IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 2, 2020, accepted April 16, 2020, date of publication April 20, 2020, date of current version May 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988764

Optimal Representative Distribution Margin Machine for Multi-Instance Learning

TIANXIANG LUAN, TINGJIN LUO, WENZHANG ZHUGE, AND CHENPING HOU, (Member, IEEE)
Department of Systems Science, National University of Defense Technology, Changsha 410073, China

Corresponding author: Chenping Hou (hcpnudt@hotmail.com)

This work was supported by the NSF of China under Grant 61922087 and Grant 61906201, and the NSF for Distinguished Young Scholars of Hunan Province under Grant 2019JJ20020.

ABSTRACT

Multi-instance learning (MIL) plays an important role in many real applications, such as image recognition and text classification. The instance-based approach selects instances in each bag to train and has drawn significant attention recently. However, less work took the distribution information in the account and the margin distribution has been proven to be important to the generalization performance. In this paper, we propose an optimal representative distribution margin approach for multi-instance learning (MIORDM). The representative instances are the samples from the instance space and the distribution of them is important for us to find the best separation hyperplane. As the representative instances are selected iteratively, in each iteration, the instances will be more precise by the best hyperplane and the model will be better in the next iteration. In this way, a well-performed method can be derived with better generalization performance. Experiments compared with other types of state-of-the-art approaches on different datasets show that our method outperforms the others and achieves better generalization performance.

INDEX TERMS Multi-instance learning, optimal margin distribution machine, representative instance.

I. INTRODUCTION

In generic supervised learning, a class label will be annotated to each training instance. However, due to the limit of the annotation cost, precise labeling is quite difficult. So we turn to do research on multi-instance learning (MIL), which is one kind of weakly supervised learning that treats a set of instances as a bag and labels the bag instead. In this way, we can reduce the annotation cost. If there exists at least one positive instance in a bag, the label of the bag will be positive, otherwise, it will be negative [1]. Owing to this advantage for simplify the annotation cost, multi-instance learning becomes one of the most popular domain [2]–[5], and this approach has been widely applied in content-based image recognition [6], [7], text classification [8], sign language recognition [9], and so on. For example, if we segment an image into several regions [10]–[12], we can regard an image as a bag and then different regions of the image are treated as instances of it. Then, we can derive a standard multi-instance task of it.

The associate editor coordinating the review of this manuscript and approving it for publication was Bohui Wang.

Although the MIL can reduce the work of annotation, the training procedure of it will be different from the generic supervised learning for the inexact of the label information. The previous work [13]–[15] can be roughly divided into three categories. The first is mapping one bag into an instance and then convert it into a standard supervised learning task, such as Diverse Density - Support Vector Machine (DD-SVM) [11], Multiple Instance Learning by Discriminative Embedding (MILDE) [16], Multiple-Instance Learning via Embedded instance Selection (MILES) [17], Multi-instance Learning based on the Fisher Vector Representation (miFV) [18], Multi-instance Learning with Discriminative Bag Mapping (MILDM) [19]. Their common points are to find the mapping function, major by trying to find the center points. In this procedure, some unsupervised methods can be used. After that, a bag can be mapped into a single instance. Another is constructing a model by instances in each bag, then the model can be used to predict the label of a bag. Many methods contribute to it, such as Support Vector Machine for Multi-instance Learning (miSVM, MISVM) [8], Key Instance Support Vector Machine (KISVM) [20], Multiple-Instance Representative Support Vector Machine

(MIRSVM) [21]. In this way, some instances will be selected from existing data and the labels of the bags will be assigned to them. This procedure is always achieved by iteration due to the ambiguity of the label. The last one is a lazy method and the most representative method is Citation k-Nearest Neighbor (CKNN) and Bayes-KNN [22]. Among them, there does not exist a model function. CKNN predicts the test bag by adding the cited number instead of simple k nearest neighborhoods and the distance of bags is calculated by minimum Hausdorff distance [23]. Meanwhile, Bayes-KNN considers the prior probability of them to predict.

Among these methods, the second class is the most direct one. The predicted result of instances can directly reflect the label of the bag without the mapping process. What is more, these approaches can be used to identify the key instances in the bag [24] and locate regions of interest in content-based image recognition [20]. For this reason, model training by these instances is more attractive for researchers. The bag-representative method is an important one among these approaches. Recent methods often pay attention to maximize the minimum margin of the representative instances for they are commonly based on SVM. However, these methods may suffer lower generalization in this way, because the crucial margin distribution is neglected. As the Optimal Margin Distribution Machine (ODM) [25] proved, when taking margin distribution into account, a classifier with generalization performance can be derived. Besides, the representative instances can be treated as the samples selected from the instance space. Hence, they may also accord with one kind of distribution. So, doing research on the margin distribution of the representative instances is valuable.

In this paper, we propose a new method based on the optimal representative margin distribution for the multi-instance (MIORDM). This method is designed by researching the margin distribution of the representative instances. Under the distribution hypothesis, margin distribution information will be utilized to modify the decision boundary and improve the performance of generalization. Representative instances will be selected and updated in each iteration process. The main contributions of our work are threefold.

- A new objective function based on the margin distribution of the representative instances is proposed for MIL.
- An effective optimization method is proposed to solve our objective function, which is non-convex.
- Five various learning tasks (including 30 datasets) are used to validate the generality of our proposed MIORDM.

The rest of this paper is organized as follows. Section II presents recent representative SVM for MIL and optimal distribution margin machine. Section III presents details of our approach and its optimization. Then, the discussion about the parameter and computation cost is in Section IV. Section V presents the experimental environment, and the result of binary classification on representative datasets, including generalization, parameter sensitivity and time cost. Finally, conclusion is in Section VI.

II. PREPARATORY KNOWLEDGE

A. MULTI-INSTANCE REPRESENTATIVE SUPPORT VECTOR MACHINE

MIRSVM [21] is a newly proposed method which is an improved method based on traditional MISVM. In traditional MISVM, its main idea is choosing the most positive instance in each positive bag as a positive one in an iterated procedure. Hence, the MIL will be transformed into supervised learning. The selected positive ones and all the instances in the negative bags will be used to train an SVM classifier. When iteration stable, the model can be used to predict the label of new bags. In the training procedure, the number of positive instances equals to the number of positive bags. However, negative bags may contain much more negative instances in them. Hence, this will bring the imbalance problem into the model and affects the effectiveness of MISVM.

As the MISVM has serious imbalance problem, the MIRSVM aims to balance the number of selected instances in two classes. In the MIRSVM, only the least negative instance will be selected to use in each negative bag. In fact, the most positive and least negative instances are the representative instances in the MIRSVM. The model will be trained by the representative ones. In this way, the instances in two classes will be more balance than the MISVM.

B. OPTIMAL DISTRIBUTION MARGIN MACHINE

In the classification tasks, the Support Vector Machine (SVM) plays an important role among all the algorithms. The idea of maximizing the minimum margin of different classes improves the performance a lot and let the SVM become the most popular algorithm. However, as the studies on the margin theory going deeper [26], it has been proved that this large margin approach may not lead to a better generalization performance. Meanwhile, the margin distribution has been proved to be important.

Recently, some approaches based on the margin distribution have been proposed [27], [28]. In these papers, the first- and second-order statistics of the margin distribution have been concerned. Among them, the Optimal margin Distribution Machine (ODM) [25] is a representative method. In the ODM, the margins are limited in an interval and the margins outside are penalized. This approach can reduce the complexity of using first- and second-order statistics and has been proved to be superior.

III. MULTIPLE INSTANCE OPTIMAL REPRESENTATIVE DISTRIBUTION MARGIN MACHINE

In this section, we will propose a new method for multi-instance learning called Multi-instance Optimal Representative Margin Machine (MIORDM). First, we will propose the formulation of our model and analyze the details of it. Next, the optimization of the model will be given. Table(1) provides a summary of the notation used in this paper.

TABLE 1. Summary of notation used in this paper.

Notation	Definition
$\boldsymbol{B}_i$	The i-th Bag
$\boldsymbol{s}_j^i$	The j-th Instance in the i-th Bag
n	Number of the Bags
N	Number of All the Instances
d	Number of the features
Y	Label Vector of the Bags
$\mathbf{Y}$	Diagonal Label Matrix of the Bags
$\mathcal{X}^{(t)}$	The Set of Representative Instance (in t-th iteration)
$\boldsymbol{x}_i^{(t)}$	The i-th Representative Instance (in t-th iteration)
$Var()$	The Variance of the Data

A. PROPOSED METHOD

In multi-instance learning, the representative instance is a popular topic because instances are the determining factors for estimating the label of a bag. If there exists at least one positive instance the bag is positive. Otherwise, it will be negative. According to this, finding representative instances is most concerned. The bag-representative method aims to find a representative instance to represent the bag, then the label of the bag will be converted to the label of instance. In this way, separation hyperplane of bags can be converted into separation hyperplane of instances.

In previous studies, most of them only consider the separability of training data. For example, MIRSVM trains an SVM classifier on the most positive and least negative instance in each positive and negative bag. But as its training procedure mostly pay attention to maximize the minimum margin, it may neglect the crucial margin distribution of the representative instances.

As discussed above, the representative instances are very important, for they are the most valuable ones in the bags. Also, the representative instances can be treated as the samples selected from the instance space. Hence, they may also follow one kind of distribution. The margin distribution of the representative instances is attractive. For this reason, we propose a new method called MIORDM, which is based on the optimal margin of representative instances.

We denote the i-th bag as $\boldsymbol{B}_i \subseteq \mathbb{R}^{m\times d}$, meanwhile, $\boldsymbol{s}_j^i \subseteq \mathbb{R}^d$ is the j-th instance in $\boldsymbol{B}_i$, and $Y = \{+1, -1\}^n$ is a n-dimension vector which shows the label of each bag $\boldsymbol{B}$. Let $\phi : \boldsymbol{x} \to \mathbb{H}$ be a mapping function by a positive semi-definite kernel K. Then, the representative instances can be written as

$$\boldsymbol{x}_i = \max_{\boldsymbol{s}_j^i \in \boldsymbol{B}_i} \boldsymbol{w}^\top \phi(\boldsymbol{s}_j^i), \tag{1}$$

then $\boldsymbol{x}_i$ is the representative instance of i-th bag.

As the distribution hypothesis, the margins of all representative instances should obey some statistical rules, e.g., follow the Gaussian distributions. Under this assumption, the margins of instances will be mostly near the mean value, and seldom much larger or smaller than it. So that the variance can be the optimization objective. To scale $\boldsymbol{w}$, we set the minimum value to 1 and the objective function is

$$\min_{\boldsymbol{w}} \ \|\boldsymbol{w}\|^2 + \lambda Var(\boldsymbol{w}^\top \phi(\mathcal{X})Y), \tag{2}$$

$$\text{s.t. } Y_i \max_j(\boldsymbol{w}^\top \phi(\boldsymbol{s}_j^i)) \geq 1, \quad \forall\, i \leq n, \tag{3}$$

where $\mathcal{X}$ is the representative instances matrix. The first part of Eq.(2) is the regularization part, and the second part is the variance of margin, the λ is the trading-off parameter. However, simply calculating the variance can lead to large computation costs, meanwhile, when the number of samples is small, the variance of samples can not describe the globe correctly, so that this objective function needs to be improved.

First, the variance describes the discrepancy of the margin. If we assumed that the mean of the margin should be a defined value, e.g., 1. Then the L_p loss can also describe the discrepancy. To pay more attention to the larger discrepancy, L_2 loss would be used.

Secondly, constraint in Eq.(3) is too strict, in order to improve the generalization performance, soft constraint can replace as

$$Y_i \max_j(\boldsymbol{w}^\top \phi(\boldsymbol{s}_j^i)) \geq 1 - \boldsymbol{\xi}_i, \quad \forall\, i \leq n.$$

However, this is still not enough, for it only controls the margin in one direction. To measure the discrepancy, the bound of the margins should be taken into account. Now, we will fix the mean value of the margin as 1 to scale $\boldsymbol{w}$. Hence margin of representative instance $Y_i \boldsymbol{w}^\top \phi(\boldsymbol{x}_i)$ will be either larger than 1 or less than 1. If we use 2 relaxation vectors $\boldsymbol{\xi}, \boldsymbol{\epsilon}$ to denote the lower and upper bound of each margin, the constraint will be

$$\begin{cases} Y_i \max_j(\boldsymbol{w}^\top \phi(\boldsymbol{s}_j^i)) \geq 1 - \boldsymbol{\xi}_i, \\ Y_i \max_j(\boldsymbol{w}^\top \phi(\boldsymbol{s}_j^i)) \leq 1 + \boldsymbol{\epsilon}_i, \end{cases} \quad \forall\, i \leq n.$$

By this effort, the second part of Eq.(2) can be replaced by the L_2 loss of $\boldsymbol{\xi}, \boldsymbol{\epsilon}$. Now, an improved objective function can de derived as

$$\min_{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\epsilon}} \ \|\boldsymbol{w}\|^2 + \frac{\lambda}{n}(\|\boldsymbol{\xi}\|^2 + \|\boldsymbol{\epsilon}\|^2), \tag{4}$$

$$\text{s.t. } Y_i \max_j(\boldsymbol{w}^\top \phi(\boldsymbol{s}_j^i)) \geq 1 - \boldsymbol{\xi}_i, \tag{5}$$

$$Y_i \max_j(\boldsymbol{w}^\top \phi(\boldsymbol{s}_j^i)) \leq 1 + \boldsymbol{\epsilon}_i, \tag{6}$$

$$\boldsymbol{\xi}_i \geq 0, \boldsymbol{\epsilon}_i \geq 0, \boldsymbol{\xi}_i \boldsymbol{\epsilon}_i = 0, \quad \forall\, i \leq n, \tag{7}$$

where the Eq.(5), Eq.(6) show the bound of each margin, for the margin can only be one definite value, either Eq.(5) or Eq.(6) will hold, there exists at least one element in $\boldsymbol{\xi}_i, \boldsymbol{\epsilon}_i$ to be 0, another will be great than or equal to 0 only if the margin equals to 1. Hence Eq.(7) will be satisfied.

Besides, simply considering the margin distribution is not enough because the effects of margins on two sides are different. This is easy to understand that if the margin is less than 1, it will be more difficult to separate these instances of two classes so that the importance of $\boldsymbol{\xi}$ is bigger than $\boldsymbol{\epsilon}$. Therefore we should add a trading-off parameter μ to balance these 2 parts. The objective function should be modified as

$$\min_{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\epsilon}} \ \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\lambda}{n}(\|\boldsymbol{\xi}\|^2 + \mu\|\boldsymbol{\epsilon}\|^2).$$

Furthermore, another problem is that in this model we use all the margins, however, it may be unnecessary for we always

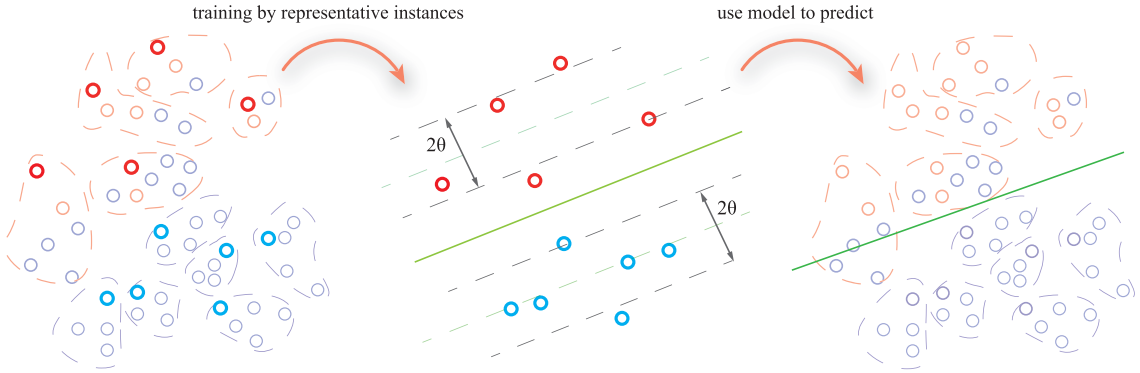


FIGURE 1. A flow chart of our approach. These red and blue circles represent the positive and negative instances, and the circles of dotted line denote bags. The bold one is the representative instance in each bag. First, we will select representative instances in each bag, then the optimal procedure derives the discriminant function. At last, this could be used to predict the label of the bag based on the label of instance.

focus much more on the small and large margins because it will control the position of the hyperplane, therefore we want to add an insensitive parameter θ, similarly to Support Vector Regression. This aims to ignore the margin values between $[1-\theta, 1+\theta]$, and penalize the margins outside of it. Only the instances, whose margins locate outside the interval $[1-\theta, 1+\theta]$, have been considered in constructing the constraints. Therefore these two vectors $\boldsymbol{\xi}, \boldsymbol{\epsilon}$ will be sparse.

According to these, we can get an improved model

$$\min_{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\epsilon}} \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\lambda}{n(1-\theta)^2}(\|\boldsymbol{\xi}\|^2 + \mu\|\boldsymbol{\epsilon}\|^2), \tag{8}$$

$$\text{s.t. } Y_i \max_j(\boldsymbol{w}^\top \phi(\boldsymbol{s}_j^i)) \geq 1-\theta-\boldsymbol{\xi}_i, \tag{9}$$

$$Y_i \max_j(\boldsymbol{w}^\top \phi(\boldsymbol{s}_j^i)) \leq 1+\theta+\boldsymbol{\epsilon}_i, \tag{10}$$

$$\theta, \mu \in [0,1), \boldsymbol{\xi}_i \geq 0, \boldsymbol{\epsilon}_i \geq 0, \boldsymbol{\xi}_i\boldsymbol{\epsilon}_i = 0, \quad \forall i \leq n,$$

where we use $(1-\theta)^2$ to scale the loss part $\|\boldsymbol{\xi}\|^2$ to $[0, 1]$, for based on Eq.(9), we have $1-\theta-\boldsymbol{\xi}_i \geq 0$.

In summary, our model aims to optimize the margin of representative instances of bags, when comes into the single instance scene, it will be as same as ODM. The whole procedure can be illustrated by Fig.(1).

B. OPTIMIZATION ALGORITHM

As shown in the previous, the optimization of our model can be divided into two parts. In the first part, we need to find the representative instances. Next, we will use these representative instances to train a classifier to predict. These two parts are coupling together. Therefore, we could solve this problem by an iterative solution.

First of all, we need to initialize the representative instance set $\mathcal{X} \subseteq \mathrm{R}^{n\times d}$. In the MISVM and MIRSVM, the prior one calculates the mean value of each positive bag as the positive instances then use these with all instances in negative bags as initial set, the superscript "+/−" denotes whether the instance or the bag is positive,

$$\begin{aligned} \boldsymbol{x}_i^+ &= \frac{1}{|\boldsymbol{B}_i^+|}\sum_{\boldsymbol{s}_j^i \in \boldsymbol{X}_i^+} \boldsymbol{s}_j^i, \\ \boldsymbol{x}^- &= \textit{all the instances in negative bags}, \\ \mathcal{X} &= \{\boldsymbol{x}_1^+, \boldsymbol{x}_2^+, \cdots; \boldsymbol{x}^-\}. \end{aligned} \tag{11}$$

The later one randomly select an instance from each bag as the initial set, this can be denoted as

$$\begin{aligned} \boldsymbol{x}_i &= \textit{RandSelect } \boldsymbol{x} \in \boldsymbol{B}_i, \\ \mathcal{X} &= \{\boldsymbol{x}_i, i = 1, 2, 3, \cdots\}. \end{aligned} \tag{12}$$

In Eq.(11), the number of negative instances is too large, this will cause a serious imbalance problem. In Eq.(12), random selection will lead to large variance in each training procedure. So we propose a new initialization method which is similar to a simple multi-instance classifier Simple-MI [29], we can calculate the mean value in each training bag as the initialize dataset,

$$\begin{aligned} \boldsymbol{x}_i &= \frac{1}{|\boldsymbol{B}_i|}\sum_{\boldsymbol{s}_j^i \in \boldsymbol{B}_i} \boldsymbol{s}_j^i, \\ \mathcal{X} &= \{\boldsymbol{x}_i, i = 1, 2, 3, \cdots\}. \end{aligned} \tag{13}$$

This will solve the imbalance problem and also can utmost save the separability of bags.

Then we will solve the objective function repeatedly to train a classifier by probably representative instances. Here we use $\boldsymbol{x}_i^t$ to represent the i-th one in the representative instance in t-th iteration $\mathcal{X}^t$, with the label vector Y. For it is an iteration procedure, the objective Eq.(8) be written as

$$\min_{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\epsilon}} \frac{1}{2}\boldsymbol{w}^\top\boldsymbol{w} + \frac{\lambda}{n(1-\theta)^2}(\boldsymbol{\xi}^\top\boldsymbol{\xi} + \mu\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}), \tag{14}$$

$$\text{s.t. } Y_i\boldsymbol{w}^\top\phi(\boldsymbol{x}_i^t) \geq 1-\theta-\boldsymbol{\xi}_i, \tag{15}$$

$$Y_i\boldsymbol{w}^\top\phi(\boldsymbol{x}_i^t) \leq 1+\theta+\boldsymbol{\epsilon}_i, \tag{16}$$

$$\theta, \mu \in [0,1], \boldsymbol{\xi}_i \geq 0, \boldsymbol{\epsilon}_i \geq 0, \boldsymbol{\xi}_i\boldsymbol{\epsilon}_i = 0, \quad \forall i \leq n. \tag{17}$$

When we analysis this objective function Eq.(14) with constraints Eq.(15) and Eq.(16), we can find that for any point, its margin value will only have three situations, lies in the interval $[1-\theta, 1+\theta]$ or smaller than $1-\theta$ or bigger than $1+\theta$. So that for the second part of the objective function, if $\boldsymbol{\xi}_i\boldsymbol{\epsilon}_i \neq 0$, there must exist a more optimization solution that will satisfy $\boldsymbol{\xi}_i\boldsymbol{\epsilon}_i = 0$, because in each situation we can optimize $\boldsymbol{\xi}_i$, $\boldsymbol{\epsilon}_i$ by set at least one parameter to be 0 to reduce the value of objective function, meanwhile it can be inferred by the same way that values of these value are all no-negative. This illustrates these constraints Eq.(17) are contained by the others. In addition, θ, μ are hyper-parameters we set before training. For this reason, we can change our model as this

$$\begin{aligned}&\min_{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\epsilon}} \frac{1}{2}\boldsymbol{w}^\top\boldsymbol{w} + \frac{\lambda}{n(1-\theta)^2}(\boldsymbol{\xi}^\top\boldsymbol{\xi} + \mu\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}),\\ &\text{s.t. } Y_i\boldsymbol{w}^\top\phi(\boldsymbol{x}_i^t) \geq 1-\theta-\boldsymbol{\xi}_i,\\ &\quad Y_i\boldsymbol{w}^\top\phi(\boldsymbol{x}_i^t) \leq 1+\theta+\boldsymbol{\epsilon}_i,\end{aligned} \tag{18}$$

This problem can be saved by the Lagrange multiplier method. The Lagrangian function of Eq.(18) can be written as

$$\begin{aligned}\min_{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\epsilon}}\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} L = &\frac{1}{2}\boldsymbol{w}^\top\boldsymbol{w} + \frac{\lambda\boldsymbol{\xi}^\top\boldsymbol{\xi}}{n(1-\theta)^2} + \frac{\lambda\mu\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}}{n(1-\theta)^2}\\ &-\sum_{i=1}^{n}\alpha_i(Y_i\boldsymbol{w}^\top\phi(\boldsymbol{x}_i^t) - 1 + \theta + \boldsymbol{\xi}_i)\\ &+\sum_{i=1}^{n}\beta_i(Y_i\boldsymbol{w}^\top\phi(\boldsymbol{x}_i^t) - 1 - \theta - \boldsymbol{\xi}_i).\end{aligned} \tag{19}$$

Calculate partial derivative of $\boldsymbol{w}$, $\boldsymbol{\xi}$, $\boldsymbol{\epsilon}$, we use $\mathbf{Y}$ denote a $n \times n$ diagonal matrix in which the diagonal elements are Y, and let $\mathbf{X}$ represents the mapping matrix of $\mathcal{X}$, which can be denoted as $\mathbf{X} = \phi(\mathcal{X})$. we have

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \mathbf{XY}(\boldsymbol{\alpha}-\boldsymbol{\beta}) = 0, \tag{20}$$

$$\frac{\partial L}{\partial \boldsymbol{\xi}} = \frac{2\lambda\boldsymbol{\xi}}{n(1-\theta)^2} - \boldsymbol{\alpha} = 0, \tag{21}$$

$$\frac{\partial L}{\partial \boldsymbol{\epsilon}} = \frac{2\lambda\mu\boldsymbol{\epsilon}}{n(1-\theta)^2} - \boldsymbol{\beta} = 0. \tag{22}$$

Then, we will get the solution of $\boldsymbol{w}$, $\boldsymbol{\xi}$, $\boldsymbol{\epsilon}$,

$$\boldsymbol{w} = \mathbf{XY}(\boldsymbol{\alpha}-\boldsymbol{\beta}), \tag{23}$$

$$\boldsymbol{\xi} = \frac{n(1-\theta)^2\boldsymbol{\alpha}}{2\lambda}, \tag{24}$$

$$\boldsymbol{\epsilon} = \frac{n(1-\theta)^2\boldsymbol{\beta}}{2\lambda\mu}. \tag{25}$$

Next, substitute Eq.(24) into Eq.(19), it will be

$$\begin{aligned}\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} L = &-\frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\beta})^\top\mathbf{YX}^\top\mathbf{XY}(\boldsymbol{\alpha}-\boldsymbol{\beta})\\ &-\frac{n(1-\theta)^2(\boldsymbol{\beta}^\top\boldsymbol{\beta} + \mu\boldsymbol{\alpha}^\top\boldsymbol{\alpha})}{4\lambda\mu}\\ &+(1-\theta)\boldsymbol{\alpha}^\top\mathbf{1} - (1+\theta)\boldsymbol{\beta}^\top\mathbf{1}.\end{aligned} \tag{26}$$

Algorithm 1 Multi-Instance Optimal Margin Distribution

1: **Input**: training data $Bags_{tr}$, parameter $\boldsymbol{w}$, $\boldsymbol{\xi}$, $\boldsymbol{\epsilon}$, $t = 0$, kernel function $\boldsymbol{K}(\cdot,\cdot)$;
2: **Output**: model parameter $\boldsymbol{\delta}$;
3: **Initialization**: initialize representative instance pool $\mathcal{X}^t$ by Eq.(13), $\boldsymbol{\delta}^{init} = \mathbf{0}$.
4: **While** not stable **do**
5: use $\mathcal{X}^t$ minimize the objective function Eq.(18), calculate $\boldsymbol{\delta}_i^t$ by Eq.(31)
6: calculate the predict value of all the training instances in all bags
7: update representative instance pool $\mathcal{X}^{t+1}$ by Eq.(32)
8: $t = t + 1$
9: **end While**

If we use $\boldsymbol{\delta} = [\boldsymbol{\alpha}^\top, \boldsymbol{\beta}^\top]^\top$, then this $2n \times 1$ vector can be added into the dual function, and we can find that it will be a simple quadratic programming with constraints as

$$\begin{aligned}\min_{\boldsymbol{\delta}} &\frac{1}{2}\boldsymbol{\delta}^\top\begin{bmatrix}\mathbf{YX}^\top\mathbf{XY} + \frac{n(1-\theta)^2}{2\lambda}\boldsymbol{I} & -\mathbf{YX}^\top\mathbf{XY}\\ -\mathbf{YX}^\top\mathbf{XY} & \mathbf{YX}^\top\mathbf{XY} + \frac{n(1-\theta)^2}{2\lambda\mu}\boldsymbol{I}\end{bmatrix}\boldsymbol{\delta}\\ &+\begin{bmatrix}(\theta-1)\mathbf{1}\\ (\theta+1)\mathbf{1}\end{bmatrix}^\top\boldsymbol{\delta},\\ \text{s.t. } &\boldsymbol{\delta} \geq 0.\end{aligned} \tag{27}$$

Note that in this function Eq.(27), all the elements are all known except λ, θ, μ are hyper-parameters which will be defined before training. This is only a convex quadratic objective function with simple constraints, as the form

$$\begin{aligned}&\min_{\boldsymbol{\delta}} f(\boldsymbol{\delta}) = \frac{1}{2}\boldsymbol{\delta}^\top\boldsymbol{A}\boldsymbol{\delta} + \boldsymbol{b}^\top\boldsymbol{\delta},\\ &\text{s.t. } \boldsymbol{\delta} \geq 0.\end{aligned} \tag{28}$$

Since it is a convex function, we can use dual coordinate descent method [30], this is based on the condition that the global optimal solution is equivalent to the optimal solution in each coordinate dimension. As we calculate iteratively, the solution will converge to the optimal solution. Through each procedure, we should update one variable $\boldsymbol{\delta}_i$, and fix the other ones as constants. What is more, in each iteration we minimize this Eq.(27), we actually update it based on an initial value $\boldsymbol{\delta}_i^{init}$. If we use $\boldsymbol{\delta}_i^{opt} = \boldsymbol{\delta}_i^{init} + k$, we can get a new formulation as

$$\min_k f(\boldsymbol{\delta}_i^{init} + k), \delta_i + k \geq 0. \tag{29}$$

This objective function of Eq.(30) is a simple quadratic function of k, when we denote the element of i-th row and j-th column as $\boldsymbol{A}_{ij}$, this can be transformed into

$$\min_k f(k) = \frac{1}{2}\boldsymbol{A}_{ii}k^2 + \nabla f(\boldsymbol{\delta}_i^{init})k + \text{constant}. \tag{30}$$

In order to solve this quadratic function with constraint $\delta_i + k \geq 0$, we can easily get a closed solution of $\boldsymbol{\delta}_i^{opt}$, written as

$$\boldsymbol{\delta}_i^{opt} = \begin{cases} 0, & \dfrac{\nabla f(\boldsymbol{\delta}_i^{init})}{\boldsymbol{A}_{ii}} \leq -\boldsymbol{\delta}_i^{init}, \\ \boldsymbol{\delta}_i^{init} - \dfrac{\nabla f(\boldsymbol{\delta}_i^{init})}{\boldsymbol{A}_{ii}}, & \text{else.} \end{cases}$$

For simplify this expression, we can denote as

$$\boldsymbol{\delta}_i^{opt} = \max(0, \boldsymbol{\delta}_i^{init} - \frac{\nabla f(\boldsymbol{\delta}_i^{init})}{\boldsymbol{A}_{ii}}). \tag{31}$$

When we update all the elements of $\boldsymbol{\delta}$ through Eq.(31), we will get the global solution of Eq.(27). Then, we can use this to predict the label of instance. Based on Eq.(23), the parameter $\boldsymbol{w} = \mathbf{XY}[\boldsymbol{I}, -\boldsymbol{I}]\boldsymbol{\delta}$, the predict value of instance z is $f(z) = \boldsymbol{w}^\top \phi(z) = \boldsymbol{\delta}^\top [\boldsymbol{I}, -\boldsymbol{I}]^\top \mathbf{Y} \phi(\mathcal{X}^t)^\top \phi(z) = \boldsymbol{\delta}^\top [\boldsymbol{I}, -\boldsymbol{I}]^\top \mathbf{Y} \boldsymbol{K}(\mathcal{X}^t, z)$.

After optimization, the next step is to update the training instance pool. This is an iterative process. We can use this classifier to calculate the predicted value of all the instances in all the training bags, then reconstruct the representative instance pool $\mathcal{X}^{t+1}$. For in each positive bag, we do not know which instance is positive, so we treat the one which has the biggest predict value as the positive for it locates furthest from the separation hyperplane. Meanwhile, in each negative bag, the one which is nearest to the separation hyperplane will be added into the representative instance pool, since it is easiest to be misclassified so that it can be treated as the bound of negative instance. This procedure can be denoted as

$$\boldsymbol{x}_i^{t+1} = \max_{\boldsymbol{s}_j^i \in \boldsymbol{B}_i} \boldsymbol{\delta}_t^\top [\boldsymbol{I}, -\boldsymbol{I}]^\top \mathbf{Y} \boldsymbol{K}(\mathcal{X}^t, \boldsymbol{s}_j^i), \tag{32}$$

where $\boldsymbol{x}_i^{t+1}$ represents the i-th representative instance in the training instances pool and $\mathcal{X}^{t+1} = \{\boldsymbol{x}_i^{t+1}, \ \forall\, i \leq n\}$.

Hence, we can get an iterated representative instances pool $\mathcal{X}^{t+1}$ based on training model by $\mathcal{X}^t$. Repeat these jobs and continue updating the parameter. The whole training procedure will stop when the representative instances pool remains unchanged. Our method can be shown by the Algorithm(1).

The main advantage of our algorithm can be summarized as follows,

- We impose the constraint in Eq.(18) to limit the margins into an interval and penalize the ones outside of this. Hence, all the information of the representative instances will be utilized. By controlling the discrepancy of the margins to modifying the decision boundary, we can get a better classification effect and generalization performance. Compared with using the variance to consider the margins, this algorithm has a simpler form and less calculation cost.
- We transform this global non-convex optimization problem into an iterated convex problem and the convergence in each iteration process is also guaranteed.
- Our initialization method will help to keep the algorithm stable and the bag-representative selection method is also an effective way.
- The optimal method is mainly based on the dual coordinate descent method, hence the optimal process is not complicated. Meanwhile, the computation complexity is not very high, shown in Subsection IV-B.

Although we have proposed an efficient algorithm in solving this non-convex optimization problem, there may be other popular optimization strategies [31], [32] that can be borrowed to tackle this problem.

IV. DISCUSSION

In this section, we will analyze the parameter of the model and the computation complexity.

A. PARAMETER ANALYSIS

In our model, we have three parameters λ, θ, μ. Here we will analyze the influence of these three according to our formulation.

As discussed above, λ in Eq.(18) is a balance parameter, which may be the most important one. It is used to balance the two parts of our formulation. In the prior section, we have illustrated that our contribution is to add the margin information to the classifier to get better performance. Now the first part of Eq.(18) is the regular terms of $\boldsymbol{w}$, and the second part is the margin part. Margin information is taken seriously, but we do not know how much we should concentrate on it. If we compare this with SVM, the λ is similar to the C in SVM. If the value is too large, when optimizing the objective function, the $\boldsymbol{w}$ may be very small to reduce the variance. In this case, the instances will be more difficult to be separated. In the opposite situation, the margin information will have less importance, and the classifier will be closer to the ordinary SVM.

Furthermore, the θ is another key parameter. It represents the width of the interval to limit the values of instances, and the one out of it will get a penalty in the objective. Therefore, this parameter can select the instances which are outside, and only the outside ones will be taken into account so that the result is sparse. The number of them can be determined by the width of the interval, hence it can control the level of sparse and reduce the calculation. As our setting is the values should be around 1, the width of the interval should be less than 2, and the θ need to be less than 1.

The last one is the weights of margins in different sides. This is easy to understand. The margin bigger than the interval contribute larger variance but easier to be separated, whereas the one less than it will both have larger variance and harder to be separated. Hence, a weight parameter should be added on the bigger ones to balance the separation level, but also this could not be too small because that will reduce its margin information.

Besides these three, another one is the kernel function. In the common scene, the RBF kernel is used widely, but when tackling data of high dimension, this will increase the complexity. Hence, in practice, proper kernel function should be selected.

B. COMPLEXITY ANALYSIS

For our method is an iterated procedure, we will analyze the complexity in each part. In the first part, we will solve the objective function to derive the model parameter, next we will use this to update the values of each instance. Assume that the number of bags is n, and the number of all the instances is N, the dimension of instance is d. Then in the first step, the time cost of calculating Eq.(27) is $O(n^2d)$. Note that we initialize the $\boldsymbol{\delta}^{init} = \mathbf{0}$, then gradient of f can be neglected, for this reason the time complexity to optimize Eq.(31) is $O(N)$. The second step is to update the predicted value of all instances, the time complexity is $O(Nnd)$. Then, assumed that iteration time is t, so the time complexity of our algorithm is $O(tnd(N + n))$.

V. EXPERIMENTS

In this section, we will introduce the experiment setup and comparison information between our method and other state-of-the-art ones on thirty benchmark datasets of binary classification. First, we will describe the basic information of these datasets and the other compared methods, then we will show the result of two criteria. For further analysis, the generalization experiment on several representative datasets also has been presented. Finally, the runtime and convergence results of each algorithm are shown. All of these are aimed to prove the efficiency of our contribution.

A. DATASET

In this part, we will introduce the datasets briefly. The details of these datasets are shown in table(2). All of them can be obtained from the web, however, some descriptions should be appended. The ''SIVAL'' is an image dataset that has 1500 images in it [6]. Among these images, there are 25 categories and each category has 60 images in different scenes. Here we use four pairs of categories to be four datasets. These four pairs are ''Apple'' and ''Banana'', ''BlueScrunge'' and ''CandleWithHolder'', ''CokeCan'' and ''DataMiningBook'', ''GlazedWoodPot'' and ''GoldMedal'', and in each pair, the prior one is treated as positive one whereas the last is negative. These 4 datasets will be called ''SIVAL-ab'', ''SIVAL-bc'', ''SIVAL-cd'', ''SIVAL-gg'' for short. Meanwhile, the ''faculty'' comes from the WebKB dataset which is a set of web pages and hyper-links data in 4 universities [33].

B. EXPERIMENT SETUP

For proving the efficiency of our method, we will introduce five state-of-the-art comparison methods in three directions.

KISVM: one state-of-the-art method, which aims to find the key instances of bags, then use these instances to train an SVM classifier. It has two versions, which is the instance level and bag level. In this paper, we will use the KISVM of bag level to compare.

EMDD: one method which calculates some hidden instances which have the maximum density by EM algorithm, then classify bags by the similarity between hidden points and instances in test bags.

TABLE 2. Details of the dataset. "P-Bags" and "N-Bags" represent the number of positive and negative bags respectively and "Total" is the total number of bags. "Instances" denotes the number of all instances, "Average" is the average number of instances in bags.

Dataset	features	P-Bags	N-Bags	Total	Instances	Average
musk1	166	47	45	13	2898	222.92
musk2	166	39	62	101	6728	66.61
tiger	230	100	100	200	1188	5.94
elephant	230	100	100	200	1391	6.96
fox	230	100	100	200	1320	6.60
SIVAL	30	60	60	120	3780	31.50
20News	200	50	50	100	6728	42.31
faculty	361	795	795	1590	4248	2.67

CKNN: one lazy method which is based on KNN, but introduced minimum Hausdorff distance to calculate the distance between bags, the label of test bag is determined by the number of neighbors and citation.

MILDM: one state-of-the-art method which was proposed in 2018, which used the spectral clustering method to construct a discriminative instance pool(DIP). Thus, bags can be mapped into instances to change the MIL task into a single-instance task. Then, KNN is used to predict the mapped instances of test bags. Here we use all the instances to obtain the DIP, called aMILGDM.

MIRSVM: one state-of-the-art method which was an improved one of MISVM in 2018, which use representative instances to denote each bag and training classifier by them.

To measure the efficiency overall, we will use two criteria, accuracy and AUC. All the experiments were run on an Intel i7-8500U CPU with 8GB RAM, and all methods are implemented in MATLAB.

C. BINARY CLASSIFICATION EXPERIMENT

In this part, we will do experiments on the datasets introduced above, and compare our method with the other representative method. Before the experiment, we need to do data preprocessing. All features are normalized into the interval [0, 1]. Then, randomly segments 80 percent of data as training data balanced, meanwhile, the left will be the test data. Besides, to eliminate the randomness, repeat this procedure 50 times and record the result in each time. As three parameters need to be selected in our method, we use 5-fold cross-validation to choose them, and λ is selected from the set $\{2^7, 2^8, 2^9, 2^{10}\}$, meanwhile θ, μ are chose in the set $\{0.6, 0.7, 0.8\}$. In application, we use the RBF Kernel to map instances into Kernel space, and the width of kernel function default equals the reciprocal of the dimension of the instance. The parameters in other methods are selected by 5-fold cross-validation. The result of the classification experiments are shown in table(3, 4). Meanwhile, to analyze the performance of these methods, we also use the non-parametric statistical test to validate the results [34], [35]. In this paper, the Friedman test has been used to show the rank of methods over these datasets and the test statistic proves that there exist significant differences among the results. The highest value of rank is the best one. When we have assumed the performances of these methods are different, the Bonferroni-Dunn post-hoc test [36] is used to find a critical value to judge the level of significant differences. The ranks are represented in the last row of each

TABLE 3. Accuracy of classification results of our method and compared ones on thirty datasets. Front one of each element in the table is the mean value of accuracy, meanwhile the second is the standard deviation of it. The last row shows the rank of each method by using the Friedman test. The best result of each dataset has been overstriking.

Dataset	KISVM	EMDD	C-KNN	aMILGDM	MIRSVM	MIORDM
musk1	0.7722(0.0819)	0.8300(0.0872)	**0.8733(0.0786)**	0.8678(0.0832)	0.6322(0.0614)	0.7589(0.0895)
musk2	0.6347(0.0948)	0.8032(0.0914)	**0.8137(0.0904)**	0.7737(0.1065)	0.6316(0.0000)	0.7326(0.0829)
tiger	0.6910(0.1007)	0.7135(0.0702)	0.7480(0.0560)	0.6400(0.0707)	0.7345(0.0747)	**0.8305(0.0530)**
elephant	0.7425(0.0693)	0.7910(0.0654)	0.7835(0.0605)	0.7390(0.0653)	0.6905(0.0696)	**0.7955(0.0599)**
fox	0.5400(0.0678)	0.5930(0.0672)	**0.5940(0.0745)**	0.5785(0.0727)	0.5110(0.0589)	0.5830(0.0688)
sival-ab	0.5458(0.0946)	0.7458(0.1386)	0.6258(0.0755)	**0.7767(0.0718)**	0.6425(0.0964)	0.7600(0.0939)
sival-bc	0.7450(0.0819)	0.7392(0.1630)	0.7350(0.0853)	0.9525(0.0469)	0.5375(0.0908)	**0.9600(0.0347)**
sival-cd	0.6408(0.1222)	0.8158(0.0915)	0.4317(0.0742)	0.7808(0.1061)	0.7292(0.0868)	**0.9283(0.0404)**
sival-gg	0.5567(0.0677)	0.7150(0.1459)	0.7083(0.0000)	**0.8525(0.0768)**	0.6183(0.0791)	0.8392(0.0850)
faculty	0.5347(0.0255)	0.8743(0.0155)	0.8767(0.0142)	0.6533(0.0177)	0.8563(0.0342)	**0.9375(0.0179)**
alt.atheism	0.5960(0.0862)	0.4970(0.0982)	0.5000(0.0000)	0.4820(0.0734)	0.5410(0.0690)	**0.6680(0.1092)**
comp.graphics	0.4811(0.0412)	0.5158(0.1184)	0.5189(0.0592)	0.5705(0.0886)	0.5263(0.0000)	**0.6284(0.0917)**
comp.os.ms-windows.misc	0.5140(0.0663)	0.5010(0.0619)	0.5000(0.0000)	0.4890(0.0816)	0.5110(0.0253)	**0.5890(0.1094)**
comp.sys.ibm.pc.hardware	0.4863(0.0814)	0.4989(0.1021)	0.4221(0.1011)	**0.5579(0.0789)**	0.5263(0.0000)	0.5253(0.0912)
comp.sys.mac.hardware	0.5770(0.0996)	0.4690(0.0775)	0.4810(0.0552)	0.4860(0.0808)	0.5140(0.0304)	**0.6140(0.0985)**
comp.windows.x	0.6758(0.0923)	0.5232(0.0965)	0.4179(0.0659)	0.5368(0.0963)	0.5263(0.0000)	**0.7147(0.0845)**
misc.forsale	0.5160(0.0823)	0.5120(0.0619)	0.5320(0.0748)	0.5000(0.0926)	0.5150(0.0323)	**0.5340(0.1002)**
rec.autos	**0.6540(0.1049)**	0.4870(0.0885)	0.4890(0.0209)	0.4990(0.1052)	0.5190(0.0348)	0.6340(0.0854)
rec.motorcycles	0.5140(0.1258)	0.5100(0.0857)	0.5000(0.0000)	0.5350(0.0944)	0.5100(0.0267)	**0.6350(0.1162)**
rec.sport.baseball	0.6320(0.1019)	0.5200(0.1152)	0.4970(0.0409)	0.5770(0.0876)	0.5130(0.0332)	**0.6980(0.0937)**
rec.sport.hockey	0.8070(0.0904)	0.4950(0.0975)	0.5210(0.0581)	0.5210(0.0701)	0.5350(0.0518)	**0.8210(0.0858)**
sci.crypt	0.6000(0.0000)	0.4810(0.1245)	0.4820(0.0698)	0.5400(0.0845)	0.5210(0.0287)	**0.6550(0.0847)**
sci.electronics	0.4768(0.0929)	0.5105(0.0929)	0.5263(0.0000)	0.5232(0.0165)	0.5263(0.0000)	**0.5758(0.0684)**
sci.med	0.6520(0.0892)	0.5080(0.1037)	0.5140(0.0417)	0.4960(0.0908)	0.5250(0.0368)	**0.6830(0.0831)**
sci.space	**0.7230(0.0910)**	0.4980(0.1015)	0.5100(0.0350)	0.5410(0.0873)	0.5160(0.0384)	0.7090(0.0988)
soc.religion.christian	0.5520(0.0833)	0.4990(0.1086)	0.5770(0.0656)	0.5010(0.1002)	0.5710(0.0598)	**0.6430(0.0964)**
talk.politics.guns	0.5800(0.1233)	0.4560(0.0983)	0.5690(0.1039)	0.5140(0.1040)	0.5870(0.0604)	**0.6670(0.1146)**
talk.politics.mideast	0.6350(0.0986)	0.4520(0.0820)	0.5200(0.0404)	0.5260(0.1084)	0.5370(0.0523)	**0.7390(0.0716)**
talk.politics.misc	0.5920(0.0835)	0.4810(0.0947)	0.5070(0.0267)	0.5850(0.0771)	0.5450(0.0527)	**0.6500(0.0942)**
talk.religion.misc	0.5442(0.1018)	0.4968(0.0916)	0.5453(0.0938)	0.5411(0.0789)	0.5263(0.0000)	**0.6116(0.1068)**
Rank	3.53	2.52	3.07	3.32	3.07	**5.50**

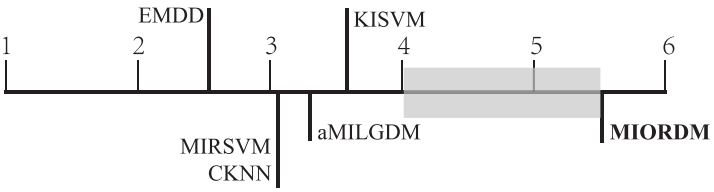


FIGURE 2. Bonferroni-Dunn test of Accuracy. The Grey area shows the critical value.

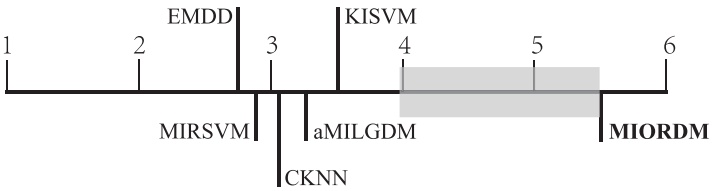


FIGURE 3. Bonferroni-Dunn test of AUC. The Grey area shows the critical value.

table and the results of the Bonferroni-Dunn test are shown in Figure.(2) and Figure.(3).

These tables show that our method beats the others on most datasets. Especially, our method has an outstanding performance on the image and text datasets, which have explicit distribution information in them.

From this, we can realize that, when we consider the margin information, we can truly get an efficient classifier that has a better performance than the state-of-the-art methods. Something needs to be explained that, as the normalization of data is important to the classifier, and the way of normalizing may be more suitable for our method, and also the other ones are affected.

D. GENERALIZATION EXPERIMENT

As we discussed above, when we consider the margin information, the separation hyperplane will be more similar to the ground truth. In application, this property can be shown as a higher generalization. Especially when the number of training data is small, we can still have a good performance. This is owing to the margin distribution reduce the sample variance, whereas the differences between globe variance and sample variance are the primary cause of error in classification.

For proving our theory, in this section, we will do experiments on several representative datasets to test the generalization. Among the procedure, we will control the training ratio of the whole dataset from range

TABLE 4. AUC of classification results of our method and compared ones on thirty datasets. Front one of each element in the table is the mean value of AUC, meanwhile the second is the standard deviation of it. The last row shows the rank of each method by using the Friedman test. The best result of each dataset has been overstriking.

Dataset	KISVM	EMDD	C-KNN	aMILGDM	MIRSVM	MIORDM
musk1	0.7647(0.1182)	0.8220(0.1126)	**0.8681(0.0990)**	0.8607(0.1003)	0.6198(0.1349)	0.7553(0.1245)
musk2	0.6667(0.1391)	0.7910(0.1212)	**0.8005(0.1173)**	0.7671(0.1244)	0.4790(0.1387)	0.7719(0.1019)
tiger	0.6767(0.1183)	0.7048(0.0760)	0.7422(0.0619)	0.6294(0.0837)	0.7216(0.0930)	**0.8238(0.0598)**
elephant	0.7399(0.0785)	0.7941(0.0794)	0.7861(0.0723)	0.7398(0.0786)	0.6953(0.0918)	**0.8015(0.0621)**
fox	0.5250(0.0960)	0.5742(0.0798)	**0.5826(0.0930)**	0.5659(0.0906)	0.4999(0.0830)	0.5618(0.0915)
sival-ab	0.5474(0.1606)	0.7454(0.1531)	0.6278(0.1196)	**0.7767(0.0959)**	0.6386(0.1523)	0.7604(0.1218)
sival-bc	0.7493(0.1115)	0.7369(0.1727)	0.7353(0.1070)	0.9550(0.0475)	0.5453(0.1265)	**0.9621(0.0418)**
sival-cd	0.6235(0.1621)	0.8062(0.1190)	0.4194(0.1114)	0.7814(0.1211)	0.7201(0.1248)	**0.9318(0.0590)**
sival-gg	0.5561(0.1334)	0.7044(0.1651)	0.7778(0.0000)	**0.8443(0.0959)**	0.6151(0.1190)	0.8343(0.1026)
faculty	0.5356(0.0291)	0.8748(0.0153)	0.8769(0.0179)	0.6551(0.0277)	0.8558(0.0365)	**0.9375(0.0186)**
alt.atheism	0.5906(0.1054)	0.4910(0.1482)	0.4886(0.1415)	0.4780(0.1237)	0.5288(0.1603)	**0.6554(0.1272)**
comp.graphics	0.5236(0.1510)	0.5238(0.1527)	0.5558(0.1324)	0.5696(0.1468)	0.5211(0.1436)	**0.6313(0.1406)**
comp.os.ms-windows.misc	0.5298(0.1472)	0.5082(0.1428)	0.5078(0.1467)	0.4996(0.1249)	0.5170(0.1466)	**0.5888(0.1412)**
comp.sys.ibm.pc.hardware	0.4827(0.1301)	0.4878(0.1397)	0.4124(0.1321)	**0.5227(0.1433)**	0.4762(0.1468)	0.5127(0.1320)
comp.sys.mac.hardware	0.5798(0.1202)	0.4682(0.1267)	0.4778(0.1340)	0.4922(0.1278)	0.5190(0.1292)	**0.6228(0.1299)**
comp.windows.x	0.6844(0.1128)	0.5144(0.1229)	0.4156(0.1078)	0.5082(0.1401)	0.4889(0.1085)	**0.7064(0.1049)**
misc.forsale	0.5076(0.1391)	0.4936(0.1200)	**0.5296(0.1296)**	0.4878(0.1147)	0.5100(0.1346)	0.5260(0.1389)
rec.autos	**0.6516(0.1394)**	0.4892(0.1218)	0.4808(0.1391)	0.4872(0.1350)	0.5120(0.1388)	0.6282(0.1283)
rec.motorcycles	0.5150(0.1632)	0.5232(0.1400)	0.5088(0.1424)	0.5414(0.1432)	0.5208(0.1426)	**0.6400(0.1458)**
rec.sport.baseball	0.6268(0.1460)	0.5136(0.1588)	0.4898(0.1433)	0.5774(0.1479)	0.5018(0.1504)	**0.6862(0.1203)**
rec.sport.hockey	0.8184(0.0998)	0.5014(0.1399)	0.5374(0.1342)	0.5232(0.1356)	0.5470(0.1443)	**0.8300(0.1140)**
sci.crypt	0.5800(0.0000)	0.4868(0.1309)	0.4940(0.0998)	0.5470(0.1044)	0.5332(0.1077)	**0.6604(0.1011)**
sci.electronics	0.4793(0.1120)	0.4902(0.1364)	0.4973(0.1185)	0.4922(0.1185)	0.4973(0.1185)	**0.5522(0.1345)**
sci.med	0.6494(0.1064)	0.5038(0.1354)	0.5070(0.1334)	0.4958(0.1309)	0.5174(0.1343)	**0.6788(0.1273)**
sci.space	**0.7138(0.0955)**	0.4940(0.1290)	0.4994(0.1069)	0.5380(0.1199)	0.5076(0.1090)	0.7038(0.1110)
soc.religion.christian	0.5366(0.1337)	0.4894(0.1319)	0.5726(0.1122)	0.5030(0.1449)	0.5652(0.1414)	**0.6366(0.1272)**
talk.politics.guns	0.5964(0.1508)	0.4624(0.1153)	0.5726(0.1327)	0.5198(0.1403)	0.5968(0.1285)	**0.6734(0.1398)**
talk.politics.mideast	0.6422(0.1119)	0.4530(0.1342)	0.5260(0.1262)	0.5362(0.1430)	0.5428(0.1268)	**0.7396(0.0972)**
talk.politics.misc	0.5898(0.1403)	0.4698(0.1454)	0.4956(0.1372)	0.5752(0.1155)	0.5318(0.1420)	**0.6402(0.1173)**
talk.religion.misc	0.5484(0.1298)	0.5073(0.1264)	0.5493(0.1259)	0.5207(0.1140)	0.4953(0.1290)	**0.5987(0.1448)**
Rank	3.48	2.83	3.05	3.24	2.91	**5.48**

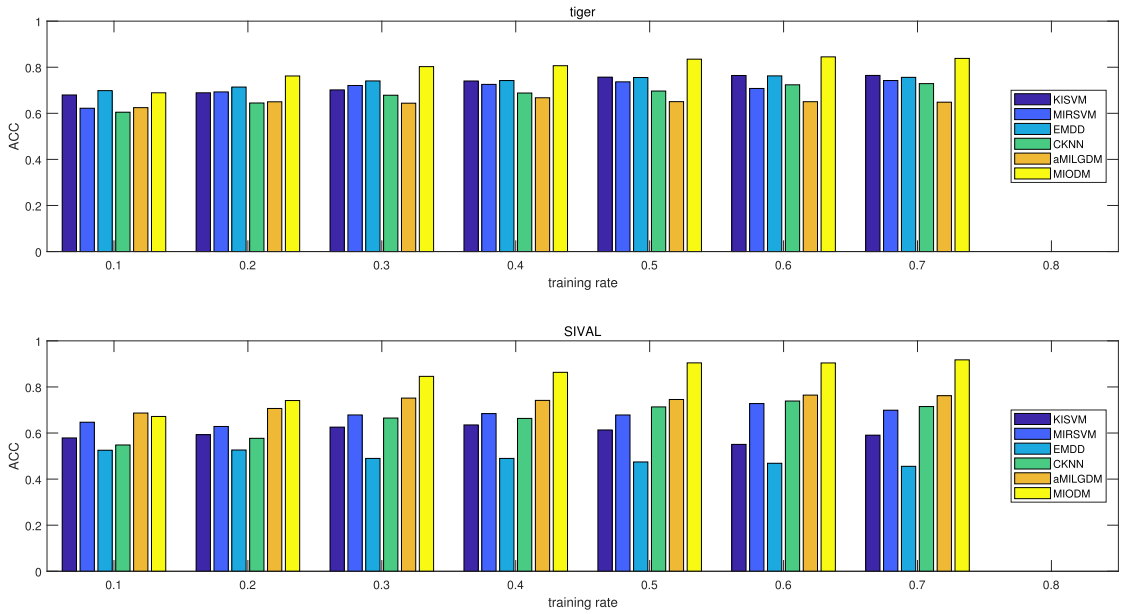


FIGURE 4. The histogram of the accuracy of MIORDM on two datasets with different training rates. The results are the mean value of accuracy after repeating 30 times.

{10%, 20%, 30%, 40%, 50%, 60%, 70%}, and randomly select training data to train, then test our method and other compared methods on the rest. The whole procedure will repeat 30 times to reduce the stochastic error and record the result in Fig.(4).

From this table, we can find that, although the training rate is low, our method can still achieve well performance which is closer to one of the high training ratio, whereas the others do worse than ours.

E. PARAMETER SENSITIVITY

In this section, we want to find out that the influence of the parameter variation. Fist, we will fix the value of $\mu = 0.6, 0.7, 0.8$, then test the performance of our method with parameter λ and θ respectively range from set $\{2^0, 2^1, 2^2, \cdots, 2^8, 2^9, 2^{10}\}$ and $\{0.1, 0.2, 0.3, \cdots, 0.8, 0.9\}$. Here we use two datasets, "tiger" and "SIVAL-cd". The result of accuracy is presented in Fig.(5).

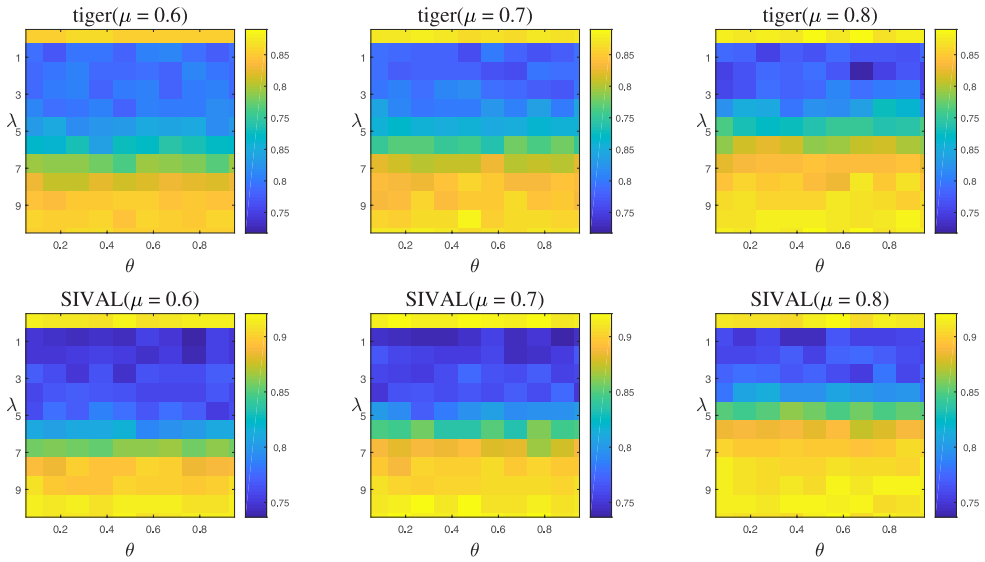


FIGURE 5. The heat map of the accuracy of MIORDM and compared methods on two datasets with different parameter pairs, the color bars in each row are the same. The results are the mean value of accuracy after repeating 50 times.

TABLE 5. The training time (CPU time measured in seconds) of our algorithms and corresponding baselines.

Dataset	KISVM	EMDD	C-KNN	aMILGDM	MIRSVM	MIORDM
musk1	0.44	52.45	1.41	0.35	0.07	0.08
musk2	1.21	64.43	300.58	58.65	0.82	1.16
tiger	1.71	220.49	26.62	3.46	0.70	0.35
elephant	2.26	165.78	29.84	3.54	0.60	0.30
fox	1.51	169.38	30.81	3.90	0.67	0.36
sival-ab	1.31	12.88	105.21	20.70	0.21	0.23
sival-bc	3.61	13.69	109.16	21.06	0.16	0.23
sival-cd	4015.15	17.28	141.04	25.82	0.26	0.29
sival-gg	8130.30	16.25	126.74	23.61	0.24	0.28
rec.autos	1.70	103.11	106.09	18.64	0.70	0.13
sci.med	1.47	106.13	82.20	14.54	0.64	0.12
faculty	45.60	5912.22	3789.70	60.98	33.69	28.84

From this Fig.(5), we can find that the parameter λ plays the most important role in our model. When the value increased, the accuracy in the "tiger" increases by nearly 10 percent. Meanwhile, as the μ increased, the yellow area expansions, which is more obvious on the "SIVAL-cd". Yet, θ influences the results lightly. However, the value of λ should not be too large, for the performance descend a little, the best interval of λ seems to be [8, 10].

In conclusion, our model really depends on the the parameter λ, meanwhile it is less sensitive to μ and θ.

F. TIME COST

Also, for demonstrating our method overall, in this part, we will show the CPU time cost of each method on some representative datasets. The result table(5) displays the details. From this, we can find that, compared with others, our method has a comparable time cost with MIRSVM, whereas others are larger orders of magnitude than ours. Overall, our contribution has been proved by the time cost and classification performance.

VI. CONCLUSION

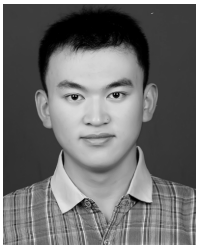
In recent studies on the MIL task, changing the separation hyperplane of bags into one of the representative instances is a valuable topic. Among these, we found that distribution information seldom anticipated the classifier training process. However, ignoring this may relate to a larger variance of data and reduce the generalization performance of the method. This paper proposed a novel algorithm for the MIL tasks, which try to optimize the distribution margin of the representative instances of bags. Besides, experiments on thirty benchmark datasets proved the effectiveness of our method with other state-of-the-art ones and illustrated the well performance of generalization.

Some foundational improvement can also be done on my work, the margin information of selected representative instances are still rough for the instances we selected might be mislabeled, some improvement could pay attention on how to maximize the value of margin of instances without exact labels. Meanwhile, the margin contribution of different classes may be different, but it is hard to measure. An additional application is also needed for this topic.

REFERENCES

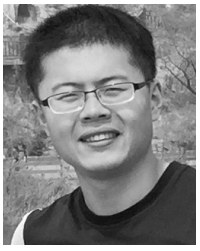
[1] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning.," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 10, 1997, pp. 570–576.
[2] S. Huang, W. Gao, and Z. Zhou, "Fast multi-instance multi-label learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2614–2627, Jul. 2019.
[3] X.-S. Wei, H.-J. Ye, X. Mu, J. Wu, C. Shen, and Z.-H. Zhou, "Multiple instance learning with emerging novel class," *IEEE Trans. Knowl. Data Eng.*, early access, 2019, doi: 10.1109/TKDE.2019.2952588.
[4] J. Wu, W. Zhuge, X. Liu, L. Liu, and C. Hou, "Fragmentary multi-instance classification," *IEEE Trans. Cybern.*, early access, 2019, doi: 10.1109/TCYB.2019.2938206.
[5] Y. Li, L. Guo, and Z. Zhou, "Towards safe weakly supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, 2019, doi: 10.1109/TPAMI.2019.2922396.
[6] W.-J. Li and D.-Y. Yeung, "Localized content-based image retrieval through evidence region identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1666–1673.
[7] H. Yuan, M. Fang, and X. Zhu, "Hierarchical sampling for multi-instance ensemble learning," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 12, pp. 2900–2905, Dec. 2013.

[8] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. Adv. Neural Inf. Process. Syst. Neural Inf. Process. Syst. (NIPS)*, 2002, pp. 561–568.
[9] D. Kelly, J. McDonald, and C. Markham, "Weakly supervised training of a sign language recognition system using multiple instance learning density matrices," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 2, pp. 526–541, Apr. 2011.
[10] C. Carson, M. Thomas, J. S. Belongie, M. J. Hellerstein, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval," in *Proc. 3rd Int. Conf. Vis. Inf. Inf. Syst. (VISUAL)*, 1999, pp. 509–516.
[11] Y. Chen and J. Z. Wang, "Image categorization by learning and reasoning with regions," *J. Mach. Learn. Res.*, vol. 5, pp. 913–939, Aug. 2004.
[12] C. Yang, M. Dong, and F. Fotouhi, "Region based image annotation through multiple-instance learning," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, 2005, pp. 435–438.
[13] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *Artif. Intell.*, vol. 201, pp. 81–105, Aug. 2013.
[14] M.-A. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon, "Multiple instance learning: A survey of problem characteristics and applications," *Pattern Recognit.*, vol. 77, pp. 329–353, May 2018.
[15] M.-L. Zhang, "Generalized multi-instance learning: Problems, algorithms and data sets," in *Proc. WRI Global Congr. Intell. Syst.*, vol. 3, 2009, pp. 539–543.
[16] J. Amores, "MILDE: Multiple instance learning by discriminative embedding," *Knowl. Inf. Syst.*, vol. 42, no. 2, pp. 381–407, Feb. 2015.
[17] Y. Chen, J. Bi, and J. Z. Wang, "MILES: Multiple-instance learning via embedded instance selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1931–1947, Dec. 2006.
[18] X.-S. Wei, J. Wu, and Z.-H. Zhou, "Scalable algorithms for multi-instance learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 4, pp. 975–987, Apr. 2017.
[19] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Multi-instance learning with discriminative bag mapping," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1065–1080, Jun. 2018.
[20] Y.-F. Li, T. J. Kwok, W. I. Tsang, and Z.-H. Zhou, "A convex method for locating regions of interest with multi-instance learning," in *Machine Learning and Knowledge Discovery in Databases (LNAI)*, vol. 5782. Berlin, Germany: Springer, 2009, pp. 15–30.
[21] G. Melki, A. Cano, and S. Ventura, "MIRSVM: Multi-instance support vector machine with bag representatives," *Pattern Recognit.*, vol. 79, pp. 228–241, Jul. 2018.
[22] J. Wang and J. D. Zucker, "Solving the multiple-instance problem: A lazy learning approach," in *Proc. 17th Int. Conf. Mach. Learn. (ICML)*, 2000, pp. 1119–1126.
[23] G. Edgar, *Measure, Topology, and Fractals Geometry*. New York, NY, USA: Springer, 1990.
[24] Z. Fu, A. Robles-Kelly, and J. Zhou, "MILIS: Multiple instance learning with instance selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 958–977, May 2011.
[25] T. Zhang and Z.-H. Zhou, "Optimal margin distribution machine," *IEEE Trans. Knowl. Data Eng.*, early access, 2019, doi: 10.1109/TKDE.2019.2897662.
[26] W. Gao and Z.-H. Zhou, "On the doubt about margin explanation of boosting," *Artif. Intell.*, vol. 203, pp. 1–18, Oct. 2013.
[27] T. Zhang and Z.-H. Zhou, "Large margin distribution machine," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: Association for Computing Machinery, Nov. 2014, pp. 313–322.
[28] Z.-H. Tan, P. Tan, Y. Jiang, and Z.-H. Zhou, "Multi-label optimal margin distribution machine," *Mach. Learn.*, vol. 109, no. 3, pp. 623–642, Mar. 2020.
[29] L. Dong, "A comparison of multi-instance learning algorithms," M.S. thesis, Univ. Waikato, Hamilton, New Zealand, 2006.
[30] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear SVM," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 408–415.
[31] Y. Chang, Y. Wang, F. Alsaadi, and G. Zong, "Adaptive fuzzy output-feedback tracking control for switched stochastic pure-feedback nonlinear systems," *Int. J. Adapt. Control Signal Process.*, vol. 33, no. 10, pp. 1567–1582, Sep. 2019.
[32] X.-H. Chang and G.-H. Yang, "Nonfragile H_∞ filtering of continuous-time fuzzy systems," *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1528–1538, Apr. 2011.
[33] M. Craven, D. Freitag, A. Mccallum, T. Mitchell, K. Nigam, and C. Quek, "Learning to extract symbolic knowledge from the world wide Web," in *Proc. AAAAI*, 1997, pp. 509–516.
[34] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011.
[35] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, May 2010.
[36] O. J. Dunn, "Multiple comparisons among means," *J. Amer. Stat. Assoc.*, vol. 56, no. 293, pp. 52–64, Mar. 1961.

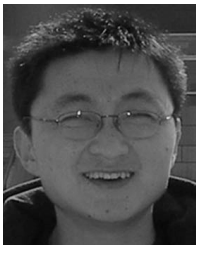


TIANXIANG LUAN received the B.S. degree, in 2012. He is currently pursuing the master's degree with the National University of Defense Technology, Changsha, China. His research interests include data mining and machine learning.

TINGJIN LUO received the B.S., master's, and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 2011, 2013, and 2018, respectively. From 2015 to 2017, he was a Visiting Ph.D. Student with the University of Michigan, Ann Arbor, MI, USA. He is currently a Lecturer with the College of Science, National University of Defense Technology. He has authored more than 15 articles in journals and conferences, such as the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and *KDD*. His research interests include machine learning, multimedia analysis, optimization, and computer vision. He has been a Program Committee Member of several conferences, including IJCAI, AAAI, and so on.



WENZHANG ZHUGE received the B.S. degree from Shandong University, Jinan, China, in 2015, and the M.S. degree from the National University of Defense Technology, Changsha, China, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include machine learning, system science, and data mining.



CHENPING HOU (Member, IEEE) received the Ph.D. degree from the National University of Defense Technology, Changsha, China, in 2009. He is currently a Full Professor with the Department of Systems Science, National University of Defense Technology. He has authored over 80 peer-reviewed articles in journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS/TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS/*TCB*, the IEEE TRANSACTIONS ON IMAGE PROCESSING, *IJCAI*, and *AAAI*. His current research interests include machine learning, data mining, and computer vision.