Pattern Recognition 126 (2022) 108583

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Bag dissimilarity regularized multi-instance learning

Shiluo Huang[a], Zheng Liu[a], Wei Jin[a,b,*], Ying Mu[a,*]

[a] Research Center for Analytical Instrumentation, Institute of Cyber-systems and Control, State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310058, P. R. China
[b] Huzhou Institute of Zhejiang University, Huzhou 313299, P. R. China

ARTICLE INFO

Article history:
Received 6 August 2021
Revised 17 December 2021
Accepted 8 February 2022
Available online 10 February 2022

Keywords:
Multi-instance learning (MIL)
Dissimilarity regularization
Fisher score

ABSTRACT

Multi-instance learning (MIL) is able to cope with the weakly supervised problems where the training data is represented by labeled bags consisting of multiple unlabeled instances. Due to its practical significance, MIL has recently drawn increasing attention. Introducing bag representations is an attractive way to learn MIL data. However, it is difficult for the existing MIL methods to utilize both implicit and explicit bag representations simultaneously. In this paper, we propose a bag dissimilarity regularized (BDR) framework that incorporates multiple bag representations regardless of explicitness or implicitness. Here, the implicit bag representations are incorporated into a regularization term that contains the intrinsic geometric information provided by the bag dissimilarities. The regularization term can be added to the objective function of supervised classifiers. An effective method for explicit bag embedding is also proposed, which exploits the Fisher score derived from factor analysis. Finally, we propose two specific BDR methods based on support vector machine and broad learning system. The proposed BDR methods are evaluated on 14 datasets, and have achieved competitive results with limited computation consumption. We also discuss the effectiveness and the characteristics of BDR framework.

1. Introduction

In many real-world tasks such as drug activity prediction [1], the data is organized as bags consisting of multiple instances, where only the labels of bags are available. The lack of instance labels makes it difficult to build satisfactory models using regular machine learning techniques. To cope with these weakly supervised problems, multi-instance learning (MIL) is proposed. In the MIL problems, the bags are usually divided into two classes: positive bags and negative bags. MIL is chiefly concerned with the binary classification problem of predicting the bag labels. Besides drug activity prediction, MIL has been applied in various fields like text categorization [2], object detection [3], remote sensing [4], saliency detection [5], and medical diagnosis [6].

In the previous studies [7], most existing MIL methods are generally divided into three categories according to their working space, namely instance space (IS), bag space (BS), and embedded space (ES). IS methods assume that the labels can be estimated by using an instance level classifier. IS methods first predict the scores of each instances using the instance level classifier, and then obtain the bag labels by aggregating the instance scores. Multi-instance support vector machines (mi-SVM, MI-SVM) [8], single instance learning methods (SIL) [9], and multi-instance network (mi-Net) [10] are the typical IS methods. BS methods treat each bag as an entity, and utilize distance functions or kernel functions to evaluate the similarity between two bags. The bag similarities can be incorporated into distance-based or kernel-based classifiers. MI-Graph and miGraph [2] define graph kernels for distinguishing the MIL bags. In [11], the bags are regarded as point sets or distributions, and several bag dissimilarity functions are proposed. ES methods explicitly map the bags into representation vectors that retain the essential information. The representation vectors form an embedded space where the classifiers are trained. Chen et al. [12] project the bags into an embedded space defined by the training instances using a similarity measure. Wei et al. [13] introduce the vector of locally aggregated descriptors (VLAD) and Fisher vector (FV) for bag representation. Deep neural networks [14] are also utilized to learn an effective mapping function.

The bag level methods, namely BS and ES methods, are usually more attractive because of their ability to provide bag representations. The bag representations make it possible to utilize the abundant regular machine learning techniques for classifier training. But for BS and ES methods, there is usually an inevitable information loss when extracting the bag representations. And it seems that the way to extract bag representations plays a major role in determining the performance of BS and ES methods [7]. Since dif-

* Corresponding authors.
E-mail addresses: jinweimy@zju.edu.cn (W. Jin), muying@zju.edu.cn (Y. Mu).

https://doi.org/10.1016/j.patcog.2022.108583
0031-3203/

ferent representations contain different information, utilizing multiple bag representations simultaneously might alleviate the information loss and thus might refine the performance.

Existing studies have shown that using multiple bag representations could boost the MIL performance. In [13], the representation vectors of miVLAD and miFV are directly concatenated to form new vectors (miV&F). The miV&F outperforms the comparison methods on several datasets. Both the BS and the ES methods try to extract the intrinsic bag representations. As the BS methods extract representations in a different way from the ES methods, the implicit representations of BS methods might be complementary to the explicit representations of ES methods, and vice versa. With fixed representation quality, greater diversity of information is likely to improve the overall performance. Thus, utilizing the representation provided by two kinds of methods simultaneously might be another way of performance refinement, besides pursuing high representation quality. However, the BS methods extract the bag representation implicitly using distance or kernel functions, while the ES methods explicitly extract the representation by projecting the bags into the embedded spaces. This difference makes it difficult to combine the representations of BS methods with those of ES methods.

Using dissimilarity feature vectors [11] seems to be a solution to the problem. The dissimilarity vector provides an explicit representation of distance functions. However, combining the representation vectors of ES methods with the dissimilarity vectors might result in high dimensional vectors, which is likely to cause performance degradation instead (Hughes phenomenon [15]). And it is also difficult to adjust the impact of each representation according to the data property. In the previous study [16], the low rank representation approaches could be guided by specific regularization terms. Thus, introducing regularization terms might be another way of incorporating the implicit representation.

In this paper, we propose a bag dissimilarity regularized (BDR) framework to combine the bag representations of ES methods with those of BS methods. The implicit representations extracted by BS methods are incorporated into a regularization term, which could be added to the objective function of supervised classifiers. After adding the regularization term, the classifier tends to classify the bags that are close in the bag space into the same class. By tuning the weight of regularization term, the influence of bag dissimilarities can be easily adjusted. Along with BDR framework, we also propose an ES method based on the Fisher score (FS) derived from factor analysis, namely multi-instance factor analysis (MIFA). Factor analysis is an effective method for modeling the covariance structure of high dimensional data. Using factor analysis as the generative model could effectively retain the discriminative information within MIL data. Finally, we incorporate the regularization term into two widely used classifiers: support vector machine (SVM) and broad learning system (BLS), proposing two BDR classifiers (BDR-SVM and BDR-BLS). In practice, the proposed BDR method could enable an instant improvement of MIL performance by utilizing multiple bag representations simultaneously. And the BDR methods could adapt to various practical tasks with the assistance of different bag representations. The major contributions of this work can be concluded as follows.

1. We propose a bag dissimilarity regularized framework that can utilize multiple bag representations regardless of whether the representations are explicit or implicit.
2. An effective explicit bag representation (MIFA) is proposed, which utilizes the FS derived from factor analysis.
3. Two widely used classifiers are transformed into BDR classifiers (BDR-SVM and BDR-BLS).

The rest of this paper is organized as follows. Section 2 presents the related work. In Section 3, we introduce the proposed method. Experiments are presented in Section 4. Section 5 gives the conclusion.

2. Related work

In this section, the existing MIL methods are reviewed. For MIL problems, the samples are organized as bags consisting of multiple instances. Let $B_i = \{x_{i1}, x_{i2}, \cdots, x_{in_i}\}$ denote a bag containing n_i instances, while each instance x_{ij} is represented by a feature vector with n_f attributes. The labels of training data are represented by $Y = \{y_1, y_2, \cdots, y_{n_b^l}\}$, where n_b^l is the number of labeled bags. In most MIL tasks, we have $y_i \in \{-1, +1\}$, where $y_i = +1$ represents a positive bag B_i and $y_j = -1$ represents a negative bag B_j. As mentioned above, the MIL methods can be broadly divided into IS, BS, and ES methods. In the following part, these three categories will be reviewed and discussed.

IS methods try to train an instance level classifier and obtain the bag labels by aggregating the instance scores. Because of the ambiguous labels, IS methods rely on assumptions to build the instance classifier and to aggregate the instance scores. The standard multi-instance assumption (SMI) [17] and the collective assumption [7] are two typical assumptions used by IS methods.

SMI states that a positive bag contains at least one positive instance while a negative bag only contains negative instances. Axis parallel rectangle (APR) [1] is one of the classical IS methods following the SMI assumption. APR tries to maximize the number of positive bags that contain at least one positive instance, and the number of negative bags that do not contain any positive instance. MI-SVM and mi-SVM [8] are another two IS methods that follow SMI assumption. Mi-SVM iteratively adjusts the labels of training instances within the positive bags according to their distance to the current hyper plane, while MI-SVM utilizes the selected witness instances for training. Multi-instance representative SVM (MIRSVM) [18] further extends the idea of MI-SVM, and the representative instances within the bags are selected to train the SVM. Diverse density (DD) [19] evaluates the likelihood that an instance is a positive instance. Expectation maximization (EM) algorithm [20] is usually used for optimizing DD (EM-DD). Deep neural networks that follow SMI assumption have also been proposed for MIL, such as mi-Net [10]. Several SMI-based methods (APR, EM-DD, mi-SVM) have been utilized to classify the multi-level features for salience detection [5].

Collective assumption states that all the instances within a bag contribute equally to the label of bag, which means that all the instances within positive bags should be involved. Single instance learning (SIL) [9] is one of the methods following this assumption, which trains a standard classifier using the all the instances within the training set. The wrapper for MIL (MIWrapper) [21] also discards the SMI and introduces weights to distinguish the instances from different bags.

BS methods define distance functions or kernel functions to implicitly extract the bag representations. The defined functions are then processed by distance-based or kernel-based classifiers. As the classifiers work on bag level, there is no need for introducing assumptions like SMI to cope with the ambiguous labels.

Usually, the distance-based and the kernel-based classifiers are used to utilize the implicit bag representations. Three K nearest neighbor (kNN) classifiers, including regular kNN, Bayesian kNN, and Citation kNN, are tested in [22], while the minimal and the maximal Hausdorff distances [23] are used for similarity estimation. An assumption is implicitly used by the kNN-based methods, which states that the similarity between two bags with the same labels is larger than the similarity between bags with different labels. As for kernel functions, the kernel function of bags can be defined as the sum of instance kernels or a kernel of representa-

tion vectors [24]. Zhou et al. [2] utilize graph kernels to represent the bags, where the bags are transformed into undirected graphs implicitly or explicitly. Inspired by the metric learning techniques, some studies also utilize deep neural networks to learn an effective similarity metric implicitly [25] or explicitly [26].

MInD [11] transforms the bag similarities or dissimilarities into regular dissimilarity vectors, and then trains the classifier in this dissimilarity space. The representation vectors in the dissimilarity space can be learned by the regular classifiers instead of the certain classifiers. In the dissimilarity space, the non-metric dissimilarity measure can be defined as the distance function, which enables the adjustment of distance function according to real-world tasks. Several metric and non-metric dissimilarities are presented in [11], which are given as follows.

$$\begin{aligned} D_{maxmin}(B_i, B_j) &= \max_m \min_n d(x_{im}, x_{jn}), \\ D_{meanmin}(B_i, B_j) &= \frac{1}{n_i}\sum_m^{n_i} \min_n d(x_{im}, x_{jn}), \\ D_{minmin}(B_i, B_j) &= \min_m \min_n d(x_{im}, x_{jn}), \\ D_{meanmean}(B_i, B_j) &= \frac{1}{n_i n_j}\sum_m^{n_i}\sum_n^{n_j} d(x_{im}, x_{jn}), \end{aligned} \tag{1}$$

where $d(x_{im}, x_{jn})$ represents the Euclidean distance between x_{im} and x_{jn}. Based on various similarities or dissimilarities [27], MInD achieves competitive performance with the straightforward representation. And in many practical tasks, using multiple (dis)similarities could improve the bag level accuracy. However, the dimension of dissimilarity representation is equal to the number of prototypes. When the number of prototypes is large, MInD becomes computation consuming. The high dimensional representation vectors might also result in performance degradation due to Hughes phenomenon.

ES methods explicitly extract the bag representation using mapping functions that project the bags into embedded spaces. Therefore, the MIL problem has been transformed into a regular supervised problem. The regular supervised classifiers can easily learn the representation vectors in the embedded space without specific restrictions. One of the major problems for ES methods is how to extract effective bag representation.

A simple multi-instance method (Simple MI) [28] represents a bag with the average of instances within the bag. Obviously, Simple MI performs well only if the positive and the negative bags have different averages from each other. Simple MI will fail to retain the discriminative information under complex conditions. For example, the average of positive bags becomes similar to that of negative bags when the witness ratio is low. To extract more effective representation, Chen et al. [12] propose multi-instance learning via embedded instance selection (MILES). MILES assumes that each instance in the training bags is a candidate for the target concepts which can be related to either positive or negative bags. The bags are represented by the similarities to the concepts, and a 1-norm SVM is used to identify the most discriminative concept.

Using VLAD [29] and FV [30] representation is another way to retain the key information. VLAD represents the bags with the dissimilarities between the instances and their corresponding centroids. The instances are first clustered into several centroids using k-means algorithm, and each instance is assigned to its nearest centroid. The centroids are treated as concepts, and their dissimilarities to the corresponding instances form the bag representation. The VLAD representation can be viewed as a simplified FV representation [13]. FV is derived from normalizing the Fisher score (FS) using the square root of $\mathcal{F}^{-1}$, where $\mathcal{F}$ is the Fisher information matrix (FIM) [31]. Assuming that the instances $X = \{x_1, x_2, \cdots, x_{n_i}\}$ are sampled from a generative model $\pi(X;\theta)$ with m parameters $\theta = \{\theta_1, \theta_2, \cdots, \theta_m\}$, the FS is defined as the gradient of log-likelihood with respect to θ:

$$G_\theta(X) = \nabla_\theta \log \pi(X;\theta). \tag{2}$$

FIM can be viewed as the covariance matrix of FS, and is calculated as

$$\mathcal{F} = E[G_\theta(X)G_\theta(X)^T]. \tag{3}$$

FS describes the contribution of parameters to the whole generative process, and FV is a normalized form of FS. In [30], Gaussian Mixture Model (GMM) is used as the generative model for deriving the FV of images. However, the number of available samples is relatively small in many MIL tasks. The lack of modeling samples along with high dimensional instances makes it difficult for GMM to model the distribution of instances, especially on small-scale MIL datasets. As the rise of deep learning, several deep neural networks have been utilized to learn a bag representation, such as MI-Net [14] and attention based network (Att. Net) [32].

Some studies have indicated that utilizing multiple bag representations simultaneously can boost the performance. For example, utilizing the hybrid of VLAD and FV can improve the accuracy on certain datasets [13]. In [11], generating the dissimilarity vectors using multiple distance functions could provide better results. Therefore, a framework that incorporates both the implicit and the explicit bag representations would be attractive.

3. Proposed method

In this section, we propose a bag dissimilarity regularized (BDR) framework that can utilize the implicit and the explicit representations simultaneously. An effective explicit bag representation based on FS is also proposed. Finally, two BDR classifiers are presented.

3.1. Bag dissimilarity regularization

In the BDR framework, the information within implicit representation is incorporated into a regularization term. Extracting the implicit representation can be viewed as implicitly mapping the bags into an embedded space where the distances between bags satisfy the dissimilarity function $d(B_i, B_j)$. With a little concession, we assume that the bags lie on a low-dimensional sub-manifold in the implicit embedded space, and the bags with the same labels are close on the sub-manifold. This assumption is similar to the manifold assumption [33], which plays an essential role in many machine learning techniques such as semi-supervised learning [34] and dimension reduction [35]. The BDR framework is shown in Fig. 1. The MIL data is transformed into explicit bag representation via ES methods, forming the input vectors. Meanwhile, the implicit bag representation of MIL data is utilized to construct a k-nearest neighbor graph (kNNG), and then exploited by the BDR classifier.

According to the previous studies [36], a nearest neighbor graph of the mapped data can effectively model the local geometric structure [37]. Suppose that there is a kNNG in the implicit embedded space, with each vertex corresponding to a bag. For a single bag B_i, it is connected with its k nearest neighbors measured by the dissimilarity function. We define a 0–1 weight matrix S on the graph, which is calculated as follows.

$$S_{i,j} = \begin{cases} 1, & \text{if } B_j \text{ is among the k nearest neighbors of } B_i. \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

$S_{i,j}$ reflects the dissimilarity between bag B_i and B_j evaluated by $d(B_i, B_j)$.

In the output space, we use the square of Euclidean distance to measure the similarity of output vectors $\hat{Y} = \{\hat{y}_1; \hat{y}_2; \cdots; \hat{y}_{n_b}\}$, where $\hat{y}_i$ is the output vector of B_i and n_b is the number of bags.

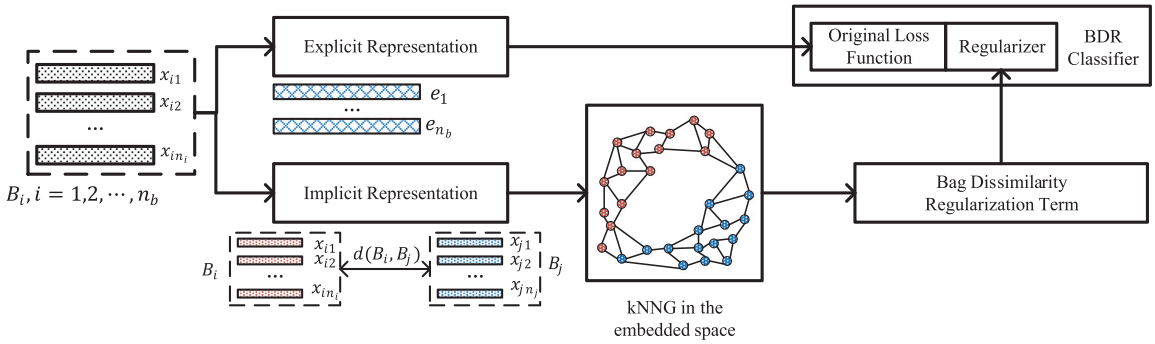

Fig. 1. The bag dissimilarity regularized framework.

For a classifier $f(\cdot)$, the output vector $\hat{y}_i$ is derived from the explicit representation of B_i, with $\hat{y}_i = f(e_i)$. e_i denotes the explicit representation of B_i. Based on the $n_b \times n_b$ weight matrix S, the following regularization term is utilized to ensure that the output vectors are constrained by the local manifold structure in the embedded space

$$\mathcal{R}_{BD} = \frac{1}{2}\sum_i \sum_j S_{i,j} \|\hat{y}_i - \hat{y}_j\|^2. \tag{5}$$

By minimizing $\mathcal{R}_{BD}$, the bags that are close on the manifold tend to be close in the output space. Therefore, the output results are regularized by the implicit bag representation defined by $d(B_i, B_j)$. The bag dissimilarity regularization term can be rewritten as follows.

$$\mathcal{R}_{BD} = \mathrm{Tr}(\hat{Y}^T L \hat{Y}), \tag{6}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix, and L is the Laplacian matrix of S. $L = D - S$, and D is a diagonal matrix that satisfies $D_{i,i} = \sum_j S_{i,j}$.

When the number of bags is adequate, we can use a set of anchor bags $\mathcal{B}_A = \{B_1, B_2, \cdots, B_{n_a}\}$ to approximate the sub-manifold, and embed the other bags in this structure. Here, n_a is the number of anchor bags. The local structure of each bag is approximated by the nearest anchor bags, which can be modeled through a $n_b \times n_a$ weight matrix S^a. S^a is a 0–1 weight matrix and is defined as follows.

$$S^a_{i,j} = \begin{cases} 1, & \text{if } B_j \in \mathcal{B}_A \text{ is among the k nearest anchor bags of } B_i. \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The $S^a_{i,j}$ reflects the position of B_i on the approximated sub-manifold.

To constrain the output vectors, another regularization term $\mathcal{R}_{BDA}$ is proposed. Minimizing $\mathcal{R}_{BDA}$ will make the output vectors become consistent with the data distribution on the manifold. $\mathcal{R}_{BDA}$ is calculated as follows:

$$\mathcal{R}_{BDA} = \frac{1}{2}\sum_i \sum_{j, j\in\mathcal{B}_A} S^a_{i,j} \|\hat{y}_i - \hat{y}_j\|^2. \tag{8}$$

If we only consider the anchor bags, $\mathcal{R}_{BDA}$ is very similar to $\mathcal{R}_{BD}$. By minimizing $\mathcal{R}_{BDA}$, the anchor bags that are close on the manifold tend to be close in the output space. For the rest bags, minimizing $\mathcal{R}_{BDA}$ could make the output vectors of these bags retain the relevance to the neighboring anchor bags on the manifold. In the previous studies focusing on other fields, some techniques are similar to $\mathcal{R}_{BDA}$ in approximating approach. In [38], for example, a landmark-based map is used for the graph construction of spectral clustering, where each data point is approximated by the combination of landmarks. And $\mathcal{R}_{BDA}$ can be rewritten as follows.

$$\mathcal{R}_{BDA} = \mathrm{Tr}\left(\frac{1}{2}\hat{Y}^T D^1 \hat{Y} + \frac{1}{2}\hat{Y_a}^T D^2 \hat{Y_a} - \hat{Y}^T S^a \hat{Y_a}\right), \tag{9}$$

where $\hat{Y_a} = \{\hat{y}_i; i \in \mathcal{B}_A\}$ denotes output vectors of anchor bags. D^1 and D^2 are two diagnoal matrices that satisfy $D^1_{i,i} = \sum_j S_{i,j}$ and $D^2_{j,j} = \sum_i S_{i,j}$. $\mathcal{R}_{BD}$ can be viewed as a special form of $\mathcal{R}_{BDA}$: $\mathcal{R}_{BDA}$ is equal to $\mathcal{R}_{BD}$ when all the bags are defined as anchor bags.

The BDR framework can utilize the intrinsic information provided by the implicit bag representations. It can be seen that $\mathcal{R}_{BD(A)}$ does not rely on the label information. To utilize the label information, we incorporate this regularization term into the objective function of a regular supervised method. The objective function can be formulated as follows.

$$\mathcal{O} = \mathcal{L}(Y, \hat{Y}_l) + \frac{\lambda_1}{2}\mathcal{R}(f) + \frac{\lambda_2}{2}\mathcal{R}_{BD(A)}, \tag{10}$$

where $\mathcal{L}(\cdot)$ represents a loss function, and $\mathcal{R}(\cdot)$ is a regularization function. $Y = \{y_1, y_2, \cdots, y_{n^l_b}\}$ denotes the labels, and $\hat{Y}_l = f(E_l)$ denotes the output vectors of the labeled bags. $E_l = \{e_1, e_2, \cdots, e_{n^l_b}\}$ is the explicit representation of the labeled bags. λ_1 and λ_2 are the hyper-parameters that control the impact of regularization terms. It should be noted that the explicit representation has already been incorporated into the output vectors ($\hat{Y}$ and $\hat{Y}_l$). As shown in Fig. 1, the feature vectors given by explicit representation serve as the input of classifiers. In fact, many regular classifiers are not able to utilize the implicit representation. If the regularization term $\mathcal{R}_{BD(A)}$ is removed, the rest part is equal to the backend classifier of explicit methods. And $\mathcal{R}_{BD(A)}$ can be viewed as the carrier of implicit representation.

For further refinement, the unlabeled bags can also be included in the BDR framework as $\mathcal{R}_{BD(A)}$ does not rely on label information. The local geometric structure in the implicit embedded space can be modeled more effectively by introducing the unlabeled bags, especially when the number of labeled bags is limited. That makes BDR method a semi-supervised MIL method, which is able to exploit the abundant unlabeled bags.

Apart from the implicit bag representations, the explicit bag representations can also be incorporated into the regularization term. A kNNG defined in the embedded space captures the local geometric information effectively. In the embedded space defined by an explicit bag representation, the distance measures like Euclidean distance can be used to evaluate the dissimilarities between bags. Incorporating extra explicit knowledge can enable the BDR methods to utilize diverse explicit representation, potentially refining the performance. The explicit representation might be complementary to not only implicit representation but also extra explicit representation.

3.2. Multi-instance factor analysis

Since the BDR framework relies on both the implicit and the explicit bag representations, an effective explicit bag representation matters a lot to the BDR methods. As mentioned above, GMM might suffer performance degradation on the MIL dataset due to the limited instances. One of the solutions to this problem is to introduce a more suitable generative model. In this section, we propose a FS-based method that uses factor analysis (FA) as the generative model, which is called MIFA.

FA is a method that forms the lower dimensional probabilistic representation of data [39]. Specifically, it models the covariance structure of high dimensional instances x using a lower dimensional latent variable or factor z. The generative model is given as:

$$x = \Lambda z + \mu + \epsilon, \tag{11}$$

where Λ is a factor loading matrix, and μ is a bias vector. ϵ represents the noise. The factor z follows the Gaussian distribution $\mathcal{N}(z; 0, I)$, and the noise ϵ follows $\mathcal{N}(\epsilon; 0, \Psi)$. FA assumes that the high dimensional instances lie close to a low dimensional linear subspace and can be approximated by the low dimensional factor z. The instances might deviate from the linear subspace, and thus a Gaussian noise ϵ is utilized to model this discrepancy.

After modeling the instances, we have to map the bags with multiple instances into representation vectors with fixed length. In practice, FS is a widely used method to extract fixed length representations from the generative models [40]. The dimensionality of FS is determined by the parameters of generative model and the dimensionality of instances. As shown in Eq. 2, FS is defined as the gradient of log likelihood with respect to the parameters of generative model.

Recently, Dixit et al. [41] have proven that the gradient of Q function in the EM algorithm is equal to the gradient of log likelihood when the two gradients are calculated using the same background model. In other words, the FS of a background model $\pi(\cdot, \theta)$ is equal to the gradient of Q function derived from $\pi(\cdot, \theta)$. As abundant studies on the EM algorithm of FA have been proposed [42], the derivation of FS from FA model can be simplified. The Q function is calculated as

$$\begin{aligned} Q &= E\left[\sum_i \log p(x_i, z_i; \theta)\right] \\ &= E\left[\sum_i \log p(x_i|z_i; \theta) + \log p(z_i; \theta)\right] \\ &= E\left[\sum_i \log \mathcal{N}(x_i; \Lambda z_i + \mu, \Psi) + \log \mathcal{N}(z_i; 0, I)\right]. \end{aligned} \tag{12}$$

In the E-step of EM algorithm, the expectation of latent factors is calculated. The factor z_i which corresponds to x_i is calculated as

$$E(z_i) = \beta(x_i - \mu), \tag{13}$$

where $\beta = \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}$. The FS with respect to the bias vector μ can be derived by the gradient of Q function with respect to μ. For a single instance x_i, FS with respect to μ is calculated as

$$\begin{aligned} G_\mu(x_i) &= \nabla_\mu Q \\ &= E(\Psi^{-1}(x_i - \mu - \Lambda z_i)) \\ &= \Psi^{-1}(x_i - \mu - \Lambda E(z_i)). \end{aligned} \tag{14}$$

Similarly, the FS with respect to factor loading matrix Λ can be derived from the gradient of Q function with respect to Λ

$$\begin{aligned} G_\Lambda(x_i) &= \nabla_\Lambda Q \\ &= E(\Psi^{-1}(x_i - \mu - \Lambda z_i) z_i^T) \\ &= \Psi^{-1}(x_i - \mu - \Lambda E(z_i)) E(z_i^T). \end{aligned} \tag{15}$$

The Eq. 15 is an approximate Fisher score where we assume that $E(z_i z_i^T) \approx E(z_i)E(z_i^T)$. This approximate representation retains the second order statics, and could effectively capture the information within MIL bags.

Inspired by [30] where the representation of each image is derived from accumulating the FS of descriptors within the image, we define that the representation of a bag is the sum of the FS deriving from the instances within this bag. However, the bag size is usually uncertain. To alleviate the influence of varied bag size, we normalize the FS by the number of instances within the bag. The FS of bags is given as

$$G_\mu(B_i) = \frac{1}{n_i} \sum_{x_j \in B_i} G_\mu(x_j), \tag{16}$$

$$G_\Lambda(B_i) = \frac{1}{n_i} \sum_{x_j \in B_i} G_\Lambda(x_j), \tag{17}$$

where n_i is the number of instances within B_i.

It should be noted that $G_\Lambda(B_i)$ is a $n_f \times n_d$ matrix, where n_d is the dimension of linear subspace and n_f is the number of instance attributes. To transform $G_\Lambda(B_i)$ into a representation vector that can be utilized by the regular classifiers, we can directly flatten the matrix or using certain processing methods. The flattened $G_\Lambda(B_i)$ can retain most of the gradient information with respect to Λ. However, the dimensionality of representation vector can become very high when n_f or n_b is large. We can use the singular values of original $G_\Lambda(B_i)$ to replace the flattened vector.

$$G_\Lambda^S(B_i) = \mathcal{S}(G_\Lambda(B_i)), \tag{18}$$

where $\mathcal{S}(\cdot)$ denotes the singular values of a matrix. The dimensionality of singular value vector is only n_d, and the singular values can still retain some useful information.

Finally, the explicit bag representation $\mathcal{E}(B_i)$ is obtained by concatenating $G_\mu(B_i)$ and $G_\Lambda^{(S)}(B_i)$. $\mathcal{E}(B_i)$ is a $n_f \times (n_d + 1)$ vector when $G_\Lambda(B_i)$ is used. The dimensionality of $\mathcal{E}(B_i)$ will be reduced to $n_f + n_d$ when $G_\Lambda^S(B_i)$ is used.

3.3. Bag dissimilarity regularized classifiers

In this section, we incorporate the bag dissimilarity regularization term into two supervised classifiers: broad learning system and support vector machine. The two BDR classifiers, namely BDR-BLS and BDR-SVM, are presented in the following part.

3.3.1. BDR-BLS

Broad learning system (BLS) is an effective neural network proposed by Chen et al. [43]. BLS has been proven effective in various fields like remote sensing [44], fault diagnosis [45], motor learning [46], and spammer detection [47]. BLS consists of feature nodes Z, enhancement nodes H, and output weights W. Suppose that there are n_1 feature mappings, n_e feature nodes, and n_h enhancement nodes. Let $Y = \{y_1, y_2, \cdots, y_{n_b^l}\}$ denote the bag labels, where n_b^l is the number of labeled bags. The explicit representation vectors of the labeled bags are denoted as E_l. The feature nodes $\{Z_i = \phi(E_l W_{ei} + b_{ei}); i = 1, 2, \cdots, n_1\}$ are obtained by projecting the data with random weights W_{ei}, where $\phi(\cdot)$ is a linear function. The collection of feature nodes is given as $Z = [Z_1, \cdots, Z_{n_1}] \in \mathbb{R}^{n_b^l \times n_e}$. The enhancement nodes is calculated as $H = \zeta(ZW_h + b_h) \in \mathbb{R}^{n_b^l \times n_h}$, where $\zeta(\cdot)$ is a non-linear function and W_h is a random projection matrix. b_{ei} and b_h are the random biases. The objective function of BLS is given as

$$\mathcal{O}_{BLS} = \|AW - Y\|^2 + \frac{\lambda_1}{2}\|W\|^2, \tag{19}$$

where $A = [Z|H] \in \mathbb{R}^{n_b^l \times (n_e+n_h)}$ is the pattern matrix of input data. Because $\mathcal{O}_{BLS}$ is convex, the output weights can be solved as

$$W = (\lambda_1 I + A^T A)^{-1} A^T Y, \tag{20}$$

where I is an identity matrix. The output vectors are calculated as $\hat{Y} = AW$.

As $\mathcal{R}_{BD}$ is a special form of $\mathcal{R}_{BDA}$, $\mathcal{R}_{BDA}$ is discussed in this section. By incorporating the regularization term into $\mathcal{O}_{BLS}$, the geometric information of another bag representation is included. As unlabeled bags might be included, a diagonal weight matrix C is introduced. We assume that $A = [A_l; A_u]$, where $A_l \in \mathbb{R}^{n_b^l \times (n_e+n_h)}$ is the pattern matrix of labeled bags and $A_u \in \mathbb{R}^{n_b^u \times (n_e+n_h)}$ is the pattern matrix of unlabeled bags. n_b^u is the number of unlabeled bags. C is a $n_b \times n_b$ diagonal matrix with $\{C_{i,i} = 1; i = 1, \cdots, n_b^l\}$ and $\{C_{i,i} = 0; i = n_b^l + 1, \cdots, n_b^l + n_b^u\}$. The objective function of bag dissimilarity regularized BLS (BDR-BLS) is

$$\begin{aligned} \mathcal{O}_{BDR-BLS} &= \|C^{1/2}(AW - \tilde{Y})\|^2 + \frac{\lambda_1}{2}\|W\|^2 + \frac{\lambda_2}{2}\mathcal{R}_{BDA} \\ &= \|C^{1/2}(AW - \tilde{Y})\|^2 + \frac{\lambda_1}{2}\|W\|^2 + \frac{\lambda_2}{2}\mathrm{Tr}(W^T \tilde{L} W), \end{aligned} \tag{21}$$

where $\tilde{L} = \frac{1}{2}A^T D^1 A + \frac{1}{2}A_a^T D^2 A_a - A^T S^a A_a$, and $A_a \in \mathbb{R}^{n_a \times (n_e+n_h)}$ represents the pattern matrix generated by the anchor bags. $\tilde{Y} = \{Y; 0\} \in \mathbb{R}^{n_b \times 1}$ is the training target. It can be found that $\mathcal{R}_{BDA}$ is convex, and thus $\mathcal{O}_{BDR-BLS}$ is a convex function with respect to W. Similarly, W can be solved with ridge regression, which is given as

$$W = (\lambda_1 I + \lambda_2 \tilde{L} + A^T C A)^{-1} A^T C \tilde{Y}. \tag{22}$$

3.3.2. BDR-SVM

As a well-known supervised classifier, support vector machine (SVM) has been empirically proven to have satisfactory performance on various real-world tasks including the MIL problems [48]. In many previous studies related to MIL, like MInD [11] and miFV [13], SVM is used as the standard classifier to deal with the extracted bag representation. Thus, extending the BDR method to SVM would be attractive. Standard SVM tries to find an optimal hyperplane with the constraint of supervision and structural risk. The objective function of SVM with soft margin can be written as

$$\begin{aligned} &\mathcal{O}_{SVM} = \frac{1}{2}\boldsymbol{\alpha}^T K \boldsymbol{\alpha} + C\sum_i^{n_b^l} \xi_i \\ &s.t. \quad y_i\left(\sum_j^{n_b^l} \alpha_i K_{i,j} + b\right) \geq 1 - \xi_i, i = 1, 2, \cdots, n_b^l \\ &\qquad \xi_i \geq 0, i = 1, 2, \cdots, n_b^l \end{aligned} \tag{23}$$

where K is the kernel matrix, and b denotes the bias term. $K_{i,j} = <\phi(e_i), \phi(e_j)>$ is the kernel function of representation pair e_i and e_j, where $\phi(\cdot)$ is a linear or non-linear function. $\boldsymbol{\alpha} = \{\alpha_1; \cdots; \alpha_{n_b^l}\}$ reflects the contribution of each sample (bag) to the prediction results. For a single explicit representation e_i, the output vector can be calculated as follows.

$$\hat{Y}_{e_i} = \boldsymbol{\alpha}^T k_{e_i} + b, \tag{24}$$

where k_{e_i} denotes the vector of kernel function values. Only the training samples with $\alpha_i \geq 0$, namely support vectors, have an influence on the prediction results. Training SVM needs to find the optimal $\boldsymbol{\alpha}$, and the training process can be expressed as a minimization of dual quadratic programming (QP) problem, which is shown as follows.

$$\begin{aligned} &\min_\beta \frac{1}{2}\boldsymbol{\beta}^T Y K Y \boldsymbol{\beta} - \sum_i^{n_b^l} \beta_i \\ &s.t. \quad \sum_i^{n_b^l} \beta_i y_i = 0 \\ &\qquad 0 \leq \beta_i \leq C, i = 1, 2, \cdots, n_b^l \end{aligned} \tag{25}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{n_b^l \times 1}$ is the vector of Lagrange multipliers, and $Y \in \mathbb{R}^{n_b^l \times n_b^l}$ is a diagonal matrix satisfying $Y = diag(y_1, y_2, \cdots, y_{n_b^l})$. The multipliers satisfy $\boldsymbol{\alpha} = Y\boldsymbol{\beta}$. The Eq. 25 can be solved by various methods such as sequential minimal optimization (SMO) [49].

Similarly, we majorly discuss $\mathcal{R}_{BDA}$, while $\mathcal{R}_{BD}$ can be viewed as a special form of $\mathcal{R}_{BDA}$. After adding the regularization term, the objective function of BDR-SVM can be written as

$$\begin{aligned} &\mathcal{O}_{BDR-SVM} = \frac{1}{2}\boldsymbol{\alpha}^T K \boldsymbol{\alpha} + C\sum_i^{n_b^l} \xi_i + \frac{\lambda_2}{2}\boldsymbol{\alpha}^T \tilde{L} \boldsymbol{\alpha} \\ &s.t. \quad y_i\left(\sum_j^{n_b} \alpha_i K_{i,j} + b\right) \geq 1 - \xi_i, i = 1, 2, \cdots, n_b^l \\ &\qquad \xi_i \geq 0, i = 1, 2, \cdots, n_b^l \end{aligned} \tag{26}$$

where $\tilde{L} = \frac{1}{2}K^T D_1 K + \frac{1}{2}K_a^T D_2 K_a - K^T S^a K_a$. Here, $K_a \in \mathbb{R}^{n_a \times n_b}$ is the kernel matrix of anchor bags, while n_a is the number of anchor bags. As mentioned above, $\mathcal{R}_{BDA}$ might introduce the representation of unlabeled bags for classification and thus $\boldsymbol{\alpha} \in \mathbb{R}^{n_b \times 1}$. The prediction result of a representation vector e_i can be written as

$$\hat{Y}_{e_i} = \boldsymbol{\alpha}^T k_{e_i} + b, \tag{27}$$

where $k_{e_i} \in \mathbb{R}^{n_b \times 1}$ is the vector of kernel function values of e_i. To solve Eq. 26, Lagrange multipliers are introduced and the Lagrange function is calculated as

$$\begin{aligned} Lag(\alpha, b, \xi, \beta, \gamma) &= \frac{1}{2}\boldsymbol{\alpha}^T K \boldsymbol{\alpha} + C\sum_i^{n_b^l} \xi_i + \frac{\lambda_2}{2}\boldsymbol{\alpha}^T \tilde{L} \boldsymbol{\alpha} \\ &- \sum_i^{n_b^l}\left[\beta_i\left(y_i\left(\sum_j^{n_b} \alpha_i K_{i,j} + b\right) - 1 + \xi_i\right) + \gamma_i \xi_i\right], \end{aligned} \tag{28}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{(n_b^l \times 1)}$ and $\boldsymbol{\gamma} \in \mathbb{R}^{(n_b^l \times 1)}$ are the Lagrange multipliers. As the gradient of Lagrange function $Lag(\cdot)$ with respect to b, ξ_i, and $\boldsymbol{\alpha}$ is zero, the following auxiliary equations are obtained.

$$\begin{aligned} &\nabla_b Lag = 0 \rightarrow \sum_i^{n_b^l} \beta_i y_i = 0, \\ &\nabla_{\xi_i} Lag = 0 \rightarrow 0 \leq \beta_i \leq C, i = 1, 2, \cdots, n_b^l, \\ &\nabla_{\boldsymbol{\alpha}} Lag = 0 \rightarrow \boldsymbol{\alpha} = \Omega K J Y \boldsymbol{\beta}, \end{aligned} \tag{29}$$

where $J = [I; 0] \in \mathbb{R}^{(n_b) \times (n_b^l)}$ with $I \in \mathbb{R}^{n_b^l \times n_b^l}$, and $\Omega = (K + \lambda_2 \tilde{L})^{-1}$. Substituting the variables in $Lag(\cdot)$ using the above auxiliary equations, the optimization problem can be rewritten as

$$\begin{aligned} &\min_\beta \frac{1}{2}\boldsymbol{\beta}^T Y \tilde{K} Y \boldsymbol{\beta} - \sum_i^{n_b^l} \beta_i \\ &s.t. \quad \sum_i^{n_b^l} \beta_i y_i = 0 \\ &\qquad 0 \leq \beta_i \leq C, i = 1, 2, \cdots, n_b^l \end{aligned} \tag{30}$$

where $\tilde{K} = J^T K^T \Omega^T K J$. This optimization problem can be solved by the standard SVM solvers. After obtaining $\boldsymbol{\beta}$, we can calculate $\boldsymbol{\alpha}$ according to the auxiliary equation.

3.3.3. Discussion

In the above sub-sections, we incorporate the bag dissimilarity regularization into two regular classifiers. It can be found that both BDR-BLS and BDR-SVM can be solved using regular optimization methods. This uniformity makes it easier to implement the BDR methods with the assistance of abundant existing techniques. The modified versions of classifiers can also be incorporated in the BDR framework. In the following part, we majorly discuss the differences and connections between $\mathcal{R}_{BD}$ and $\mathcal{R}_{BDA}$.

Compared with $\mathcal{R}_{BD}$, $\mathcal{R}_{BDA}$ uses a limited number of anchor bags to capture the local geometric information. Intuitively, this results in a performance degradation as some of the information might be lost. But the degradation will be limited as long as the anchor bags can retain the overall structure of manifold. Empirically, a set of randomly selected bags is enough to retain the manifold information when the dataset is large enough. Without loss of generality, the randomly selected bags are used as the anchor bags in this paper.

One of the motivations for $\mathcal{R}_{BDA}$ is to improve the efficiency. $\mathcal{R}_{BD}$ becomes computation consuming when there are too many bags included for training. Assume that there are n_b^l labeled and n_b^u unlabeled bags are included, while $n_b = n_b^l + n_b^u$. We also suppose that the effect of bag number is far greater than that of other factors. Firstly, using $\mathcal{R}_{BDA}$ can reduce the complexity of graph construction. Constructing a regular graph costs $O(n_b^2)$, while the computation complexity of anchor graph is $O(n_b n_a)$. For BDR-BLS with $\mathcal{R}_{BD}$, calculating the output weights W costs $O(n_b^2)$. When $\mathcal{R}_{BDA}$ with n_a anchor bags is used, the complexity of calculating W is $O(n_b n_a)$. While solving BDR-SVM, most computation complexity results from Ω. The computation complexity of Ω reaches $O(n_b^3)$, which is impractical for large-scale data. However, $\mathcal{R}_{BDA}$ is not very effective for BDR-SVM. Although the computation complexity of $\tilde{L}$ can be effectively reduced, the cubic complexity results from computing the inverse of $(K + \lambda_2 \tilde{L})$ have not been alleviated.

In general, $\mathcal{R}_{BDA}$ can reduce the quadratic complexity of BDR-BLS to a linear complexity, which is quite useful for handling large-scale data. The performance degradation caused by $\mathcal{R}_{BDA}$ is also limited, which is proved by the following experiments. Both BDR-BLS and BDR-SVM can be used to classify MIL data with a limited number of bags. When the number of bags is large, BDR-BLS along with $\mathcal{R}_{BDA}$ would be more suitable.

4. Experiments

In this section, various comparison experiments and analysis are presented. The experimental settings and the datasets for evaluation are shown in Section 4.1. A comparison of the proposed method and the existing methods is presented in Section 4.2. Finally, extra experiments are carried out for the analysis of proposed method in the Section 4.3.

4.1. Experimental setting

There are 14 datasets included in our experiments, and the information of these datasets is shown in Table 1. The motivations of datasets include drug activity prediction (Musk), image classification (Elephant, Tiger, Fox, messidor [50]), mutagenicity prediction (Mutagenesis [51]), webpage classification (Web [52]), and textile inspection (Textile). Except for the Textile dataset, all the datasets along with brief documents are available online. Textile dataset contains 79 hyperspectral images (HSI) [53] of the polyester textiles collected from Zhejiang Hengyi Group. The HSIs are captured using an imaging spectrometer produced by Zolix. We aim at detecting the polyester yarn with abnormal dyeing property which will lower the grade of fabrics [54], both automatically and efficiently. The defective yarn is also uneven: a part of the yarn has abnormal dyeing property, while the rest of the yarn is normal. It is difficult to identify which part of the defective yarn is abnormal. The defective yarn samples are labeled as positive, and the normal samples are labeled as negative. Each HSI is regarded as a bag, while the spectral pixels within the HSI are the instances.

Table 1
Details of the MIL datasets used for evaluation.

Dataset	Bag (total)	Bag (positive)	Bag (negative)	Instance	Attribute
Musk1	92	47	45	476	166
Musk2	102	39	63	6598	166
Elephant	200	100	100	1220	230
Fox	200	100	100	1320	230
Tiger	200	100	100	1391	230
Mutagenesis-A	188	125	63	1618	10
Mutagenesis-B	188	125	63	4081	16
Mutagenesis-C	188	125	63	5424	24
Web1	113	21	92	3423	5863
Web2	113	21	92	3423	6519
Web3	113	21	92	3423	6306
Web4	113	89	24	3423	6059
Textile	79	39	40	6876	188
Messidor	1200	654	546	12,352	687

The methods are evaluated using cross-validation, where the bags are split into training and testing bags. In this paper, we use ten-fold cross-validation, which means that around 10% bags are used as testing bags. The cross-validation is repeated for ten times, and the bags are split differently each time, which is in consistence with the previous studies. Experiments are carried out on a laptop with 16GB memory and Core i5 CPU. The tested methods are implemented using Python 3.6. The implementation of BDR-SVM is assisted by the LIBSVM library [48]. The parameters of comparison methods follow the recommended settings in the original papers, if they are provided. Otherwise, the values of hyper-parameters are determined by the results of ten-fold cross-validation.

All the datasets are included in the comparison experiments, where the results of proposed methods and those of other MIL methods are compared. In some of the following experiments, only the benchmark datasets (Musk1, Musk2, Elephant, Fox, Tiger) are included, as they are the five most widely used MIL datasets. For Musk1 and Musk2 datasets, each bag corresponds to a molecule with each instance representing a conformation. The 3 animal datasets (elephant, fox, tiger) aim at image classification or annotation, where images containing target animal are labeled as positive and the other images are labeled as negative. Each image is regarded as a bag, while the instances are the segments characterized by several descriptors. The segments of certain animals, like fox, might be similar to the background segments. And the accuracy of tested methods is relatively low on the fox dataset. The results on these benchmark datasets are regarded as the representatives.

The proposed BDR method can utilize the bag representations provided by BS or ES methods, which makes it flexible enough to cope with the real-world MIL tasks. Therefore, different bag representations might result in different results. In this paper, we use a joint bag representation based on MIFA (Section 3.2) and miVLAD as the explicit representation, while the multiple dissimilarities proposed in [11] are used as the implicit representation. Specifically, the dissimilarity between two bags is defined as the average of meanmean, meanmin, maxmin, and minmin dissimilarities. The results shown below might be further refined if more suitable bag representations are utilized. All the bags within training and testing set are used as the anchor bags if there is no special statement.

Table 2
The accuracy of tested methods on benchmark datasets, where the top 3 results are highlighted in bold.

	Musk1	Musk2	Elephant	Fox	Tiger	Rank	Avg.
Simple-MI	0.832(0.123)	0.853(0.111)	0.801(0.088)	0.546(0.092)	0.778(0.092)	12.4(0.4)	0.762
mi-SVM	0.874(0.120)	0.836(0.088)	0.822(0.073)	0.582(0.102)	0.789(0.089)	11.4(0.9)	0.781
EM-DD	0.849(0.098)	0.869(0.108)	0.771(0.098)	0.609(0.101)	0.730(0.069)	11.0(2.1)	0.766
MInD	0.893(0.019)	**0.888(0.034)**	0.857(0.018)	**0.651(0.011)**	0.819(0.021)	**5.0(2.7)**[3]	**0.822**[3]
miVLAD	0.871(0.097)	0.872(0.095)	0.850(0.080)	0.620(0.098)	0.811(0.087)	8.6(2.0)	0.805
miFV	**0.909(0.089)**	**0.884(0.094)**	0.852(0.081)	0.621(0.109)	0.813(0.083)	5.6(2.7)	0.816
mi-Net	0.889(0.088)	0.858(0.110)	0.858(0.083)	0.613(0.078)	0.824(0.076)	7.8(1.6)	0.808
MI-Net	0.887(0.091)	0.859(0.102)	0.862(0.077)	0.622(0.084)	0.830(0.072)	6.4(2.0)	0.812
MI-Net with DS	0.894(0.093)	0.874(0.097)	**0.872(0.072)**	**0.630(0.080)**	**0.845(0.087)**	**3.0(1.2)**[2]	**0.823**[2]
MI-Net with RC	0.898(0.097)	0.873(0.098)	0.857(0.089)	0.619(0.104)	0.836(0.083)	5.4(0.9)	0.817
Att. Net	0.892(0.040)	0.858(0.048)	**0.868(0.022)**	0.615(0.043)	0.839(0.022)	6.6(2.6)	0.814
Gated Att. Net	**0.900(0.050)**	0.863(0.042)	0.857(0.027)	0.603(0.029)	**0.845(0.018)**	6.4(2.8)	0.814
Proposed	**0.926(0.079)**	**0.905(0.092)**	**0.908(0.054)**	**0.629(0.110)**	**0.869(0.066)**	**1.4(0.7)**[1]	**0.847**[1]

Table 3
The accuracy on Mutagenesis datasets, where the top 3 results are highlighted in bold.

	Mutagenesis-A	Mutagenesis-B	Mutagenesis-C	Rank	Avg.
Simple-MI	0.644(0.038)	0.798(0.078)	0.788(0.084)	6.7(0.9)	0.743
miVLAD	0.761(0.128)	**0.857(0.081)**	**0.841(0.062)**	**2.7(0.9)**[2]	**0.820**[2]
miFV	**0.798(0.081)**	**0.840(0.095)**	0.804(0.010)	3.3(1.2)	0.814
MInD	0.751(0.103)	0.801(0.085)	0.815(0.088)	4.7(0.5)	0.789
mi-Net	0.649(0.046)	0.659(0.028)	0.691(0.082)	7.7(0.5)	0.666
MI-Net	**0.771(0.083)**	0.824(0.068)	**0.851(0.081)**	**2.7(1.2)**[3]	**0.815**[3]
Att.Net	0.665(0.023)	0.659(0.087)	0.712(0.059)	6.7(0.5)	0.679
Proposed	**0.808(0.088)**	**0.859(0.075)**	**0.836(0.087)**	**1.7(0.9)**[1]	**0.834**[1]

4.2. Comparison with existing methods

In this section, there are 12 existing MIL methods used for comparison. The comparison methods include: SimpleMI [28], mi-SVM [8], EM-DD [20], MInD [11], miVLAD and miFV [13], mi-Net [10], MI-Net, MI-Net with DS, MI-Net with RC [14], Att. Net and Gated Att. Net [32]. On the benchmark datasets, all the 12 methods are included. Meanwhile, 7 methods among them are selected as the representatives and tested on the other datasets. The representatives for regular methods are: SimpleMI, MInD, miVLAD and miFV, which are typical MIL methods based on bag representations. Three basic MIL networks, namely mi-Net, MI-Net, and Att. Net, are selected as the representatives for deep networks. Many deep MIL networks can be viewed as the variety of these three methods. We present the best results from the two proposed BDR classifiers, namely BDR-BLS and BDR-SVM.

Table 2 shows the results on the five benchmark datasets. The proposed method achieves the best results on four out of five datasets. In terms of overall performance, both the average accuracy (0.847) and the average rank (1.4) of the proposed method are the highest. Compared with MInD, the results of proposed method are significantly better because of the assistance from the extra explicit bag representation. The bag dissimilarity regularization has ensured that the proposed methods can achieve better results than the ES methods like miVLAD and miFV. It should be noted that the proposed method also outperforms the deep MIL networks. On such small-scale datasets, the training consumption of proposed method is much less than that of deep networks, and therefore the proposed method can be viewed as an alternative to these deep networks.

The results on Mutagenesis datasets and Web datasets are presented in Table 3 and Table 4. On the Web datasets, the dimensionality of instances is reduced to 70 using principal component analysis (PCA). On the Mutagenesis datasets, the MIL methods using explicit bag representations, namely miVLAD and miFV, perform better than the methods using implicit representation. On the contrary, MInD which utilizes implicit representations performs better on the Web datasets. By utilizing both the explicit and the implicit representations simultaneously, the proposed method achieves satisfactory results on all these datasets. The average accuracy of proposed method reaches 0.834 on the Mutagenesis datasets, while the average accuracy on the Web datasets is 0.845. These results illustrate that the proposed method has achieved better adaptability compared with the methods relying on a single bag representation.

Table 5 presents the results on Textile and Messidor datasets. For the Messidor dataset, we reduce the dimensionality of instances to 100 using PCA. The proposed method achieves the accuracy of 0.855 on the Textile datasets, which is higher than that of miFV and MInD. It can also be found that miVLAD, miFV and MInD outperform the rest comparison methods on textile dataset. Thus, the second order statics captured by miFV and the bag dissimilarities defined by MInD might be more suitable for textile dataset. The proposed method can utilize both the second order statics and the bag dissimilarities, which is likely to be the origin of high performance. Assisted by the hyperspectral techniques, the proposed method has provided a potential solution to the problem of automated dyeing uniformity testing. The results are similar on the Messidor dataset, where the proposed method outperforms all the compared methods. The results also further prove that the proposed BDR framework could improve the accuracy and the generalization ability.

The results of Freidman test [55] are shown in Fig. 2, which presents the overall accuracy of the methods on all the datasets. The proposed method and the 7 representative methods are included for Freidman analysis. To obtain the results above, these methods are tested on all the datasets. It can be found that the average rank of the proposed method is the best. MInD and miFV also achieve competitive results. Except for these two methods, the proposed method performs significantly better than the other methods on these datasets.

The deep learning-based MIL methods have drawn increasing attention, due to their competitive performance. Representation learning ability enables the deep MIL methods to learn a bag representation under certain constraints, which could be more effective than the predefined representations. In this paper, the BDR classifiers assisted by Fisher score and bag dissimilarity outperform

Table 4
The accuracy of tested methods on Web datasets, where the top 3 results are highlighted in bold.

	Web1	Web2	Web3	Web4	Rank	Avg.
Simple-MI	0.761(0.073)	0.769(0.089)	0.814(0.022)	0.845(0.091)	6.0(2.0)	0.797
miVLAD	0.823(0.087)	0.812(0.085)	0.814(0.022)	**0.858(0.098)**	**3.8(0.4)**[3]	0.827
miFV	0.814(0.044)	0.808(0.032)	0.814(0.022)	**0.878(0.091)**	4.0(1.2)	**0.829**[3]
MInD	**0.841(0.091)**	0.808(0.096)	**0.844(0.088)**	0.842(0.102)	**3.0(2.0)**[2]	**0.834**[2]
mi-Net	0.806(0.089)	**0.823(0.036)**	0.805(0.075)	0.815(0.082)	5.5(2.7)	0.750
MI-Net	**0.831(0.050)**	0.806(0.116)	0.804(0.091)	0.831(0.110)	6.0(1.9)	0.818
Att.Net	0.789(0.145)	**0.814(0.116)**	**0.821(0.091)**	0.830(0.136)	5.0(2.0)	0.814
Proposed	**0.833(0.085)**	**0.821(0.035)**	**0.834(0.060)**	**0.882(0.106)**	**1.7(0.4)**[1]	**0.843**[1]

Table 5
The accuracy on Textile and Messidor datasets, where the top 3 results are highlighted in bold.

	Textile	Messidor	Rank	Avg.
Simple-MI	0.544(0.137)	0.693(0.046)	5.5(0.5)	0.619
miVLAD	**0.798(0.091)**	0.691(0.037)	4.5(1.5)	**0.744**[3]
miFV	**0.810(0.185)**	**0.709(0.029)**	**2.5(0.5)**[2]	**0.760**[2]
MInD	0.769(0.130)	0.665(0.071)	5.5(1.5)	0.717
mi-Net	0.505(0.045)	0.602(0.065)	8.0(0.0)	0.569
MI-Net	0.507(0.021)	**0.717(0.031)**	4.5(2.5)	0.612
Att.Net	0.545(0.057)	0.700(0.067)	**4.5(0.5)**[3]	0.623
Proposed	**0.855(0.088)**	**0.727(0.022)**	**1.0(0.0)**[1]	**0.791**[1]

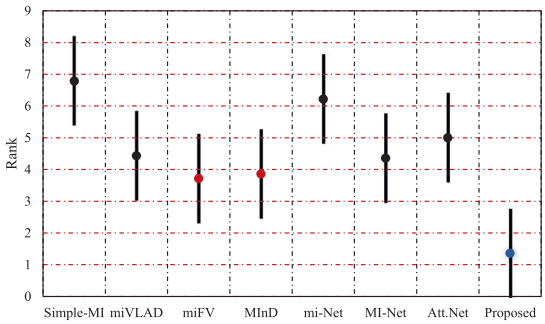

Fig. 2. The results of Friedman test. The points represent the average ranks, while the bars indicate the critical values for a two-tailed test at 95% significance.

several deep learning methods, in terms of both accuracy and efficiency. It is also possible for the BDR classifiers to utilize the representations learned by deep neural networks, and the learned representations could be incorporated into the proposed BDR framework in a similar way.

4.3. Analysis

In this sub-section, several extra experiments are carried out to explore the characteristics of proposed BDR method. First, the two BDR classifiers are compared with each other, in terms of both accuracy and efficiency. Then, the effect of anchor bags is discussed and tested. Additionally, an ablation experiment is conducted to evaluate the effectiveness of proposed method.

4.3.1. BDR classifiers

In this paper, we incorporate the regularization term into two classifiers, proposing BDR-BLS and BDR-SVM. The BDR-SVM can achieve competitive results on many MIL datasets, but the cubic computation complexity limits its application on large-scale datasets. With $\mathcal{R}_{BDA}$ based on anchor bags, the computation consumption of BDR-BLS only increases linearly when the number of bags increases. Thus, BDR-BLS is a relatively better choice for large-scale dataset.

Here, BDR-SVM uses the RBF kernel, while σ represents the parameter of kernel. C and λ_2 are the two hyper-parameters that adjust the characteristics of classifier. Higher λ_2 means that the effect of bag dissimilarity becomes larger, and vice versa. BDR-BLS has five hyper-parameters: N_1, N_2, N_3, λ_1, and λ_2. N_1 and N_2 control the number of feature nodes, which are fixed at 10 in this paper. N_3 represents the number of enhancement nodes. λ_1 represents the effect of l_2 regularization term, while λ_2 controls the influence of bag dissimilarity.

The parameter settings and the accuracy of the BDR classifiers are presented in Table 6. BDR-SVM performs better on 9 out of 14 datasets, which indicates that BDR-SVM is relatively more adaptive on these datasets. It should be noted that BDR-BLS is tested with a fixed number of feature nodes, and the results might be improved if the parameters (N_1 and N_2) are adjusted. The value of optimal λ_2 indicates the effectiveness of BDR, to a certain extent. On the three Mutagenesis datasets, for example, it seems that the performance improvements are limited as the optimal λ_2 is lower than 0.0001. On the contrary, the optimal λ_2 is larger on the benchmark datasets, which means that these datasets are more likely to satisfy our assumption.

The training time of BDR-SVM and BDR-BLS is presented in Fig. 3. It should be noted that the vertical axes are shown in log-scale, where the second is used as the unit of time. Four benchmark datasets are used: Musk1, Musk2, Elephant, and Fox. We record the total training time to accomplish a single ten-fold cross-validation. And the optimal values of hyper-parameters are used in this experiment. Although the proposed methods are slightly slower than MInD, this efficiency degradation is acceptable, considering the accuracy improvement. On elephant dataset, for example, BDR-SVM achieves better accuracy (0.908) than MInD (0.857) with similar training time (8.66s versus 9.72s). BDR-SVM is slightly faster than BDR-BLS on Musk1 dataset, which is contrary to the results on the other datasets. This might result from the different parameter setting and the characteristics of datasets. The training time of BDR methods is much less than the that of deep learning methods, and thus the proposed methods can be used as the alternatives to deep MIL when the computation power is limited.

4.3.2. Anchor bags

Like many manifold-based and graph-based methods, the BDR methods suffer from quadratic or even cubic complexity, especially when dealing with the large-scale datasets. Thus, we propose bag dissimilarity regularization based on anchor bags to lower the complexity of some BDR classifiers, such as BDR-BLS. As we use a set of anchor bags to approximate the manifold structure, there will be inevitable performance degradation due to the loss of local information. Here, we test the performance of BDR-BLS on several datasets to evaluate the effect of anchor bags.

We evaluate the performance of BDR-BLS with different numbers of anchor bags on four relatively small datasets, namely Musk1, Musk2, Elephant, and Fox. Here, we focus on the dependence of performance on the number of anchor bags. During the experiment, only the number of anchor bags changes while the number of training bags remains unchanged. The results are shown in Fig. 4, where the accuracy and the total training time of a sin-

Table 6
The comparison between BDR-SVM and BDR-BLS.

	BDR-SVM Acc.	C	σ	λ_2	BDR-BLS Acc.	N_1	N_2	N_3	λ_1	λ_2
Musk1	0.917(0.090)	5	0.001	0.01	**0.926(0.079)**	10	10	1000	0.1	0.1
Musk2	**0.905(0.092)**	10	0.001	0.01	0.859(0.108)	10	10	300	10	1
Elephant	**0.908(0.054)**	10	0.001	0.05	0.869(0.073)	10	10	1000	0.1	1
Fox	0.625(0.103)	2	0.0005	0.001	**0.629(0.110)**	10	10	1200	0.1	0.01
Tiger	**0.869(0.067)**	80	0.002	0.001	0.850(0.087)	10	10	1000	0.01	0.05
Mutagenesis-A	**0.808(0.088)**	200	0.05	0.0001	0.782(0.111)	10	10	1000	0.01	0.01
Mutagenesis-B	**0.859(0.086)**	10	0.05	0.00001	0.859(0.075)	10	10	1000	0.0001	0.01
Mutagenesis-C	0.832(0.081)	50	0.01	0.00001	**0.836(0.087)**	10	10	1000	0.01	0.001
Web1	**0.833(0.085)**	10	0.001	0.1	0.833(0.102)	10	10	1000	0.01	0.0001
Web2	**0.821(0.035)**	10	0.0001	0.001	0.791(0.109)	10	10	1200	0.01	0.01
Web3	**0.834(0.060)**	10	0.0001	0.001	0.769(0.113)	10	10	1200	0.01	0.01
Web4	0.845(0.084)	10	0.0001	0.001	**0.882(0.106)**	10	10	1200	0.01	0.01
Textile	**0.855(0.088)**	100	0.001	0.001	0.851(0.092)	10	10	1000	0.01	0.01
Messidor	0.710(0.038)	10	0.001	0.00001	**0.727(0.022)**	10	10	4000	0.01	0.005

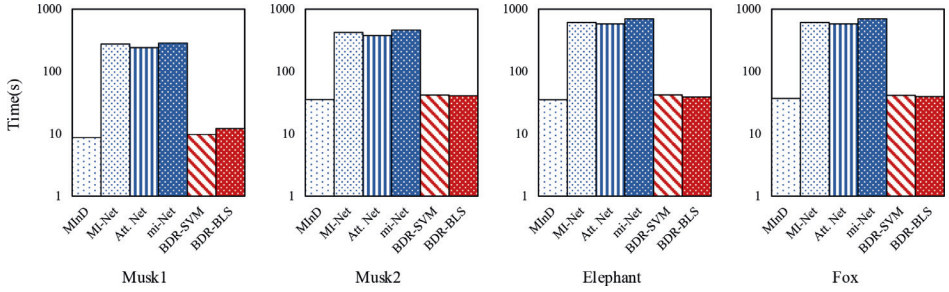

Fig. 3. The training time of the tested methods on Musk1, Musk2, Elephant, and Fox datasets.

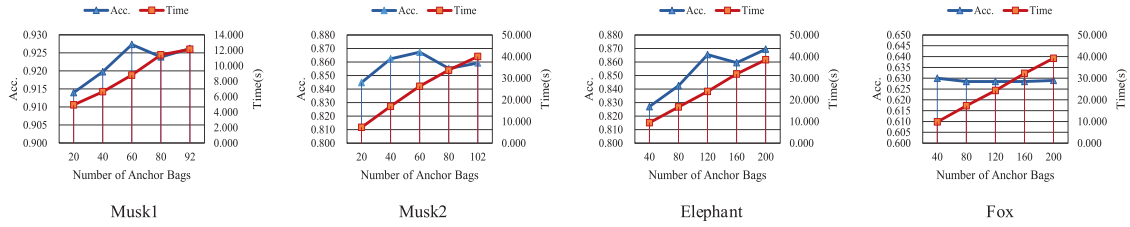

Fig. 4. The accuracy and training time of BDR-BLS with different number of anchor bags on Musk1, Musk2, Elephant, and Fox datasets.

gle ten-fold cross-validation are presented. On all the datasets, the training time increases linearly as the number of anchor bags increases, which is in accord with the discussion in Section 3.3.3. As for the accuracy curve, the experiment results are against intuitive sense. The accuracy is not always proportionate to the number of anchor bags. In theory, using the anchor bags to approximate the local geometric structure of data can result in accuracy loss. But in practice, the MIL data might not strictly follow the manifold assumption. In other words, some of the bags does not lie exactly on the sub-manifold in the bag space. And the approximation might alleviate the performance loss caused by the inconsistency. Moreover, different datasets have different characteristics, resulting in different performance curves. The fox dataset, for example, has distinctive characteristics which is illustrated in the difficulty of classification. And the performance curve of BDR-BLS on fox dataset is also distinctive, with a stable performance hardly affected by the number of anchor bags.

4.3.3. Ablation experiment

To prove the effectiveness of proposed methods, an ablation experiment is carried out. In this experiment, we evaluate the proposed BDR framework and MIFA. The feature vector of MIFA is conjoined with that of miVLAD with a single centroid. The final representation vector is classified by two classifiers: SVM (MIFA-SVM) and BLS (MIFA-BLS). A typical ES method (miV&F) is introduced and the results from [13] are used for comparison. The results of proposed method are obtained by repeating ten-fold cross-validation for 10 times, which is same as above. Five benchmark datasets are used for evaluation, and the results are shown in Fig. 5. Compared with miV&F, the overall performance of MIFA-SVM is slightly better. It should be noted that the miVLAD representations for MIFA-based methods have not been tuned, while those of miV&F are adjusted to obtain better accuracy in [13]. As for the effectiveness of bag dissimilarity regularization, it can be found that the bag dissimilarity regularization has improved the performance. On the Elephant dataset, for example, the accuracy of BDR-SVM is higher than that of MIFA-SVM by 4.2%. For BLS-based methods, the dissimilarity regularization is also effective. The results of BDR-BLS are better than those of MIFA-BLS by 2.5% on the Elephant dataset.

To further illustrate the superiority of proposed BDR framework, another experiment is conducted. It has been reported that directly concatenating multiple representation vectors might improve the performance. We carried out an experiment to compare this simple method with the BDR methods. As only the explicit bag representations can be conjoined with each other, the implicit representations must be transformed into explicit ones. Here, we use dissimilarity vectors [11] to extract explicit features of the implicit representation. The dissimilarity vectors are conjoined with the vectors of MIFA and classified by a SVM (MInD+MIFA). The results

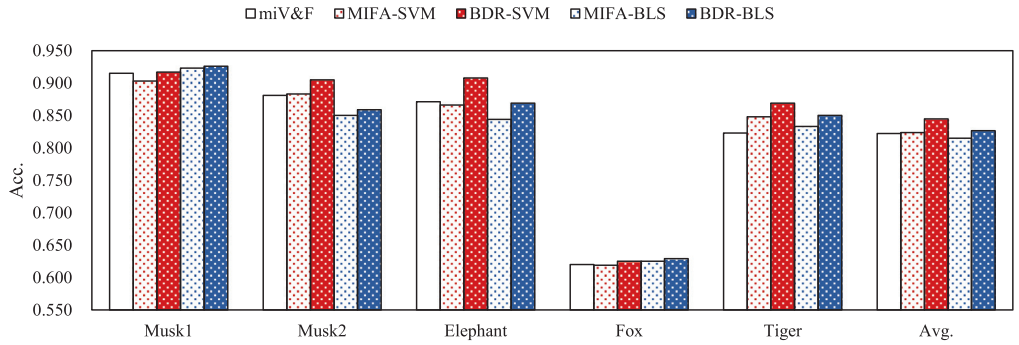

Fig. 5. The results of ablation experiment on the five benchmark datasets.

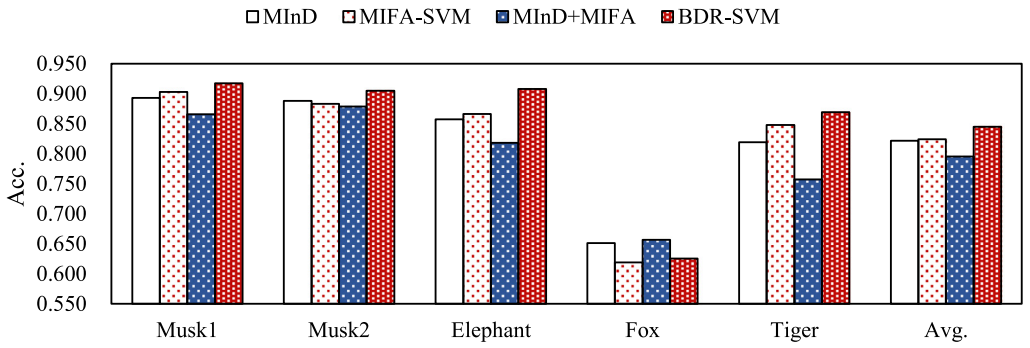

Fig. 6. The comparison between MInD+MIFA and BDR-SVM on the five benchmark datasets.

are shown in Fig. 6. The results indicated that the combination of dissimilarity vectors and MIFA fails to improve the performance. MInD+MIFA has poorer performance on four out of five datasets, compared with MInD and MIFA-SVM. On the contrary, the BDR has effectively improved the accuracy. The average accuracy of BDR-SVM outperforms the two basic methods (MInD and MIFA-SVM) by more than 2%, for example.

Concatenating multiple feature vectors might improve the performance on certain condition, but this simple method has several drawbacks. Firstly, this method can only utilize explicit representations, while BDR methods can utilize both implicit and explicit representations. Secondly, the higher dimensionality results from conjunction will not only increase computation consumption, but also lower the accuracy. With limited training bags, the performance will deteriorates as the dimensionality increases, which is known as Hughes phenomenon [15]. The BDR methods incorporate extra representations by utilizing the correlation between bags without increasing the dimensionality. Thirdly, it is difficult to adjust the weights of representations when concatenating them into a single vector. For BDR methods, the influence of representations is controlled by the hyper-parameters, which enables experts to adjust the effect of representations according to the characteristics of data.

5. Conclusion

It has been found that utilizing the multiple bag representations derived from different MIL methods could improve the performance. Directly concatenating several explicit bag representations might obtain better results on certain conditions. However, this method could not utilize the implicit representations, and fails to cope with the diverse representations. In this paper, we propose a BDR framework that can incorporate multiple bag representations, regardless of whether the representations are implicit or explicit. The proposed BDR methods model the local geometric structure of data in the implicit embedded space, and incorporate it into the regular classifiers. The influence of bag representations can be adjusted by changing the values of hyper-parameters. Besides that, an ES method based on FA (MIFA) is proposed. We incorporate the regularization term into two supervised classifiers and propose two BDR classifiers, namely BDR-BLS and BDR-SVM. Utilizing anchor bags for the approximation of submanifold structure can reduce the computation complexity of BDR-BLS to a linear complexity. The experiment results on 14 datasets have indicated the effectiveness of proposed methods. The characteristics of the BDR methods are also analyzed and discussed.

As for future work, we are going to extend the BDR framework to deep MIL methods. It is undeniable that some deep MIL methods have achieved state-of-art results. And the representation learning ability of deep networks is crucial to some real-world tasks. A deep MIL network that is able to learn multiple representations would be effective.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under grant 62073287.

References

[1] T.G. Dietterich, R.H. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, Artif. Intell. 89 (1) (1997) 31–71, doi:10.1016/S0004-3702(96)00034-3.
[2] Z.H. Zhou, Y.Y. Sun, Y.F. Li, Multi-instance learning by treating instances as non-i.i.d. samples, in: Proc. 26th Int. Conf. Mach. Learn. (ICML), 2009, pp. 1249–1256, doi:10.1145/1553374.1553534.
[3] K. Ali, K. Saenko, Confidence-rated multiple instance boosting for object detection, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), 2014, pp. 2433–2440, doi:10.1109/CVPR.2014.312.

[4] A. Zare, C. Jiao, T. Glenn, Discriminative multiple instance hyperspectral target characterization, IEEE Trans. Pattern Anal. Mach. Intell. 40 (10) (2018) 2342–2354, doi:10.1109/TPAMI.2017.2756632.
[5] Q. Wang, Y. Yuan, P. Yan, X. Li, Saliency detection by multiple-instance learning, IEEE Trans. Cybern. 43 (2) (2013) 660–672, doi:10.1109/TSMCB.2012.2214210.
[6] K. He, W. Zhao, X. Xie, et al., Synergistic learning of lung lobe segmentation and hierarchical multi-instance classification for automated severity assessment of covid-19 in ct images, Pattern Recognit. 113 (2021) 107828, doi:10.1016/j.patcog.2021.107828.
[7] J. Amores, Multiple instance classification: review, taxonomy and comparative study, Artif. Intell. 201 (2013) 81–105, doi:10.1016/j.artint.2013.06.003.
[8] S. Andrews, I. Tsochantaridis, T. Hofmann, Support vector machines for multiple-instance learning, in: Proc. Neural Inf. Process Syst. (NIPS), 2002, pp. 1073–1080.
[9] R.C. Bunescu, R.J. Mooney, Multiple instance learning for sparse positive bags, in: Proc. 24th Int. Conf. Mach. Learn. (ICML), 2007, pp. 105–112, doi:10.1145/1273496.1273510.
[10] J. Wu, Y. Yu, C. Huang, K. Yu, Deep multiple instance learning for image classification and auto-annotation, in: 2015 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 3460–3469, doi:10.1109/CVPR.2015.7298968.
[11] V. Cheplygina, D.M. Tax, M. Loog, Multiple instance learning with bag dissimilarities, Pattern Recognit. 48 (1) (2015) 264–275, doi:10.1016/j.patcog.2014.07.022.
[12] Y. Chen, J. Bi, J. Wang, Miles: multiple-instance learning via embedded instance selection, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 1931–1947, doi:10.1109/TPAMI.2006.248.
[13] X. Wei, J. Wu, Z. Zhou, Scalable algorithms for multi-instance learning, IEEE Trans. Neural Netw. Learn. Syst. 28 (4) (2017) 975–987, doi:10.1109/TNNLS.2016.2519102.
[14] X. Wang, Y. Yan, P. Tang, X. Bai, W. Liu, Revisiting multiple instance neural networks, Pattern Recognit. 74 (2018) 15–24, doi:10.1016/j.patcog.2017.08.026.
[15] G. Hughes, On the mean accuracy of statistical pattern recognizers, IEEE Trans. Inf. Theory 14 (1) (1968) 55–63.
[16] Q. Wang, X. He, X. Li, Locality and structure regularized low rank representation for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 57 (2) (2019) 911–923, doi:10.1109/TGRS.2018.2862899.
[17] J. Foulds, E. Frank, A review of multi-instance learning assumptions, Knowl. Eng. Rev. 25 (1) (2010) 1—-25.
[18] G. Melki, A. Cano, S. Ventura, Mirsvm: multi-instance support vector machine with bag representatives, Pattern Recognit. 79 (2018) 228–241, doi:10.1016/j.patcog.2018.02.007.
[19] D.R. Dooly, Q. Zhang, S.A. Goldman, R.A. Amar, Multiple instance learning of real valued data, J. Mach. Learn. Res. 3 (null) (2003) 651–678.
[20] Q. Zhang, S.A. Goldman, Em-dd: An improved multiple-instance learning technique, in: Proc. Neural Inf. Process. Syst. (NIPS), 2001, pp. 1073–1080, doi:10.5555/2980539.2980677.
[21] E. Frank, X. Xu, Applying propositional learning algorithms to multi-instance data, Technical Report, Dept. Comput. Sci., Univ. Waikato, 2003.
[22] J. Wang, J.-D. Zucker, Solving the multiple-instance problem: A lazy learning approach, in: Proc. 7th Int. Conf. Mach. Learn. (ICML), 2000, pp. 1119–1126.
[23] G.A. Edgar, Measure, Topology, and Fractal Geometry, Springer-Verlag, New York, 1995.
[24] T. Gärtner, P. Flach, A. Kowalczyk, A. Smola, Multi-instance kernels, in: Proc. 9th Int. Conf. Mach. Learn. (ICML), 2002, pp. 179–186.
[25] X. Wang, Y. Yan, P. Tang, W. Liu, X. Guo, Bag similarity network for deep multi-instance learning, Inform. Sciences 504 (2019) 578–588, doi:10.1016/j.ins.2019.07.071.
[26] Z. Chi, Z. Wang, W. Du, Explicit metric-based multiconcept multi-instance learning with triplet and superbag, IEEE Trans. Neural Netw. Learn. Syst. (2021) 1–10, doi:10.1109/TNNLS.2021.3071814.
[27] D.M.J. Tax, M. Loog, R.P.W. Duin, V. Cheplygina, W.-J. Lee, Bag dissimilarities for multiple instance learning, in: Proceedings of the First International Conference on Similarity-Based Pattern Recognition, in: SIMBAD'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 222–234.
[28] J. Foulds, Learning instance weights in multi-instance learning, 2008,
[29] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 3304–3311, doi:10.1109/CVPR.2010.5540039.
[30] J. Sánchez, F. Perronnin, T. Mensink, Image classification with the fisher vector: theory and practice, Int. J. Comput. Vis. 105 (2013) 222–245.
[31] T.S. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, in: Proc. 11th Int. Conf. Neural Inf. Process. Syst., in: NIPS'98, MIT Press, Cambridge, MA, USA, 1998, pp. 487–493.
[32] M. Ilse, J. Tomczak, M. Welling, Attention-based deep multiple instance learning, in: J. Dy, A. Krause (Eds.), Proc. 35th Int. Conf. Mach. learn. (ICML), volume 80, 2018, pp. 2127–2136.
[33] D. Cai, X. Wang, X. He, Probabilistic dyadic data analysis with local and global consistency, in: Proc. 26th Int. Conf. Mach. Learn., in: ICML '09, Association for Computing Machinery, New York, NY, USA, 2009, pp. 105–112, doi:10.1145/1553374.1553388.
[34] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[35] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: Proc. 14th Int. Conf. Neural Inf. Process. Syst., in: NIPS'01, MIT Press, Cambridge, MA, USA, 2001, pp. 585–591.
[36] F.R.K. Chung, Spectral graph theory, in: CBMS Regional Conference Series in Mathematics, volume 92, Am. Math. Soc., 1997.
[37] D. Cai, X. He, J. Han, T.S. Huang, Graph regularized nonnegative matrix factorization for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1548–1560, doi:10.1109/TPAMI.2010.231.
[38] X. Chen, D. Cai, Large scale spectral clustering with landmark-based representation, in: Proc. 25th AAAI Conf. Artif. Intell., 2011.
[39] S. Zhao, C. Gao, S. Mukherjee, B.E. Engelhardt, Bayesian group factor analysis with structured sparsity, J. Mach. Learn. Res. 17 (1) (2016) 6868–6914.
[40] O. Aran, L. Akarun, A multi-class classification strategy for fisher scores: application to signer independent sign language recognition, Pattern Recognit. 43 (5) (2010) 1776–1788, doi:10.1016/j.patcog.2009.12.002.
[41] M. Dixit, Y. Li, N. Vasconcelos, Semantic fisher scores for task transfer: using objects to classify scenes, IEEE Trans. Pattern Anal. Mach. Intell. 42 (12) (2020) 3102–3118, doi:10.1109/TPAMI.2019.2921960.
[42] Z. Ghahramani, G.E. Hinton, The EM algorithm for mixtures of factor analyzers, Technical Report, University of Toronto, Toronto, Canada, 1997.
[43] C.L.P. Chen, Z. Liu, Broad learning system: an effective and efficient incremental learning system without the need for deep architecture, IEEE Trans. Neural Netw. Learn. Syst. 29 (1) (2018) 10–24, doi:10.1109/TNNLS.2017.2716952.
[44] Y. Kong, Y. Cheng, C.L.P. Chen, X. Wang, Hyperspectral image clustering based on unsupervised broad learning, IEEE Geosci. Remote Sens. Lett. 16 (11) (2019) 1741–1745, doi:10.1109/LGRS.2019.2907598.
[45] W. Yu, C. Zhao, Broad convolutional neural network based industrial process fault diagnosis with incremental learning capability, IEEE Trans. Ind. Electron. 67 (6) (2020) 5081–5091, doi:10.1109/TIE.2019.2931255.
[46] H. Huang, T. Zhang, C. Yang, C.L.P. Chen, Motor learning and generalization using broad learning adaptive neural control, IEEE Trans. Ind. Electron. 67 (10) (2020) 8608–8617, doi:10.1109/TIE.2019.2950853.
[47] T. Qiu, X. Liu, X. Zhou, W. Qu, Z. Ning, C.L.P. Chen, An adaptive social spammer detection model with semi-supervised broad learning, IEEE Trans. Knowl. Data Eng. (2020), doi:10.1109/TKDE.2020.3047857. 1–1
[48] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011), doi:10.1145/1961189.1961199.
[49] J. Platt, Using analytic qp and sparseness to speed training of support vector machines, in: Proc. 12th Int. Conf. Neural Inf. Process. Syst., MIT Press, Cambridge, MA, USA, 1999, pp. 557–563.
[50] E. Decencière, X. Zhang, G. Cazuguel, Feedback on a publicly distributed image database: the messidor database, Image Anal. Stereol. (2014) 231–234.
[51] A. Srinivasan, S. Muggleton, R. King, Comparing the use of background knowledge by inductive logic programming systems, in: Proc. 5th Int. Workshop Inductive Log. Program., 1995, pp. 199–230.
[52] Z.-H. Zhou, K. Jiang, M. Li, Multi-instance learning based web mining, Appl. Intell. 22 (2) (2005) 135–147.
[53] B. Rasti, D. Hong, R. Hang, P. Ghamisi, X. Kang, J. Chanussot, J.A. Benediktsson, Feature extraction for hyperspectral imagery: the evolution from shallow to deep (overview and toolbox), IEEE Geosc. Remote Sens. Mag. 8 (4) (2020) 60–88, doi:10.1109/MGRS.2020.2979764.
[54] U. Syed, R.H. Wardman, Assessment of uniformity of fibre coloration in tencel woven fabrics dyed using reactive dyes, Coloration Technol. 127 (6) (2011) 418–425.
[55] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

Shiluo Huang received the B.Sc. degree from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2018. He is currently pursuing the Ph.D. degree with the college of control science and engineering, Zhejiang University, Hangzhou, China. His current research interests include machine learning and its application to instrumentation.

Zheng Liu is currently working toward the Ph.D. degree in the Department of Control Science and Engineering, Zhejiang University, Hangzhou, China. His research interests include machine learning and its applications

Wei Jin received the B.E. degree in mechanical and electrical engineering from Dalian Polytechnic University, China, in 1991, and the Ph.D. degree from the Jilin University, China, in 2007. He has been a researcher at Zhejiang University, China, since 2007. His current interests include atomic spectrometry analysis, on line detection technology and machine learning. In these research fields, he has been supported by the National Key Research and Development Project, China, for many times.

Ying Mu is a professor in State Key Laboratory of Industrial Control Technology, Zhejiang University, China. She received her M.D. from the Norman Bethune University of Medical Sciences in 1997. From 2000 to 2006, she was a professor at Key Laboratory for Molecular Enzymology and Engineering of the Ministry of Education, Jilin University. Since 2006, she was hired as a professor at Zhejiang University. Dr. Mu has published over 100 journal papers and several books on microfluidics and biochemistry. Her research focuses on the development of multiple microfluidics-based chips and devices for efficient molecular diagnosis