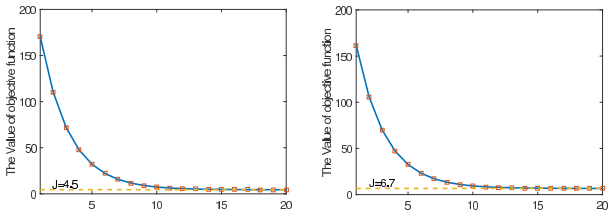
The mean error rates on artificial datasets and benchmark datasets are given in Table II and III respectively. In Table II, ker-MIML performs best in all the six datasets, and MIMLCB and ker-MIMLCB both perform better than 1NN and baseline methods on most datasets. In Table III, MIMLCB and ker-MIMLCB both obtain four best results on nine datasets. It validates the effectiveness of our model in real world datasets. It verifies three points: (1) Our model can consistently learn desired metric from collapsed bags, leading to better performance than 1NN with Euclidean distance; (2) Nonlinear transformation can extract more information than linear transformation. (3) Cluster centroid contains valid information from its intra-cluster instance. However, ker-MIMLCB performs not well on elephant, fox and tiger, which may be resulted from that nonlinear transformation in the model can not exploit useful information from image features.

TABLE I
STATISTICS OF SELECTED DATASETS.

Datasets	inst.	attr.	bags	bag+	bag-	inst/bag	inst/C-bags
Rhombus1-6	2000	2	200	100	100	10	4.5
musk1	476	166	92	47	45	5.2	1.9
musk2	6598	166	102	39	63	64.7	2.0
elephant	1391	230	200	100	100	7.0	4.1
fox	1320	230	200	100	100	6.6	3.8
tiger	1220	230	200	100	100	6.1	3.4
westeast	213	24	20	10	10	10.7	3.1
atom	1618	10	188	125	63	8.6	5.4
bond	3995	16	188	125	63	21.3	5.4
chain	5349	24	188	125	63	28.5	5.6

D. Model convergence and Parameters effects

In this section, the convergence of our proposed model and parameters effects will be investigated. We made experiments on Musk1 and set the maximal iteration to be 20. The



(a) Musk1-linear (b) Musk1-kernel

Fig. 2. The Convergence of our proposed models.

curves of objective function value of MIMLCB and ker-MIMLCB are displayed in Fig.2(a) and Fig.2(b) respectively. It is obviously that the function value become stable after the 10-th iteration, which verified the convergence of our model.

We will then explore the effects of parameters, including the number of clusters K, the trade-off λ in our model. In Figure 3, the experiments on Rhombus1, Rhombus2, Musk1, Musk2, Elephant, Fix, Atom, Bond were made to explore the effect of cluster number on mean error rate. λ, η were set to be 2 and 0.1 respectively. In ker-MIMLCB, the kernel parameter γ is 0.1. The cluster number K was searched from the set {2, 5, 10, 15, 20}. It can be seen that the lowest error rate is obtained at different cluster numbers in different datasets, which may be result from the diverse distributions. But our model performs competitively when K is in the set {2, 5, 10, 15, 20}. To investigate the effects of the trade-off λ, we made experiments on Rhombus1 and Musk1. The values for λ were selected from {1, 2, 4, 8, 16}. The values for K, η, γ are 10, 0.1, 0.1 respectively. The corresponding mean error rates were displayed in Figure 4. The error rates keep steady when λ changes from 1 to 16. It verified that our model is robust to the trade-off.

V. CONCLUSIONS AND FUTURE WORKS

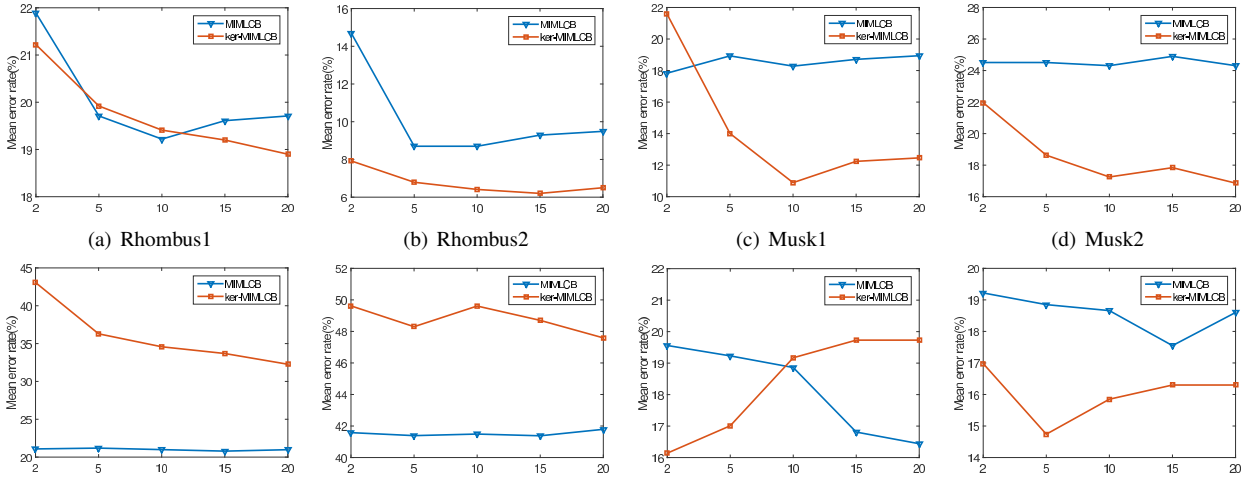
In this paper, a novel approach called MIMLCB for multi-instance classification is proposed, which utilized the technique of metric learning. A relatively better distance for bag is employed to extract more information from pairwise instance. To reduce computation burden and fuse valid information, we use $k-$means++[2] to partition all the instances into several clusters. Each bag can be transformed as collapsed bag by replacing each instance with its corresponds centroid of cluster. In the new training set, a linear transformation is learned by optimizing the goal that each bag is enforced to pull the similar bags in its neighborhood nearer and push dissimilar bags further. Gradient descent algorithm is applied to solve the convex programming. To extract nonlinear information from the inputs, MIMLCB is further kernelized as ker-MIMLCB. It maps the inputs from the original space into a high dimensional Hilbert space and looks for a Mahalanobis distance metric in the mapped feature space. Our future research will focus on the combination of deep learning and metric learning, extracting more useful features from bags and construct more proper object function for multi-instance problem.

TABLE II
CLASSIFICATION ERROR RATES ON ARTIFICAL DATASETS

Datasets	APR	Citation-kNN	Clust	MInD-m	MInD-h	1NN	MIMLCB	ker-MIMLCB
Rhombus1	44.19±7.71	18.59±3.89	33.91±5.39	43.77±6.37	45.03±7.76	20.01±4.63	17.91±4.06	**15.07±4.76**
Rhombus2	39.86±8.29	10.89±3.73	35.97±7.17	51.03±5.05	49.51±6.12	9.89±2.78	6.39±2.47	**3.00±1.50**
Rhombus3	33.51±6.51	6.99±1.99	38.83±5.89	37.68±4.91	33.64±9.34	3.58±2.29	2.58±1.71	**0.30±0.62**
Rhombus4	34.89±5.78	5.29±2.23	31.48±6.18	32.29±5.49	31.20±7.93	0.90±0.95	1.40±1.06	**0.30±0.84**
Rhombus5	37.59±11.09	3.31±2.55	36.69±6.46	28.08±4.18	20.4711.53	0.50±0.73	0.20±0.53	**0.00±0.00**
Rhombus6	35.28±6.80	3.80±2.04	38.91±5.04	25.78±5.38	8.68±5.48	0.70±0.96	0.40±0.69	**0.20±0.53**

TABLE III
CLASSIFICATION ERROR RATES ON BENCHMARK DATASETS

Datasets	APR	Citation-kNN	Clust	MInD-m	MInD-h	1NN	MIMLCB	ker-MIMLCB
Musk1	24.11±8.02	23.69±8.40	37.48±8.83	18.33±7.06	27.82±8.28	18.50±6.53	15.71±5.99	**10.89±5.23**
Musk2	22.16±8.08	22.03±5.84	37.81±4.95	27.06±7.31	31.76±7.14	24.71±5.65	23.33±5.15	**17.25±4.29**
Elephant	26.05±7.14	23.89±5.16	28.27±6.82	28.20±4.54	23.37±4.13	21.08±4.52	**20.89±4.79**	32.27±7.59
Fox	43.12±4.92	46.65±4.71	50.69±6.20	44.39±5.09	43.98±5.88	41.68±3.57	**40.89±3.51**	47.59±5.83
Tiger	41.01±5.83	25.03±6.66	43.49±5.65	26.73±3.74	27.51±4.20	**23.39±3.82**	23.59±3.90	29.80±4.15
Westeast	52.78±14.06	**36.99±21.57**	48.29±19.29	51.94±19.15	43.89±13.07	45.00±15.92	38.61±15.19	40.00±21.81
Atom	33.83±9.37	23.68±7.90	45.51±13.50	28.08±5.30	33.61±3.86	19.56±7.52	**18.31±7.97**	18.52±7.35
Bond	33.22±10.96	25.90±7.77	40.81±9.53	29.94±6.34	33.93±5.08	19.22±7.04	17.35±7.50	**14.81±6.21**
Chain	33.63±11.60	27.83±10.63	34.18±12.81	26.81±2.86	33.38±6.03	22.85±9.19	22.11±10.51	**20.99±9.53**



(a) Rhombus1 (b) Rhombus2 (c) Musk1 (d) Musk2

(e) Elephant (f) Fox (g) Atom (h) Bond

Fig. 3. Mean error rates on different numbers of cluster.

ACKNOWLEDGEMENT

This work has been partially supported by grants from National Natural Science Foundation of China (Nos .61472390, 11271361, 71331005, and 11226089), Major International (Regional) Joint Research Project (No. 71110107026) and the Beijing Natural Science Foundation (No.1162005).

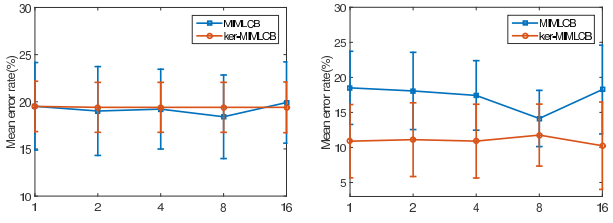
REFERENCES

[1] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Advances in neural information processing systems*, pages 561–568, 2002.

[2] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] Xinyuan Cai, Chunheng Wang, Baihua Xiao, Xue Chen,



(a) Rhombus1 (b) Musk1

Fig. 4. Mean error rates on different values for λ.

and Ji Zhou. Deep nonlinear metric learning with independent subspace analysis for face verification. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 749–752. ACM, 2012.

[5] Marc-André Carbonneau, Eric Granger, Alexandre J Raymond, and Ghyslain Gagnon. Robust multiple-instance learning ensembles using random subspace instance selection. *Pattern Recognition*, 58:83–99, 2016.

[6] Yixin Chen, Jinbo Bi, and James Ze Wang. Miles: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1931–1947, 2006.

[7] Veronika Cheplygina, David MJ Tax, and Marco Loog. Multiple instance learning with bag dissimilarities. *Pattern Recognition*, 48(1):264–275, 2015.

[8] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

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

[10] Thomas Gärtner, Peter A Flach, Adam Kowalczyk, and Alexander J Smola. Multi-instance kernels. In *ICML*, volume 2, pages 179–186, 2002.

[11] Amir Globerson and Sam T Roweis. Metric learning by collapsing classes. In *Advances in neural information processing systems*, pages 451–458, 2005.

[12] Jacob Goldberger, Geoffrey E Hinton, Sam T Roweis, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *Advances in neural information processing systems*, pages 513–520, 2004.

[13] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *Computer Vision–ECCV 2010*, pages 634–647. Springer, 2010.

[14] Dewei Li and Yingjie Tian. Multi-view metric learning for multi-instance image classification. *arXiv preprint arXiv:1610.06671*, 2016.

[15] Wu-Jun Li et al. Mild: Multiple-instance learning via disambiguation. *IEEE Transactions on Knowledge and Data Engineering*, 22(1):76–89, 2010.

[16] Jiwen Lu, Xiuzhuang Zhou, Yap-Pen Tan, Yuanyuan Shang, and Jie Zhou. Neighborhood repulsed metric learning for kinship verification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(2):331–345, 2014.

[17] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. *Advances in neural information processing systems*, pages 570–576, 1998.

[18] Halsey Lawrence Royden and Patrick Fitzpatrick. *Real analysis*, volume 198. Macmillan New York, 1988.

[19] Chunhua Shen, Junae Kim, Lei Wang, and Anton Hengel. Positive semidefinite metric learning with boosting. In *Advances in neural information processing systems*, pages 1651–1659, 2009.

[20] Cheplygina V. Tax, D.M.J. MIL, a matlab toolbox for multiple instance learning, Jun 2016. version 1.2.1.

[21] David MJ Tax, E Hendriks, Michel François Valstar, and Maja Pantic. The detection of concept frames using clustering multi-instance learning. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2917–2920. IEEE, 2010.

[22] Lorenzo Torresani and Kuang-chih Lee. Large margin component analysis. In *Advances in neural information processing systems*, pages 1385–1392, 2006.

[23] Jun Wang and Jean-Daniel Zucker. Solving multiple-instance problem: A lazy learning approach. 2000.

[24] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2005.

[25] Kilian Q Weinberger and Lawrence K Saul. Fast solvers and efficient implementations for distance metric learning. In *Proceedings of the 25th international conference on Machine learning*, pages 1160–1167. ACM, 2008.

[26] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.

[27] Jia Wu, Shirui Pan, Xingquan Zhu, and Zhihua Cai. Boosting for multi-graph classification. *IEEE transactions on cybernetics*, 45(3):416–429, 2015.

[28] Jia Wu, Xingquan Zhu, Chengqi Zhang, and S Yu Philip. Bag constrained structure pattern mining for multi-graph classification. *IEEE Transactions on Knowledge and Data Engineering*, 26(10):2382–2396, 2014.

[29] Eric P Xing, Michael I Jordan, Stuart Russell, and Andrew Y Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 505–512, 2002.

[30] Ye Xu, Wei Ping, and Andrew T Campbell. Multi-instance metric learning. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 874–883. IEEE, 2011.

[31] Yang Yi and Maoqing Lin. Human action recognition with graph-based multiple-instance learning. *Pattern Recognition*, 53:148–162, 2016.

[32] Yiming Ying and Peng Li. Distance metric learning with eigenvalue optimization. *The Journal of Machine Learning Research*, 13(1):1–26, 2012.

[33] Cha Zhang, John C Platt, and Paul A Viola. Multiple instance boosting for object detection. In *Advances in neural information processing systems*, pages 1417–1424, 2005.

[34] Qi Zhang and Sally A Goldman. Em-dd: An improved multiple-instance learning technique. In *Advances in neural information processing systems*, pages 1073–1080, 2001.