SubMIL: Discriminative subspaces for multi-instance learning

Jiazheng Yuan\textsuperscript{a,b,*}, Xiankai Huang\textsuperscript{c}, Hongzhe Liu\textsuperscript{a}, Bing Li\textsuperscript{d}, Weihua Xiong\textsuperscript{d}

\textsuperscript{a} Beijing Key Laboratory of Information Service Engineering, Beijing 100101, China
\textsuperscript{b} Computer Technology Institute of Beijing Union University, Beijing 100101, China
\textsuperscript{c} Tourism Institute of Beijing Union University, Beijing 100101, China
\textsuperscript{d} National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

\section*{A R T I C L E  I N F O}

Keywords: Multi-instance learning Low rank Subspace

\section*{A B S T R A C T}

As an important learning scheme for Multi-Instance Learning (MIL), the Instance Prototype (IP) selection-based MIL algorithms transform bags into a new instance feature space and achieve impressed classification performance. However, the number of IPs in the existing algorithms linearly increases with the scale of the training data. The performance and efficiencies of these algorithms are easily limited by the high dimension and noise when facing a large scale of training data. This paper proposes a discriminative subspaces-based instance prototype selection method that is suitable for reducing the computation complexity for large scale training data. In the proposed algorithm, we introduce the low-rank matrix recovery technique to find two discriminative and clean subspaces with less noise; then present a $\ell_2$ norm-based self-expressive sparse coding model to select the most representative instances in each subspace. Experimental results on several data sets show that our algorithm achieves superior and stable performance but with lower dimension compared with other IP selection strategies.

\section*{1. Introduction}

As a variant of the supervised learning framework, Multiple Instance Learning (MIL) represents a sample with a bag of several instances instead of a single instance. It only gives each bag, not each instance, a discrete or real-value label. In binary classification case, MIL assumes that all instances in a negative bag are actually negative, while at least one positive instance in a positive bag is actually positive. Therefore, a positive bag may contain some negative instances. The MIL problem is firstly introduced in the context of drug activity prediction\cite{1}. Since then, a wide variety of different applications have been formulated as multiple instances learning problem, such as stock prediction\cite{2}, identification of proteins\cite{3}, and text classification\cite{4}. Another important emerging application comes from the image understanding, in which an image is viewed as a bag of local regions (or objects), and then the MIL technique is used for content based image retrieval\cite{5,9}, image categorization\cite{6}, image emotion analysis\cite{7}, and social media\cite{8}.

\subsection*{1.1. Related work}

Past decades have witnessed great progress in learning algorithms for MIL. They can be roughly divided into two major groups: generative model and discriminative model. Most earlier approaches, including Axis-Parallel Rectangles (APR)\cite{1}, diverse density (DD)\cite{2}, and EM-based diverse density (EM-DD)\cite{10} belong to the generative category, in which they locate a region of interest in the instance feature space that has minimal distances from all positive instances and maximal distances from all negative instances. The methods belonging to the discriminative category try to transform the MIL problem into single-instance problem and then employ the standard discriminative learning paradigm. Citation K-nearest neighbor (KNN)\cite{11} extends the standard KNN classifier to MIL. mi-SVM, MI-SVM\cite{4}, DD-SVM\cite{2}, and missSVM\cite{12} are all variants of Support Vector Machine (SVM) classifier for MIL. Sparse representation model has been extended for MIL\cite{13,14} and achieves good performance. Recently, deep learning is also applied to solve MIL problem\cite{28}. Other prominent classifiers have also been extended in MIL problems, for example, Multiple Instance Boosting (MI-Boosting)\cite{15}, multi-instance logistic regression\cite{16}, and fast bundle algorithm for MIL\cite{26}.

Although there are many elaborate learning schemes for MIL, Chen et al.\cite{6} points that most of the existing algorithms are unsuitable for tackling some specific MIL applications such as object recognition. Because the object recognition problem cannot strictly satisfy the original definition of MIL\cite{1,27}. To address this
issue, a novel solution, instance selection-based algorithm, has been proposed for MIL and achieves improved performance. The representative methods include MIL-Embedded instance Selection (MILES) [6], MIL-Disambiguration (MILD) [17], and MIL-Instance Selection (MILIS) [18]. The essential idea of these methods is to map each bag into a new instance feature space based on the distances from a subset of instances (termed as instance prototypes, IP) selected from training bags. Therefore, a good IP selection strategy is crucial to the performance of these IP selection-based MIL methods.

The earliest instance selection solution may be dated back to MILES [6]. Actually, MILES does not explicitly give out the selection of instances; instead, the instances participate in the bag feature mapping and the selection is implicitly conducted through feature selection based on ℓ1 norm SVM classifier. Since each instance is used to form the bag-level feature mapping in MILES, the dimensionality of feature vector is very high and given by the total number of instances in training data, which in turn leads to high complexity for both feature computation and ℓ1 norm optimization. In order to reduce the feature dimension, MILIS [18] proposes to use a normalized probability density function (PDF) to model all negative instances in negative bags, then select the most positive and negative instance from each positive and negative bag, respectively. Although MILIS selects much less IPs than MILES, it still has following problems. (1) Since one instance must be selected from each bag, the number of IPs in MILIS is equal to the number of bags, which may still inevitably result in high-dimensional feature vector when facing a large number of bags in training set. (2) Although the most positive (negative) instances in bags are very typical training samples, they cannot represent the distribution of positive (negative) instances very well, and have little effect on the real classification hyperplane. (3) The IPs are devoid of representativeness and compactness, resulting in the information embedded in them is easy to be redundant and sensitive to noise or corruption.

1.2. Our work

In this paper, we propose a novel MIL algorithm based on discriminative subspaces (SubMIL). The proposed algorithm is efficient and suitable for large-scale MIL problems. The method presented here has three major advantages. Firstly, we find two compact and discriminative subspaces before IP selection through the low-rank matrix recovery technique [LR] [19], one for positive instances and the other for negative instances. So the obtained subspaces are more robust to noise and outlier due to the noise removal via low-rank matrix recovery. Second, the ℓ2,1 norm-based self-expressive sparse coding model is used to select the most representative, but not necessarily most positive/negative instances, in each subspace. Compared with IP selection strategy in MILIS that always extracts the most positive (negative) instances, in each subspace. Compared with IP selection strategy in MILIS that always extracts the most positive (negative) instances, the selected instances from our method can represent the distribution of instances in each subspace more effectively. Thirdly, the ℓ2,1 norm-based self-expressive sparse coding model can guarantee the sparsity and stableness of the selected instances. In other words, the number of the selected IPs of our algorithm is much lower than the other method’s and does not obviously grows up with the number of bags/instances in training set. The experiment results on both benchmark and image categorization data sets demonstrate that the proposed approach is superior to the state-of-the-art instance selection strategies for MIL.

The remainder of the paper is organized as follows: Section 2 presents an overview of the proposed SubMIL algorithm; Section 3 gives out the details of each step in SubMIL; the experiments and analysis are give out in Section 4, and Section 5 concludes our paper.

2. Preliminaries and algorithm overview

Before giving out the details of the proposed algorithm, we briefly review the definition of MIL as following. Let X denote the instance space. Given a data set \(\{(x_1, y_1), \ldots, (x_N, y_N)\}\), where \(x_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}\} \subseteq X\) is called a bag and \(y \in \{\pm 1\}\) is the label of the bag X. Here \(x_{ij} \in R^d\) is called an instance with feature dimension of z in the bag X. If there exists \(m \in \{1, \ldots, n_i\}\) such that \(x_{im}\) is a positive instance, then \(X_i\) is a positive bag and \(y_i = 1\); otherwise \(y_i = -1\). Here, the concrete value of m is always unknown. That is, for any positive bag, we can only know that there is at least one positive instance in it, but cannot figure out which ones they are from. Therefore, the goal of multi-instance learning is to learn a classifier to predict the labels of unseen bags. With these ingredients, we give out an overview of our proposed method in Fig. 1. It includes four main steps.

\textbf{Step 1: Initialize instance set:} This step is to initialize the positive instance set \(B^0_+\) and negative instance set \(B^0_-\) (t is the iteration number) from the training set \(\{(x_1, y_1), \ldots, (x_N, y_N)\}\). The initial negative set \(B^0_- = \{x_j | x_j \in X_i \cap y = -1\}\) includes all instances in negative bags but ignores those potential negative ones in positive bags. The positive instance set is initialized to be null set \(B^0_+ = \emptyset\), due to the fact that there is no instance unambiguously assigned as positive. The positive instance set will be updated in the following steps.

\textbf{Step 2: Finding discriminative instance subspace:} By stacking all feature vectors of instances in both \(B^0_+\) and \(B^0_-\) column by column, we can obtain two instance feature matrix as \(B_0^0 \in R^{n^0 \times m^0}\) and \(B_0^n \in R^{n^1 \times m^1}\), where \(n^0\) and \(n^1\) are the number of instances in \(B^0_+\) and \(B^0_-\). Then a novel method based on the low-rank matrix recovery technique [19] is employed to remove the noise and find two discriminative subspaces: \(A_0^0\) for negative instance set and \(A_0^n\) for positive instance sets.

\textbf{Step 3: Selecting representative instance prototypes:} We select the most representative IP subsets \(P_0^0\) and \(P_0^n\) from both subspaces via ℓ2,1 norm-based self-expressive sparse coding model.

\textbf{Step 4: Updating positive instance set:} In this step, given selected IP sets \(P_0^0\) and \(P_0^n\), we fix the negative instance set \(B_0^n\) but update the positive instance set \(B_0^+\). We set \(B_0^+\) untouched due to the fact that all instances in negative bags are necessarily negative. The positive set is updated through selecting the instance that is most likely to belong to the positive subspace from each positive bag via another sparse coding model. The algorithm proceeds between step 2 and step 4 iteratively, until positive instance set converges.

![Fig. 1. Block diagram of the proposed SubMIL algorithm.](attachment:image.png)
Step 5: Bag-level feature mapping and classifier training: After obtaining the final IP sets \(P_0^1\) and \(P_1^1\), we transform each bag into a bag-level feature vector through a similarity-based feature mapping function, and feed all the feature vectors from the training bags along with corresponding labels into a linear SVM to find an optimal classifier.

3. The proposed SubMIL algorithm

3.1. Finding discriminative subspaces

The selected IPs play a key role in the success of IP selection-based MIL. Since they should be compact, representative and clean, it is more reliable if all IPs are selected from class-specific and discriminative subspaces where different classes are well separated. Therefore an effective subspace segmentation algorithm must be helpful in finding suitable IPs. Despite significant progress in subspace segmentation, most popular methods \([20, 21]\) can only exactly solve the problem when the original data is clean and the samples can be strictly drawn from the subspaces. All of these method face lots of challenges when they are applied to segment the instances in MIL into two real underlying subspaces due to the following facts: (1) Noise or corruptions may bias the negative instances’ distribution in MIL; (2) For positive instances in MIL, since there in no unambiguously positive label assigned to any instance, we have to “guess” the most potentially positive instance in each positive bag according to the instance distribution. This “guess” operation inevitably results in some fake positive instances that interfere with subspace segmentation.

In this paper, to combat the noise and ambiguous lab problems, we borrow the strength of the robust Principal Component Analysis (robust PCA) recently proposed by Wright et al. \([19]\) and propose a discriminative low-rank matrix recovery model to segment the discriminative underlying instance subspaces for MIL. The key technique of Robust PCA is low-rank matrix recovery. It seeks to recover a low-rank matrix \(M\) from a highly corrupted matrix \(D = M + E\), where \(M\) implies the clean date in a low dimensional subspace, and \(E\) is the associated sparse noise. More precisely, given the input data matrix \(D\), the robust PCA model can be formulated as \([19]\)

\[
\min_{M, E} \|M\|_* + \lambda \|E\|_1 \quad \text{s.t.} \quad D = M + E,
\]

where the nuclear norm \(\|\|_\ast\|\) (sum of the singular values of a matrix) is a convex relaxation of the matrix rank function, the \(\ell_1\) norm \(\|\|_1\) is to promote sparsity, and the parameter \(\lambda > 0\) is a trade-off between the two items.

In our subspace segmentation method, we decompose the original instance matrix \(B_0^1\) (or \(B_1^1\)) into a clean low-rank matrix \(A_0^1\) (or \(A_1^1\)) and a sparse error matrix \(E_0^1\) (or \(E_1^1\)). In addition, to make sure that the clean matrices \(A_0^1\) and \(A_1^1\) lie in two discriminative subspaces, we add a regularization term to promote the incoherence between the two low-rank matrices, as

\[
\min_{A_0^1, A_1^1, E_0^1, E_1^1} \frac{1}{q} \sum_{i=0}^{q} \|A_i^0\|_* + \lambda \|E_i^1\|_1 + \eta \text{tr}(A_i^0)^T A_i^1), \quad \text{s.t.} \quad B_i^1 = A_i^0 + E_i^1,
\]

where \(1\) is a \(n_1 \times n_0\) matrix, the second term sums up the dot production between each pair of the low-rank matrices \(A_i^0\) and \(A_i^1\), and is penalized by the parameter \(\eta\) balancing the low-rank matrix approximation and matrix incoherence. Since there are two variants \(A_0^1\) and \(A_1^1\) to be determined, we should fix one and optimize the other. To the end, we rewrite Formula (2) as a class-wise optimization problem across different classes as

\[
\min_{A_0^1, A_1^1, E_0^1, E_1^1} \|A_i^0\|_* + \lambda \|E_i^1\|_1 + \eta \text{tr}(A_i^0)^T A_i^1), \quad \text{s.t.} \quad B_i^1 = A_i^0 + E_i^1.
\]

Without ambiguity, we discard the iteration number and introduce a new parameter \(Z_i^1\). Thus, Formula (3) can be converted into the following equivalent problem:

\[
\min_{A_0^1, A_1^1, E_0^1, E_1^1} \|Z_i^1\|_* + \lambda \|E_i^1\|_1 + \eta \text{tr}(A_i^0)^T A_i^1), \quad \text{s.t.} \quad B_i^1 = A_i^0 + E_i^1, Z_i^1 = A_i^1.
\]

To solve the optimization in Formula (4), we extend the Augmented Lagrange Multipliers (ALM) \([22]\) that has been widely used for the standard low-rank problem. The Formula (4) can be solved by the ALM method that minimizes the following augmented Lagrange function:

\[
\min_{A_0^1, A_1^1, E_0^1, E_1^1} \|Z_i^1\|_* + \lambda \|E_i^1\|_1 + \eta \text{tr}(A_i^0)^T A_i^1), \quad \text{s.t.} \quad B_i^1 = A_i^0 + E_i^1, Z_i^1 = A_i^1.
\]

Algorithm 1. Pseudo-code for instance subspace segmentation.

**Input:** Instance matrix \(B_0^1\) and \(B_1^1\), parameter \(\lambda\) and \(\eta\).

1: Initialize \([A_0^0]_t = [A_1^0]_t = 0, [Y_1^0]_t = [Y_2^0]_t = 0, \mu^0 > 0, m = 0;\)
2: While not converged do
3: \(q = m + \frac{1}{2}, k = 0;\)
4: While not converged do
5: \([Z_i^k]_{i+1} = \arg\min_{Z} \|Z_i^k\|_* + \lambda \|E_i^k\|_1, |Y_1^k|_1, |Y_2^k|_1, (\mu^k));\)
6: \([A_0^k]_{i+1} = \arg\min_{A} \|Z_i^k\|_* + \lambda \|E_i^k\|_1, |A_i^0|_1, |Y_1^k|_1, |Y_2^k|_1, (\mu^k));\)
7: \([E_i^k]_{i+1} = \arg\min_{E} \|Z_i^k\|_* + \lambda \|E_i^k\|_1, |A_i^0|_1, E_i^k|_{Y_1^k|_1}, |Y_2^k|_1, (\mu^k));\)
8: \([Y_1^k]_{i+1} = [Y_1^k]_{i+1} + |\mu^k|^2 (B_i^1 - [A_i^0]_t^k - [E_i^k]_{i+1})/|\mu^k|_2;\)
9: \([Y_2^k]_{i+1} = [Y_2^k]_{i+1} + |\mu^k|^2 ([Z_i^k]_{i+1} - [A_i^1]_t^k);\)
10: \(\mu^k = \rho |\mu^k|_2;\)
11: \(k = k + 1;\)
12: End While
13: \(m = m + 1;\)
14: End While
15: Output: \(A_0^1\) and \(A_1^1\).

**Updating** \([Z_i^k]_{i+1}.\) To update \([Z_i^0]_{i+1}\), we have to fix \([A_i^0]_{i+1}\) and \([E_i^0]_{i+1}\) and solve the following problem accordingly:

\[
[Z_i^0]_{i+1} = \arg\min_{Z} \|Z_i^0\|_* + \lambda \|E_i^0\|_1 + \eta \text{tr}(A_i^0)^T A_i^1), \quad \text{s.t.} \quad B_i^1 = A_i^0 + E_i^0, Z_i^1 = A_i^1.
\]

where \(\epsilon = 1/|\mu^0|^2\) and \(X = [A_i^0]_{i+1} + [Y_2^0]_t^k/|\mu^0|^k.\) As suggested by \([23]\),
the solution to the above problem can be solved as
\[ Z^{k+1} = US, V^{T} = UT_{s}(SV^{T}) \]
where \((U, S, V^{T}) = \text{SVD}(X)\),
(7)
where \(\text{SVD}(\cdot)\) is the singular value decomposition (SVD), \(S\) is the singular value matrix of \(X\). The operator \(T_{s}[\cdot]\) is defined by element-wise thresholding of \(S\) [23], i.e., \(T_{s}[S] = \text{diag}(\{t_{s}[s_{1}], t_{s}[s_{2}], \ldots, t_{s}[s_{s}]\})\) for rank of \(S\) being \(r\), and each \(t_{s}[s]\) is determined as
\begin{align*}
t_{s}[s] &= \begin{cases} 
-\epsilon & \text{if } s < -\epsilon, \\
\epsilon & \text{if } s > \epsilon, \\
0 & \text{otherwise.}
\end{cases}
\end{align*}
(8)

\[
\text{updating } (A^{k})^{t+1}: \text{To update the } (A^{k})^{t+1}, \text{we get the derivative of } L \text{ with other variables fixed and obtain the following form:}
\]
\[
\frac{\partial L}{\partial A^{k}} = 0
\]
\[
(A^{k})^{t+1} = \frac{1}{2} \left( (B^{0} - [E]^{k+1} + Z^{k+1}) + \frac{\|Y_{1}^{k+1} + Y_{2}^{k+1} - \eta A^{k} 1\} \|}{\mu^{0}} \right)
\]
\[
\text{updating } (E^{k})^{t+1}: \text{Similar to updating } (Z^{k})^{t+1}, \text{we fix } (A^{k})^{t+1} \text{ and } (Z^{k})^{t+1} \text{ and solve the following problem:}
\]
\[
(E^{k})^{t+1} = \min_{E^{0}} \|E^{0} - Y_{1}^{k} - E^{0} - \eta A^{k} 1\| + \frac{\|\mu^{0} E^{0}\|}{2}
\]
\[
(E^{k})^{t+1} = \min_{E^{0}} \sigma \|E^{0} - Y_{1}^{k} - E^{0} - \eta A^{k} 1\| + \frac{\|\mu^{0} E^{0}\|}{2}
\]
\[
\text{Once } A^{k} \text{ is obtained, we can iteratively optimize } A^{k} \text{ using the similar updating method. The convergence of the two matrixes indicates the termination of the optimization process.}
\]

3.2. Selecting representative IPs

Given the obtained clean instance data lying in two low dimensional subspaces, this section details how to select representative IPs from each subspace. It is worth noting that there are two necessary principles in IP selection: (1) Sparsity. Since the number of selected IPs equals to the dimension of the final bag feature after bag-instance feature mapping [6,18], we should select IPs as few as possible to keep the low dimension. (2) Representativeness. The selected IPs should be representative in corresponding subspace. In other words, for a liner subspace, all samples in the subspace should be able to be linearly presented by the selected IPs.

Following the two principles and inspired by the compact dictionary learning [24], we introduce a self-expressive sparse coding model (SSC) with \(\epsilon_{2,1}\) norm for IP selection. Let \(A_{k}^{q} = [a_{q}^{1}, a_{q}^{2}, \ldots, a_{q}^{m}]^{T} \in R^{m \times n^{q}}\), the goal of the SSC is to find a subset of representative instances with much less number that can best reconstruct all the instances in \(A_{k}^{q}\). It can be formulated as
\[
\min_{C} \sum_{i=1}^{n^{q}} \|a_{q}^{i} - A_{k}^{q} c_{i}\|_{2}^{2} + \gamma \|C\|_{2,1}, \quad C = [c_{1}, \ldots, c_{m}],
\]
(11)
where \(c_{i} \in R^{m}\) is the coefficient matrix \(C \in R^{m \times n^{q}}\) is the reconstruction coefficient of \(a_{q}^{i}\), and the \(\epsilon_{2,1}\) norm \(\|C\|_{2,1}\) is the convex relaxation of \(\|C\|_{2,0}\), to promote row-sparsity of \(C\) [25]. If let \(c_{j}^{*}\) be the jth row vector of \(C\), the rows of \(C\) with higher \(\epsilon_{2,1}\) norm \(\|c_{j}^{*}\|_{2}\) values imply that the corresponding instances in \(A_{k}^{q}\) are more representative. So we reorder the instances \(a_{q}^{1}, a_{q}^{2}, \ldots, a_{q}^{m}\) as \(a_{q}^{1}, a_{q}^{2}, \ldots, a_{q}^{m}\) such that \(\|c_{1}^{*}\|_{2} \geq \|c_{2}^{*}\|_{2} \geq \cdots \geq \|c_{m}^{*}\|_{2}\), so we can select \(n^{q}\) most discriminative instances as IPs
\[
P_{k}^{q} = \{a_{q}^{j} \mid j \leq n^{q}\}
\]
(13)
where \(n^{q}\) is the number of selected positive (or negative) IPs. The immediate advantage of this selection strategy is that the number of selected IPs cannot linearly increase with the scale of instances in training set, because they are only related to instance distribution, rather than instances scale.

3.3. Updating positive instance set

In this subsection, the positive instance set is subsequently updated using the selected IPs. This can be cast as a subspace selection problem for each instance in positive bags. The essence of the update operation is to reselect the instance, which is most likely to belong to extracted positive subspace, from each positive bag to compose the updated positive instance set.

We introduce the sparse coding model again. But this time, we represent all the instances in positive bags using the selected IPs. In particular, we stack all the IPs in \(P_{k}^{q}\) column by column to obtain an IP matrix as \(P_{k}^{q}\). Given a instance \(x_{i}\) from a positive bag \(X_{i}\), it can be sparsely represented as
\[
\min_{a_{i}^{q}} \|x_{i} - P_{k}^{q} a_{i}^{q}\|_{2} + \beta \|a_{i}^{q}\|_{1},
\]
(14)
where \(P_{k}^{q} = [P_{k}^{q}, P_{k}^{q}]\); the first term of Formula (14) is the reconstruction error, and the second term is used to control the sparsity of the coefficients vector \(a_{i}^{q}\) with regularization coefficient \(\beta\). We further decompose \(a_{i}^{q}\) into \(a_{i}^{q+1}\) and \(a_{i}^{q}\) that are the coefficient vectors corresponding to \(P_{k}^{q}\) and \(P_{k}^{q}\), respectively. Now the positive instance set is updated as
\[
B_{k+1}^{q} = \left\{ x_{i} \mid Y_{1} \cap x_{i} = \arg \min_{x_{i}} (d_{i}) \right\}, \quad \text{where } d_{i} = \|x_{i} - P_{k}^{q} a_{i}^{q+1}\|_{2}
\]
(15)
It shows that the new positive instance set \(B_{k+1}^{q}\) includes those instances that are most likely to be near to positive subspace and far from negative subspace.

3.4. Bag feature mapping and classification

After getting the final IPs after iterates in Fig. 1, we then compute the bag-level feature through the similarity-based feature mapping, the same as that in MILES [6] and MILIS [18]. For a bag \(X_{i}\), and the kth IP \(P_{k}^{q}\), the similarity between them is defined as
\[
s(X_{i}, P_{k}^{q}) = \max_{k} x_{i} \cdot \exp(-\gamma \|x_{i} - P_{k}^{q}\|^{2})
\]
(16)
Then, the bag-level feature vector of \(X_{i}\) is constructed based on the similarities between \(X_{i}\) and all the IPs in \(P_{k}^{q}\) as
\[
f_{X_{i}} = [s(X_{i}, P_{k}^{q}), s(X_{i}, P_{k}^{q}), \ldots, s(X_{i}, P_{k}^{q})]^{T}
\]
(17)
where \(n^{q} + 1\) is the total number of selected IPs in \(P_{k}^{q}\). The final stage is to use all the bag-level feature vectors of training bags and corresponding labels to train a linear SVM classifier to test bags' classification.
4. Experiments

The section is to evaluate the proposed algorithm on two different data sets, and compare it with other prevailing IP selection-based and state-of-the-art MIL algorithms.

4.1. Data set

Two data sets are adopted in this paper for evaluating the proposed algorithm. The first data set includes five benchmark subsets that are widely used in the studies of multi-instance learning, including Musk1, Musk2, Elephant, Fox and Tiger. Musk1 contains 47 positive and 45 negative bags, Musk2 contains 39 positive and 63 negative bags, and each of the other three data sets contains 100 positive and 100 negative bags. More details of these five data subsets can be found in [1,4]. The second set is usually used for image categorization, one of the most successful applications of multi-instance learning. It includes two subsets, 1000-Image set and 2000-Image set, which contain ten and twenty categories of COREL images, respectively. Each category has 100 images. Each image is regarded as a bag, and the ROIs (Region of Interests) in the image are regarded as instances described by nine features [6].

4.2. Experiments on benchmark sets

(1) Experiments on performance comparison: In this experiment, we compare our SubMIL with two representative IP selection-based MIL methods (MILES [6] and MILIS [18]) and other prevailing MIL algorithms, including MI-SVM, mi-SVM [4], MissSVM [12], PPM kernel [11], the Diverse Density algorithm [2] and EM-DD [10]. For each algorithm, we repeat 10-fold cross validations 10 times that follow the same procedure as many previous works and use the average test accuracy as the final result. The parameters of each algorithm are determined through cross validation on training data. In addition, for fair comparison, the number of IPs in SubMIL is set to be equal to the number of training bags N, as same as MILIS [18].

Table 1 lists the results of different algorithms. It shows that the performance of the proposed SubMIL is pretty good. It achieves best performance on Musk2, Elephant, Fox and Tiger sets. Furthermore, the proposed SubMIL firstly promotes the performance on Elephant set arriving at 90%. Comparing with other two IP selection-based algorithms, the SubMIL outperforms MILES on all the five sets, outperforms MILIS on Musk1, and is comparable to MILIS on Musk2. It implies that the selected IPs in SubMIL are much more representative and discriminative than those in MILIS and MILES.

(2) Comparison of the number of IPs: According to the results in Table 1, the proposed SubMIL achieves better performance than MILIS and MILES. We further compare the number of IPs used in the SubMIL with those used in the MILIS in the previous experiment, as shown in Fig. 2. The comparison in Fig. 2 indicates that the number of IPs in the SubMIL is much less than the MILIS and is no more than 50% of the number of IPs used in MILIS. Especially, the SubMIL only uses 40 IPs in the fox set, while the MILIS should use 200 IPs. Since the dimension of the final feature of each bag equals to the number of IPs in both SubMIL and MILIS. The less IPs used in SubMIL imply low dimensional features of each bag, which also indicates high efficiency for training and test.

(3) Performance change with the Number of IPs: In order to test the performance change with the number of selected IPs in SubMIL, we set the total number of the positive and negative IPs in SubMIL to be v% of the positive and negative bag numbers, i.e. π0 + π1 = v% × N. We try different value of v from {10, 20, ..., 100}. For each value of v, the average accuracies of 10 times 10-fold cross validations on the five benchmark sets are given in Fig. 3. It is very interesting to notice that the SubMIL with 40% × N IPs can achieve stable and high performance. The performance with v = 40 is even comparable to MILIS and MILES. That is to say, to achieve comparable performance, the number of IPs in SubMIL is only 40% of that in MILIS, which means much lower dimension and lower computation complexity. All of these facts consistently prove that IPs selected by our SubMIL are very discriminative and representative.

(4) Performance comparison between with and without discriminative regularization: In order to make the positive instances and negative instances lie in two discriminative subspaces, the

Table 1
Accuracy (%) on benchmark sets.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Musk1</th>
<th>Musk2</th>
<th>Elephant</th>
<th>Fox</th>
<th>Tiger</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubMIL</td>
<td>90.2</td>
<td>91.3</td>
<td>90.0</td>
<td>66.2</td>
<td>86.3</td>
</tr>
<tr>
<td>MILIS</td>
<td>88.6</td>
<td>91.1</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>MILES</td>
<td>86.3</td>
<td>87.7</td>
<td>86.5</td>
<td>64.7</td>
<td>85.3</td>
</tr>
<tr>
<td>MI-Kernel</td>
<td>88.0</td>
<td>89.3</td>
<td>84.3</td>
<td>60.3</td>
<td>84.2</td>
</tr>
<tr>
<td>MI-SVM</td>
<td>77.9</td>
<td>84.3</td>
<td>81.4</td>
<td>59.4</td>
<td>84.0</td>
</tr>
<tr>
<td>mi-SVM</td>
<td>87.4</td>
<td>83.6</td>
<td>82.0</td>
<td>58.2</td>
<td>78.9</td>
</tr>
<tr>
<td>missSVM</td>
<td>87.6</td>
<td>80.0</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>PPMk</td>
<td>95.6</td>
<td>81.2</td>
<td>82.4</td>
<td>60.3</td>
<td>82.4</td>
</tr>
<tr>
<td>DD</td>
<td>88.0</td>
<td>84.0</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>EMDDD</td>
<td>84.8</td>
<td>84.9</td>
<td>78.3</td>
<td>56.1</td>
<td>72.1</td>
</tr>
</tbody>
</table>
proposed SubMIL uses a discriminative regularization term $\text{tr}(A \Sigma A^T)$ in Eq. (2) to separate the two subspaces. To check the effect of the discriminative regularization, we compare the performance of the SubMIL with it to the performance of the SubMIL without it (i.e. $\eta = 0$) on the benchmark sets. The comparison results are shown in Fig. 4. The SubMIL with the discriminative regularization obtains obvious performance improvements on 4 data sets, which shows that the discriminative regularization can effectively improve the discrimination of the SubMIL.

4.3. Experiments on image categorization sets

The second experiment is conducted on those two image categorization subsets. We use the same experimental routine as that described in [6]. The number of IPs in this experiment is also set as $40\% \times N$. For each data subset, we randomly partition the images within each category in half, and use one group for training and leave the other one for testing. The experiment is repeated five times with five random splits, and the average results are recorded. The overall accuracy as well as 95% confidence intervals is also provided in Table 2. For reference, the table also shows the best results of some other MIL methods.

According to Table 2, the SubMIL outperforms all other algorithms. Compared with MILIS, the SubMIL with much lower dimension has comparable performance on 1000-Image set and better performance on 2000-Image set. The confidence intervals of the SubMIL are always much smaller than the others, which indicates that the SubMIL has a better stableness.

5. Conclusion

Instance prototype (IP) selection has been proven to be powerful for addressing high computation complexity problem in MIL. However, noisy and redundant data, as well as high dimension embedded in the existing IP selection-based MIL algorithms still limit their efficiencies. In this paper, we propose a novel MIL algorithm based on discriminative subspaces (SubMIL) that aims to embed IP selection within subspace segmentation model. The proposed algorithm firstly finds two compact and discriminative subspaces based on low-rank matrix recovery technique; then the $\ell_2,1$ norm based self-expressive sparse coding model is used to select the most representative instances as IPs in each subspace; and each bag is transformed into a bag-level feature based on these selected IPs for final classifier training. The experiments on both benchmark and image categorization data sets demonstrate that the proposed approach has higher performance and lower dimension than the state-of-the-art IP selection-based algorithms.

Fig. 4. Performance comparison between with and without discriminative regularization.

Table 2

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>1000-Image</th>
<th>2000-Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubMIL</td>
<td>83.8:82.8,84.6</td>
<td>72.4:71.1,73.7</td>
</tr>
<tr>
<td>MILIS</td>
<td>83.8:82.5,85.1</td>
<td>70.1:68.5,71.8</td>
</tr>
<tr>
<td>MILES</td>
<td>82.6:81.4,83.7</td>
<td>68.7:67.3,70.1</td>
</tr>
<tr>
<td>MI-Kernel</td>
<td>81.8:80.1,83.6</td>
<td>72.0:71.2,72.8</td>
</tr>
<tr>
<td>MI-SVM</td>
<td>74.7:74.1,75.3</td>
<td>54.6:53.1,56.1</td>
</tr>
<tr>
<td>DD-SVM</td>
<td>81.5:78.5,84.5</td>
<td>67.5:66.1,68.9</td>
</tr>
<tr>
<td>missSVM</td>
<td>78.0:75.8,80.2</td>
<td>65.2:62.0,68.3</td>
</tr>
<tr>
<td>Kmeans-SVM</td>
<td>69.8:67.9,71.7</td>
<td>52.3:51.6,52.9</td>
</tr>
</tbody>
</table>

Acknowledgments

This paper is supported by the following projects: the National Natural Science Foundation of China (Nos. 61271269, 61372148, and 61370038), Beijing Natural Science Foundation (4152016, 4152018), the National Key Technology R&D Program (2014BAK08B02, 2015AHJ55F03), Funding Project for Academic Human Resources Development in Beijing Union University (BPHR2014A04, BPHR2014E02), and the Project of Construction of Innovative Teams and Teacher Career Development for Universities and Colleges under Beijing Municipality (CIT&TCD 20130513, IDHT 20140508).

References


Jiazheng Yuan, born in 1971, received his Ph.D. degree in Computer Science Department from Beijing Jiaotong University, in 2007. He is now a Professor of Software Engineering of Beijing Union University in Beijing of China, his current research interests include graph and image processing, machine learning and artificial intelligence.

Xiankai Huang, born in 1964, received his Ph.D. degree of from Chinese Academy of Sciences in 1999, Professor of Beijing Union University, in Beijing of China, Ph.D. supervisor, his current research interests are e-tourism and artificial intelligence.

Hongzhe Liu, born in 1971, received her Ph.D. degree in Computer Science Department from Beijing Jiaotong University, in 2012. She is now a Professor of Software Engineering of Beijing Union University in Beijing of China, her current research interests include semantic computing, image processing and artificial intelligence.

Bing Li received the Ph.D. degree from the Department of Computer Science and Engineering, Beijing Jiaotong University, China, in 2009. Currently, he is an Associate Professor in the Institute of Automation, Chinese Academy of Sciences. His research interests include color constancy, visual saliency and web content mining.

Weihua Xiong received the Ph.D. degree from the Department of Computer Science, Simon Fraser University, Canada, in 2007. His research interests include color science, computer vision, color image processing, and stereo vision.