2011 11th IEEE International Conference on Data Mining

Context-Aware Multi-Instance Learning based on Hierarchical Sparse Representation

Bing Li
NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
Email: bli@nlpr.ia.ac.cn

Weihua Xiong
OmniVision Technologies, Sunnyvale, CA, USA
Email:wallace.xiong@gmail.com

Weiming Hu
NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
Email: wmhu@nlpr.ia.ac.cn

***Abstract*—Multi-instance learning (MIL), a variant of supervised learning framework, has been applied in many applications. More recently, researchers focus on two important issues for MIL: Instances' contextual structures representation in the same bag and online MIL schemes. In this paper, we present an effective context-aware multi-instance learning technique using a hierarchical sparse representation (HSR-MIL) that addresses the two challenges simultaneously. We firstly construct the inner contextual structure among instances in the same bag based on a novel sparse ε-graph. We then propose a graph kernel based sparse bag classifier through a modified kernel sparse coding in higher-dimension feature space. At last, the HSR-MIL approach is extended to achieve online learning manner with an incremental kernel matrix update scheme. The experiments on several data sets demonstrate that our method has better performances and online learning ability.**

***Keywords*-Context-aware; Multi-Instance Learning; Hierarchical Sparse Representation**

I. Introduction

As a variant of 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, the bag will be considered to be positive if at least one instance in it is positive, and will be considered to be negative if all instances in it are negative.

The first MIL algorithm is proposed to predict the drug molecule activity level [1]. Since then, MIL has been used in many applications, including image categorization [2][3], image retrieval [4], text categorization [5][6], computer security [7], face detection [8][9], visual tracking [18] and computer-aided medical diagnosis [10], etc.

More recently, researchers begin to focus on two important issues of MIL: Instances' contextual structures in the same bag [17]and online learning scheme[18][19]. In this paper, we propose a novel Hierarchical Sparse Representation for Multi-Instance Learning (HSR-MIL) algorithm that addresses these two challenges simultaneously. Specially, the proposed algorithm includes two levels, each being solved through sparse coding[20][21]: one is to obtain contextual structures among instances in the same bag and the other one is to obtain an optimal classifier for the bags. The contributions in this paper include three major parts: (1) A novel sparse ε-graph is proposed to represent the inner structural information in bags. (2) A sparse classifier is defined in higher dimensional space through kernel function on graphs. (3) An online MIL classifier is given out using an incremental kernel matrix update scheme for HSR-MIL. The experiments on several data sets show that our method has better performances and online learning ability.

The remainder of this paper is organized as follows. We briefly review related work in section 2. Section 3 briefly introduces the sparse coding technique. The details of proposed HSR-MIL are given out in Section 4. The experimental results and analysis are reported in Section 5. Section 6 concludes this paper.

II. Related Work

Past decades have witnessed great progress in mathematical models for the MIL problem, from axis-parallel concepts [1] to Diverse Density method [11], k-Nearest Neighbor based algorithm Citation-kNN [13], and Expectation-Maximization version of Diverse Density(EMDD) [12]. In addition, kernel method is also introduced for solving MIL problem. MI-kernel method proposed by Gartner et al [15] regards each bag as a set of feature vectors and then applies set kernel directly for bag classification. Besides these, Andrews et al[5] proposed mi-SVM and MI-SVM through extending Support Vector Machine (SVM). The mi-SVM tries to identify a maximal margin hyperplane for the instances with the constraints that at least one instance of each positive bag locates in the positive half-space; MI-SVM tries to identify a maximal margin hyperplane for the bags by regarding margin of the "most positive instance" in a bag as the margin of that bag. Zhou et al [16] proposed MissSVM method by regarding instances of negative bags as labeled examples while those of positive bags as unlabeled examples with positive constraints. Wang et al [14] proposed the adaptive p-posterior mixture-model(PPMM) kernel by representing each bag as some aggregate posteriors of a mixture model derived on unlabeled data. However, as Zhou

1550-4786/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDM.2011.43

et al[16] indicated, all these MIL algorithms always treated the instances in a bag as independently and identically distributed (i. i. d), which is not true in reality and will inevitably impairs the performance of classification. Therefore, they [17] proposed two multi-instance learning methods, miGraph and MIGrph, which treat the instances non-i. i. d through defining the contextual structure information with ε-graph. We can categorize these two methods as context-aware MIL methods. The better performance are shown to be gained by the structural information in each bag.

Although divers MIL methods have been proposed, they are trained in batch settings, in which whole training set should be available before training procedure begins. But it is not true for many applications, such as object tracking, video understanding, etc. To solve this problem, some online MIL algorithms are recently given out. Babenko et al. [18] proposed an online MI algorithm based on boosting technique, and obtained encouraging object tracking results on several challenging video sequences. However, this online MIL method imposes a strong assumption that all the instances in a positive bag are positive, which can be easily violated in many other practical multi-instance applications. Recently, Li et al [19] extended MILES to an online MIL algorithm. The big weak point of both online methods is the fact that neither of them takes the structural information of instances into account.

The above analysis shows that the existing context-aware MIL methods cannot be trained in online manner, while the existing online MIL methods take no structural information into account. In this paper, we aim to propose a novel MIL classifier that simultaneously takes instances' structural information and online learning scheme into account. To this end, we extend the sparse coding, an efficient technique for many applications, into MIL problem by proposing a novel MIL algorithm based on Hierarchical Sparse Representation (HSR-MIL). In particular, our HSR-MIL builds up a hierarchical graph framework by sparse coding technique to find relationship between instances and optimal classifier for bags.

III. Sparse Coding Review

Because sparse coding is the basis of the proposed algorithm, we start with a brief overview of it. Sparse coding technique recently is widely applied in many practical applications, such as face recognition, image classification, etc[20][21][27]. The goal of sparse coding is to sparsely represent input vectors approximately as a weighted linear combination of a number of "basis vectors". Concretely, given input vector $x \in R^k$ and basis vectors $\mathbf{U} = [u_1, u_2, ..., u_n] \in R^{k \times n}$, the goal of sparse coding is to find a sparse vector of coefficients $\alpha \in R^n$, such that $x \approx \mathbf{U}\alpha = \sum_j u_j \alpha_j$. It equals to solving the following objective.

$$\min_{\alpha} \|x - \mathbf{U}\alpha\|^2 + \lambda \|\alpha\|_0, \tag{1}$$

where $\|\alpha\|_0$ denotes the ℓ^0-norm, which counts the number of nonzero entries in a vector α. But it is well known that the sparsest representation problem is NP-hard in general case, and difficult even to approximate. However, recent results[29][21] show that if the solution is sparse enough, the sparse representation can be recovered by the following convex ℓ^1-norm minimization [29][21] as:

$$\min_{\alpha} \|x - \mathbf{U}\alpha\|^2 + \lambda \|\alpha\|_1, \tag{2}$$

where the first term of Eq(2) is the reconstruction error, and the second term is used to control the sparsity of the coefficients vector α with the ℓ^1 norm. λ is regularization coefficient to control the sparsity of α. The larger λ implies the sparser solution of α. Recently, Lee et al [26] proposed an efficient approximation method, called Feature-Sign Search algorithm (FSS), to solve the optimization in Eq(2). And because $\|x - \mathbf{U}\alpha\|^2 = x^T x + \alpha^T \mathbf{U}^T \mathbf{U} \alpha - 2\alpha^T \mathbf{U}^T x$, FSS only needs the $\mathbf{U}^T\mathbf{U}$ and $\mathbf{U}^T x$, which are the dot product matrix among training samples and the dot product vector between testing vector and training samples respectively, to obtain the optimized sparse coding (more details can be found in [26]).

IV. Hierarchical Sparse Representation for Multi-Instance Learning

Hierarchical sparse representation for multi-instance learning (HSR-MIL) proposed in this paper is based on two-level sparse representation: the first level uses sparse coding to represent the contextual structure among instances in each bag through a sparse ε-graph, and the second one uses sparse coding to build up a classifier among bags by introducing graph kernel function.

Before giving out the details of the algorithm, we briefly review the formal definition of multi-instance learning as following. Let χ denote the instance space. Given a data set $\{(X_1, y_1), ..., (X_i, y_i), ..., (X_N, y_N)\}$, where $X_i = \{x_{i,1}, x_{i,2}, ..., x_{i,n_i}\} \subseteq \chi$ is called a bag and $y_i \in \Psi = \{-1, +1\}$ is the label of bag X_i. Here $x_{i,j} \in R^k$ (suppose each $x_{i,j}$ is normalized to have unit ℓ^2 norm) is called an instance in bag X_i. If there exists $m \in \{1, ..., n_i\}$ such that $x_{i,m}$ 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.

A. Sparse ε-Graph for Bag Inner Structure Representation

The importance of instances structure in MIL has attracted researchers' attention. Zhou et al [17] used the ε-graph [22] to model the local manifold structure among instances in the same bag. Since the ε-graph is from pairwise Euclidean distance and global threshold, it is sensitive

to noises and brings several isolated vertexes easily. On the other hand, intrigued by the research on manifold learning that shows the efficiency of sparse graph in characterizing locality relations for classification purpose, Cheng et al [23] construct a ℓ^1-graph whose edge weights between any two adjacent vertex are from sparse coding. However, locality must lead to sparsity but not necessary vice versa [24][25], i.e., the adjacent vertexes in ℓ^1-graph generated by sparse coding cannot guarantee that they are also near in Euclidean distance metric. Consequently, the ℓ^1-graph can easily result in adjacent vertexes with larger Euclidean distance.

To address the disadvantages from these existing graph techniques, we build a new ε-graph, called "sparse ε-graph", by integrating the advantages of ℓ^1-graph and ε-graph. Comparing with ε-graph, the sparse ε-graph considers the relationship between any two instances locally and adaptively through introducing sparse coding under Euclidean distance constrains.

In the sparse ε-graph, given any instance $x_{i,j}$ and other instances $\mathbf{U} = [x_{i,1}, x_{i,2}, ...x_{i,j-1}, x_{i,j+1}, ..., x_{i,n_i}] \in R^{k\times(n_i-1)}$ in bag X_i , we find a sparse vector of coefficients $\alpha \in R^{n_i-1}$ under a Euclidean distance constrain so that $x_{i,j}$ can be approximated as a weighted linear combination of others. Different from traditional sparse coding, we not only consider the minimization of reconstruction error, but also take Euclidean distances from $x_{i,j}$ to others into account, so the object function is extended from Eq (2) and redefined as:

$$\begin{aligned} &\min_{\alpha} \|x_{i,j} - \mathbf{U}\alpha\|^2 + \lambda\|\mathbf{D}\alpha\|_1 \\ &\mathbf{D} = diag(\|x_{i,j} - x_{i,1}\|, ... \|x_{i,j} - x_{i,j-1}\|, \\ &\|x_{i,j} - x_{i,j+1}\|, ..., \|x_{i,j} - x_{i,n_i}\|) \end{aligned} \tag{3}$$

where the first term of Eq(3) is reconstruction error, the same as that in Eq(2); $\mathbf{D}$ represents the Euclidean distances from $x_{i,j}$ to other instances. So the regularization item $\lambda\|\mathbf{D}\alpha\|_1$ considers both sparsity of and Euclidean distances.

The optimization in Eq(3) is not straightforward. Inspired by solution of Locality-constrained Linear Coding (LLC)[24] , we give out an efficient approximation solution via FSS. Considering that dot products embedded in the $\mathbf{U}^T\mathbf{U}$ and $\mathbf{U}^T x_{i,j}$ in FSS represent the similarities between any two instances, we redefine them by a new calculation $P(x_{i,p}, x_{i,q})$, with a threshold ε to control the locality, shown in Eq(4).

$$P(x_{i,p}, x_{i,q}) = \begin{cases} x_{i,p}^T x_{i,q}, & \|x_{i,p} - x_{i,q}\| \le \varepsilon \\ 0, & \|x_{i,p} - x_{i,q}\| > \varepsilon \end{cases} \tag{4}$$

We can use this new dot product formula $P(x_{i,p}, x_{i,q})$ in the embedded matrix $\mathbf{U}^T\mathbf{U}$ and $\mathbf{U}^T x_{i,j}$ to obtain the sparse code solve α^* in Eq(2) via FSS. The sparse code α^* that considers both sparsity and locality constrains can be viewed as an approximated solution for Eq(3). After getting the sparse code α^*, the sparse ε-graph construction algorithm for each bag in HSR-MIL can be summarized as table 1.

Table I
SPARSE ε-GRAPH CONSTRUCTION FOR EACH BAG.

Algorithm 1 sparse ε-graph construction for each bag.

1: **Input:** A bag in MIL as $X_i = \{x_{i,1}, x_{i,2}, ..., x_{i,n_i}\} \subseteq \chi$, regularization coefficient λ and locality threshold ε

2: For $j = 1 : n_i$ Do
  Set $\mathbf{U} = [X_i \backslash x_{i,j}]$.
  Solve the sparse ε-graph problem $\min_{\alpha} \|x_{i,j} - \mathbf{U}\alpha\|^2 + \lambda\|\mathbf{D}\alpha\|_1$ in Eq(3) by the proposed approximated solution via FSS, and obtain the approximation value of sparse code α^* .
  Set $\alpha* = {|\alpha*|}/{\|\alpha*\|_1}$.
  For $t = 1 : n_i$ Do
    If $t < j$, set $W_{j,t} = \alpha_t^*$;
    If $t == j$, set $W_{j,t} = 1$;
    If $t > j$, set $W_{j,t} = \alpha_{t-1}^*$;
  End
End

3: **Output:** $G = \{X_i, \mathbf{W}\}$ as the inner directed weighted graph with vertex X_i and adjacency weights matrix $\mathbf{W} = \{W_{j,t}\}$.

Obviously, MI-kernel and ℓ^1-graph can be interpreted as the same algorithm applied with different instantiations of threshold ε in the sparse ε-graph framework. If $\varepsilon \le 0$, all the elements in $\mathbf{U}^T\mathbf{U}$ and $\mathbf{U}^T x_{i,j}$ are equal to 0 and α^* is a zero vector. The sparse ε-graph becomes a set of independent instances. The HSR-MIL algorithm will be degenerated into a MI-kernel method without structural information. If $\varepsilon \ge 1$, the $P(x_{i,p}, x_{i,q})$ is equivalent to general dot production, and the sparse ε-graph actually is ℓ^1-graph [23]. If ε is set to be between 0 and 1, λ will be used to indicate sparsity of the edges, the lower λ is, the less sparse the edges will be.

B. Bag Classification based on Graph Kernel Sparse Classifier

After getting sparse ε-graph representation of instances in each bag, the following step is to build second level sparse representation in which each node is a bag with a graph pattern. Consequently, the MIL here can be treated as a graph pattern classification problem. Although there are many existing classifiers, such as SVM [17], they cannot solve imbalance samples and online learning very well. Therefore, we use sparse coding technique again and develop a graph kernel sparse classifier. In comparison with SVM, the sparse classifier is a training free classification scheme. It does not need to learn a model to predict the unseen samples, but directly uses the existing training samples and their corresponding labels to predict the test samples. Moreover, the prediction procedure in sparse classifier is only based on sparse "support" training samples with nonzero coefficients; so it is relatively robust to handle imbalance training samples in classification.

Given a bag data set $\{(X_1, G_1, y_1), ..., (X_i, G_i, y_i), ..., (X_N, G_N, y_N)\}$, where G_i is the sparse ε-graph in bag X_i. Suppose $y_i \in \{1, \ldots, C\}$ is an integer class tag. A test bag with a sparse ε-graph

is also given as (X', G'). Unfortunately, the test graph cannot directly be represented by training bags based on sparse coding as Eq(2). But we can apply a feature mapping function $\varphi : G \rightarrow R^d$ to maps the graph G to a higher dimensional feature space as: $G \rightarrow \varphi(G)$. Thus the basis matrix $\mathbf{U}$ in Eq(2) can be replace by $\mathbf{V} = [\varphi(G_1), \varphi(G_2), ..., \varphi(G_n)]$. And the sparse coding in Eq(2) can be rewritten in high dimensional feature space as :

$$\min_{\beta} \|\varphi(G') - \mathbf{V}\beta\|^2 + \lambda' \|\beta\|_1, \tag{5}$$

where

$$\begin{aligned}
&\|\varphi(G') - \mathbf{V}\beta\|^2 = [\varphi(G')]^T \varphi(G') + \beta^T \mathbf{V}^T \mathbf{V}\beta - 2\beta^T \mathbf{V}^T \varphi(G') \\
&= K(G', G') \\
&+\beta^T \begin{bmatrix} K_g(G_1, G_1) & K_g(G_1, G_2) & ... & K_g(G_1, G_N) \\ K_g(G_2, G_1) & K_g(G_2, G_2) & ... & K_g(G_2, G_N) \\ ... & & & ... \\ K_g(G_N, G_1) & K_g(G_N, G_2) & ... & K_g(G_N, G_N) \end{bmatrix} \beta \\
&-2\beta^T \begin{bmatrix} K_g(G_1, G') \\ K_g(G_2, G') \\ ... \\ K_g(G_N, G') \end{bmatrix} \\
&= 1 + \beta^T \mathbf{K_{VV}} \beta - 2\beta^T \mathbf{K_{V}}_{G'}
\end{aligned} \tag{6}$$

where $K_g()$ is a kernel function that expresses the dot product of graphs in the high dimensional feature space. The $\mathbf{K_{VV}}$ and $\mathbf{K_{V}}_{G'}$ are the key points for solving Eq (5) via FSS, because they represent the correlations and differentials among training bags with different labels. Many existing graph kernel functions can be applied. To compare with Zhou's work [17], we use the same graph kernel function in their work:

$$\begin{aligned}
K_g(G_i, G_j) &= \frac{\sum_{a=1}^{n_i} \sum_{b=1}^{n_j} \omega_{i,a} \omega_{j,b} K(x_{i,a}, x_{j,b})}{\sum_{a=1}^{n_i} \omega_{i,a} \sum_{b=1}^{n_j} \omega_{j,b}} \\
K(x_{i,a}, x_{j,b}) &= \exp\left(-\gamma \|x_{i,a} - x_{j,b}\|^2\right)
\end{aligned}, \tag{7}$$

where $\omega_{i,a} = 1/\sum_{u=1}^{n_i} W_{a,u}^i$, $\omega_{j,b} = 1/\sum_{u=1}^{n_i} W_{b,u}^j$, W^i and W^j are the adjacency weights matrixes for bag X_i and X_j, respectively. In addition, $K(x_{i,a}, x_{j,b})$ is defined using Gaussian radial basis function (RBF) kernel. Once the graph kernel is defined, we can easily calculate the kernel matrix $\mathbf{K_{VV}}$ and $\mathbf{K_{V}}_{G'}$ in Eq(6), then the sparse code of test bag (X', G') can also be obtained as β via FSS. Thus the reconstruction residual of (X', G') in class q is defined as:

$$\begin{aligned}
&r_q(G') = \|\varphi(G') - \mathbf{V}\delta_q(\beta)\|^2 \\
&= 1 + \delta_q(\beta)^T \mathbf{K_{VV}} \delta_q(\beta) - 2\delta_q(\beta)^T \mathbf{K_{V}}_{G'}, \\
&[\delta_q(\beta)]_k = \begin{cases} \beta_k, & y_k = q \\ 0, & y_k \neq q \end{cases}
\end{aligned} \tag{8}$$

where $\delta_q(\beta)$ is a coefficients selector that only selects coefficients associated with class q . The final class c that is assigned to the test bag (X', G') is the one that gives the smallest residual, as:

$$c = \arg\min_q (r_q(G')). \tag{9}$$

C. Online HSR-MIL

In Comparison with other existing online learning algorithms [17, 18], the training free character embedded in the sparse classifier makes it possible to be extended as an online MIL classifier. The proposed online HSR-MIL can not only online update the classifier through learning the new training samples with seen labels, but also online add new classes to the classifier through the new training samples with unseen labels. In addition, the online HSR-MIL with decremental update can immediately forget the training samples or labels that have no use in the future classification. This forgetting ability can avoid obviously impossible misclassification so as to improve the classification performances. This ability is also necessary in many applications, such as forgetting operation in visual tracking.

Considering that the key factors for the graph kernel spare classifier are the kernel matrix $\mathbf{K_{VV}}$ in Eq(6) and the corresponding tag of each training sample, we propose an online training scheme by incrementally updating the kernel matrix, $\mathbf{K_{VV}}$. The accompany advantage is to overcome the runtime limitation, the computation complexity of the kernel matrix $\mathbf{K_{VV}}$ can be reduced from $O(n^2)$ to $O(n)$. The details of update algorithms are given out in Table 2. These update schemes in Table 2 include two operations: incremental update and decremental update. The incremental operation is to update the kernel matrix $\mathbf{K_{VV}}$ with new incoming samples with seen or unseen labels. The decremental operation is to remove the certain samples that should be forgotten from the kernel matrix.

V. Experiments

The experiments in this paper include two parts: the first part includes the experiments on the HSR-MIL with batch training scheme; the second one include the experiments with online HSR-MIL.

A. Date Set

Two popular data sets are adopted in this paper for evaluating the proposed algorithms. The first data set includes five benchmark data sets 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 sets can be found in [1] [5].

The second set is an image categorization set, one of the most successful applications of multi-instance learning. It includes two subsets: 1000-Image set and 2000-Image set that contain ten and twenty categories of COREL images,

Table II
ONLINE UPDATE FOR HSR-MIL.

Algorithm 2 Online update for HSR-MIL.

Incremental Update:

1: **Input:** Existing training bags $\mathbf{B} = [X_1, X_2, ..., X_N]$, corresponding Graphs $\mathbf{G} = [G_1, G_2, ..., G_N]$ and tags $T = [y_1, y_2, ..., y_N]$; the existing kernel matrix $\mathbf{K_{VV}}$.
A new training bag X_{N+1} and its tag y_{N+1} .

2: Compute the inner sparse ε-graph G_{N+1} of the bag X_{N+1} using the sparse ε-graph construction algorithm.

3: For $j = 1 : N$ Do
Compute $K_g(X_i, X_{N+1})$.
Set $K_{N+1} = [K_{N+1}, K_g(X_i, X_{N+1})]$.
End

4: Update: $\mathbf{B} = [\mathbf{B}, X_{N+1}]$, $\mathbf{G} = [G, G_{N+1}]$, $T = [T, y_{N+1}]$ and $\mathbf{K_{VV}} = \begin{bmatrix} \mathbf{K_{VV}} & K_{N+1}^T \\ K_{N+1} & 1 \end{bmatrix}$.

5: **Output:** $\mathbf{B}$, $\mathbf{G}$, T and $\mathbf{K_{VV}}$.

Decremental Update:

1: **Input:** Existing training bags $\mathbf{B} = [X_1, X_2, ..., X_N]$, corresponding Graphs $\mathbf{G} = [G_1, G_2, ..., G_N]$ and tags $T = [y_1, y_2, ..., y_N]$; the existing kernel matrix $\mathbf{K_{VV}}$.
A bag X_p and its tag y_p that will be removed from training set.

2: Update: $\mathbf{B} = \mathbf{B} \backslash X_p$, $\mathbf{G} = \mathbf{G} \backslash G_p$, $T = T \backslash y_p$, and $\mathbf{K_{VV}} = \begin{bmatrix} (\mathbf{K_{VV}})_{1\to p,1\to p} & (\mathbf{K_{VV}})_{1\to p,p+1\to N} \\ (\mathbf{K_{VV}})_{p+1\to N,1\to p} & (\mathbf{K_{VV}})_{p+1\to N,p+1\to N} \end{bmatrix}$.

3: **Output:** $\mathbf{B}$, $\mathbf{G}$, T and $\mathbf{K_{VV}}$.

respectively. Each category of these two image subsets 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 [3] [2].

B. Experiments on HSR-MIL

1) Results on Benchmark Data Sets: In this subsection, we compare HSR-MIL with miGraph, MIGraph and MI-Kernel via repeating 10-fold cross validations ten times through following the same procedure described in [17]. In order to validate the effectivity of the proposed sparse ε-graph, we also use SVM, the same classifier as miGraph, on the sparse ε-graph (denoted as SG-SVM) for bags classification. The same as Zhou's experiment's setting[17], the parameters are determined through cross validation on training sets. The average test accuracy and standard deviations are shown in Table 3. The experimental results of other methods, including MI-SVM and mi-SVM [5], MissSVM [16], PPMM kernel [14], the Diverse Density algorithm[11] and EM-DD [12], are cited from the work of Zhout et al [17].

Table 3 shows that the performance of HSR-MIL is pretty good. It achieves better performances than MIGraph and miGraph on Musk1, Elephant, Fox and Tiger sets. The performances of HSR-MIL, MIGraph, miGraph and MI-Kernel on Musk2 are comparable. In addition, we can notice

Table III
ACCURACY (%) ON BENCHMARK SETS.

Algorithm	Musk1	Musk2	Elephant	Fox	Tiger
HSR-MIL	91.8(±1.7)	88.9(±1.8)	87.5(±0.9)	63.4(±1.5)	86.6(±0.8)
SG-SVM	89.6(±1.5)	88.6(±1.7)	88.4(±1.2)	62.8(±1.4)	87.8(±1.6)
miGraph	88.9(±3.3)	90.3(±2.6)	86.8(±0.7)	61.6(±2.8)	86.0(±1.6)
MIGraph	90.0(±3.8)	90.0(±2.7)	85.1(±2.8)	61.2(±1.7)	81.9(±1.5)
MI-Kernel	88.0(±3.1)	89.3(±1.5)	84.3(±1.6)	60.3(±1.9)	84.2(±1.0)
MI-SVM	77.9	84.3	81.4	59.4	84.0
mi-SVM	87.4	83.6	82.0	58.2	78.9
missSVM	87.6	80.0	N/A	N/A	N/A
PPMM	95.6	81.2	82.4	60.3	82.4
DD	88.0	84.0	N/A	N/A	N/A
EMDD	84.8	84.9	78.3	56.1	72.1

Table IV
ACCURACY (%) ON IMAGE CATEGORIZATION.

Algorithm	1000-Image	2000-Image
HSR-MIL	81.2:[80.8,82.2]	67.7:[66.2,68.4]
SG-SVM	82.8:[81.9,83.2]	69.2:[66.5,69.8]
miGraph	82.4:[80.2,82.6]	70.5:[68.7,72.3]
MIGraph	83.9:[81.2,85.7]	72.1:[71.0,73.2]
MI-Kernel	81.8:[80.1,83.6]	72.0:[71.2,72.8]
MI-SVM	74.7:[74.1,75.3]	54.6:[53.1,56.1]
DD-SVM	81.5:[78.5,84.5]	67.5:[66.1,68.9]
missSVM	78.0:[75.8,80.2]	65.2:[62.0,68.3]
Kmeans-SVM	69.8:[67.9,71.7]	52.3:[51.6,52.9]
MILES	82.6:[81.4,83.7]	68.7:[67.3,70.1]

that the proposed HSR-MIL has lower standard deviations on different benchmark sets, which indicates the stableness of HSR-MIL.

Furthermore, HSR-MIL gains higher performances than SG-SVM on Musk1, Musk2, and Fox sets; but lower performances on Elephant and Tiger sets. This phenomenon implies that the graph kernel sparse classifier is comparable to SVM on the benchmark sets. The performances of SG-SVM are also generally better than miGraph, which indicates that the proposed sparse ε-graph is much more effective than the ε-graph on inner contextual structure representation for MIL in these sets.

2) Results on Image Categorization Sets: The second experiment is conducted on the two image categorization sets. We use the same experimental routine as that described in [2]. For each data set, we randomly partition the images within each category in half, and use one subset 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 4. For reference, the table also shows the best results of some other MIL methods that are given out by Zhou et al. [17]

From table 4, we can find that the SG-SVM has comparable performances to miGraph on 1000-Image and 2000-Image sets, which again validates the effectivity of sparse ε-graph. Although the proposed HSR-MIL has better performances than most MIL methods without structural in-

formation, the accuracy of HSR-MIL is slightly lower than miGraph and SG-SVM on these two sets.

By analyzing and comparing the results in table 3 and table 4, we may obtain an observation that the graph kernel sparse classifier has relatively lower performances than SVM when facing multi-class classification. However, the proposed HSR-MIL, a good alternative MIL method, has many other advantages that will be discussed in the following experiments.

3) Learning with imbalance Samples: We next conduct experiments on robustness of HSR-MIL for imbalance samples. Considering both scale and classification accuracy range of each set in Table 1, Elephant and Tiger sets are selected in this experiment. In each set, we select 20 positive bags and 20 negative bags to compose the test set. The left 80 negative bags are used as negative samples in training set. Then we respectively pick out 10, 20, 30, $\cdots$,80 positive bags from the left 80 positive bags to compose the positive samples in training set. In order to compare the robustness between sparse classifier and SVM, The HSR-MIL and SG-SVM are trained on the training set with 10pos/80neg, 20pos/80neg, $\cdots$, 80pos/80neg samples respectively, and tested on the test set. The experimental results with different rates of positive and negative samples are shown in Fig.1.

Fig. 1 shows that the change ranges of HSR-MIL are [0.70, 0.90] and [0.65, 0.85] on the two sets, while the ranges of SG-SVM are [0.525, 0.905] and [0.50, 0.875]. The performance change ranges of HSR-MIL are much lower than these of SVM. It shows that our HSR-MIL classifier has much more stable accuracy values than SVM with imbalance data sets.

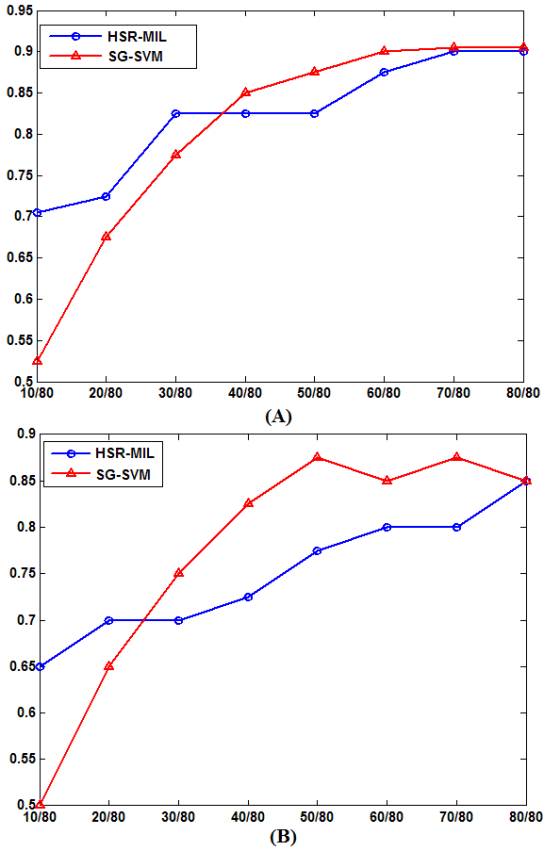

Figure 1. (A) Accuracy with imbalance samples on Elephant set. (B) Accuracy with imbalance samples on Tiger set.

C. Experiments on Online HSR-MIL

In this subsection, we evaluate the online HSR-MIL from three aspects: incremental online training with known labels, incremental online training with new labels and decremental online training.

1) Online HSR-MIL with Known Labels: We use the Elephant and Tiger sets, including 200 samples, to evaluate online HSR-MIL with known labels. Inspired by the experimental setting for online neural networks in [28], we select 20 positive bags and 20 negative bags in each set to compose the test set, and divide the remaining 80 positive and 80 negative bags into 8 training subsets evenly. In each training procedure, a new training subset is added in, and the classification accuracy on the same test set is calculated.

We compare our method with the online MIL algorithm in [18] (referred to be OMIL) on these two data sets. The results shown in Figure 2 indicate that the classification performances of both algorithms are increasing with the growth of training set. And the proposed HSR-MIL is much better. This is because that OMIL is specially based on the hypothesis [18] that nearly all instances in positive bag are positive, which may be right in object tracking, but cannot be satisfied well in general multi-instance problems. In addition, there is no cumulative loss for online HSR-MIL due to its training free character. That is to say, the online HSR-MIL has the same performances to the HSR-MIL with retrain manner.

2) Online HSR-MIL with New Labels: Online learning with new labels is also important for online classifier to many practical applications, such as a new object appearing in the video surveillance. In this experiment, the 1000-image categorization set is used . There are 10 different categories, each of which includes 100 images.We partition all images within each category into half, first 50 images for training and the last 50 images for testing. Now we have 10 training subsets denoted as $\{s_1, s_2, ..., s_{10}\}$ and 10 test subsets denoted as $\{t_1, t_2, ..., t_{10}\}$.

The whole experiment is divided into 9 phases. Initially, the training set is $S = s_1$ and test set is $T = t_1$. In the ith phase ($i = 1...9$), a new training subset s_{i+1} is added

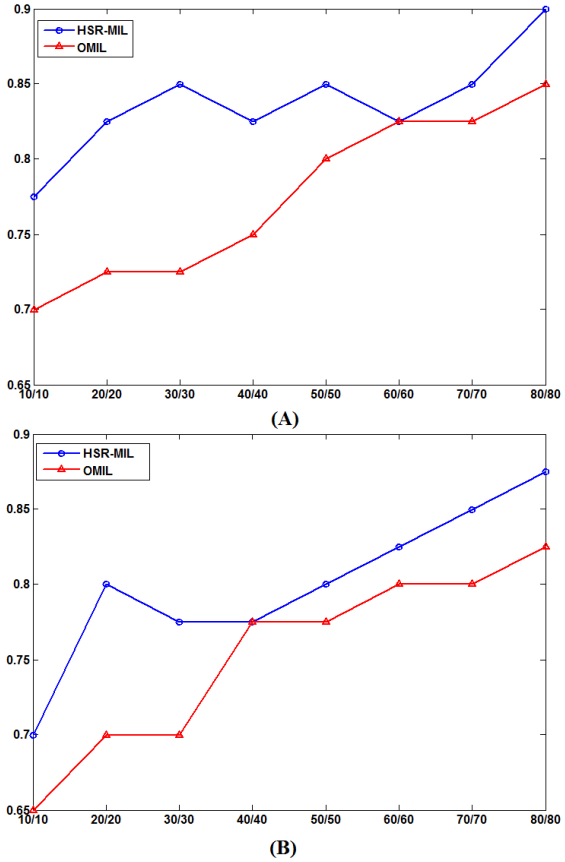

Figure 2. (A) Accuracy of online learning on Elephant set. (B) Accuracy of online learning on Tiger set.

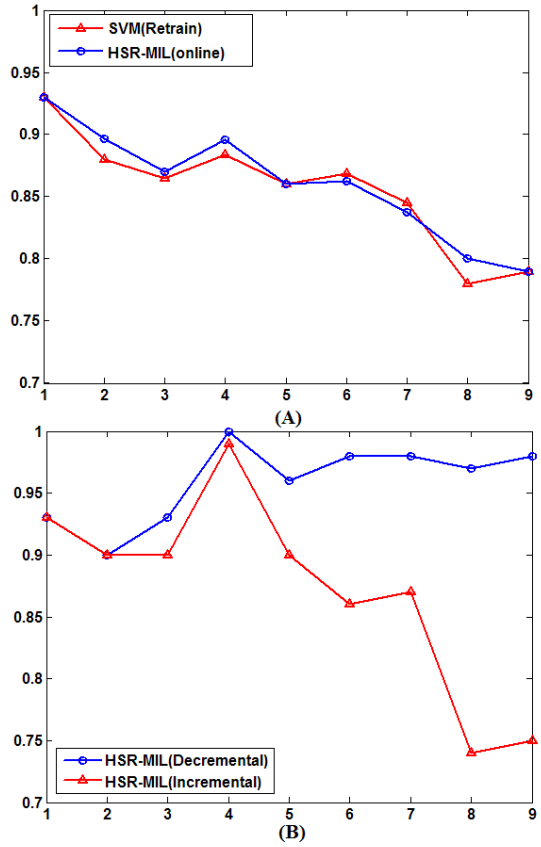

Figure 3. (A) Online learning with new labels. (B) Online learning with decremental training.

into the training set as $S = S \cup s_{i+1}$, and a new test subset t_{i+1} is also added into the test set as $T = T \cup t_{i+1}$. This kind of experimental setting can guarantee that there is always a new added-in label in each phase. To evaluate the classification performance, we also use SVM to retrain the whole training data for classification in each phase. The comparison results between SVM and HSR-MIL are shown in Fig.3(A). According to the experimental results, even though the HSR-MIL learns with online manner and SVM learns with retrain manner, the HSR-MIL is still comparable to SVM. This result also implies the good online learning performance of online HSR-MIL.

3) Online HSR-MIL with Decremental Training: In many practical applications, an online classifier should not only learn new data dynamically, but also "forget" some former samples, such as those samples with the labels that won't appear any more. The final experiment comes from online decremental learning with HSR-MIL. The same as what we have done in the previous experiment, the procedure is also divided into 9 phases. The initial training set is set as $S = s_1$ and test set is set as $T = t_1$. In the ith phase, the test set is set as $T = t_i \cup t_{i+1}$, and a new training subset s_{i+1} is added to training set as $S = S \cup s_{i+1}$. Because the labels of test samples are in either ith or $(i+1)th$ category in each phase, it is better to forget the training samples fall in category 1 to category $i-1$ in order to reduce the obvious misclassification. Consequently, the online HSR-MIL with the decremental update operation given out in algorithm 2 is applied to address this online classification issue. The experimental results of decremental HSR-MIL (denoted as HSR-MIL(Decremental)) and its comparison with online incremental HSR-MIL excluding decremental operation (denoted as HSR-MIL(Incremental)) are shown in figure 3(B). The result tells us that the HSR-MIL with decremental update operation has higher and stable performances, which justifies the necessity of decremental learning in this situation. From the results shown in Fig.3(B), the HSR-MIL without decremental update operation has much lower performances. Furthermore, the performance of HSR-MIL without decremental update rapidly decreases with the new samples coming. This phenomena further implies the necessity of decremental learning in this situation. The performance reduction from HSR-MIL(Incremental) is due

to the misclassification of the labels that no longer appears.

VI. Conclusion

In this paper, we have proposed a novel context-aware multiple instance learning model based on hierarchical sparse representation (HSR-MIL) that aims to simultaneously address instances' structural information and online learning scheme for MIL. To the end, we first give out a novel sparse ε-graph based on sparse coding to represent the interactions between any two instances in a bag. Then, through extending the sparse coding to kernel sparse coding, we present an advanced graph-based sparse classifier for bag classification. Finally, the HSR-MIL is extended to be an dynamically online MIL classifier. We have tested our approach on a wide variety of data sets and studied its online training performances. The experimental results show that our model is superior to most prevailing MIL methods.

Acknowledgment

This work is supported by National Nature Science Foundation of China (No. 61005030, 60935002 and 60825204) and the Excellent SKL Project of NSFC (No.60723005).

References

[1] T. G. Dietterich, R. H. Lathrop and T. Lozano-Perez. Solving the multiple-instance problem with axis-parallel rectangles. Artif. Intell., 89(1-2): 31-71, 1997.

[2] Y. Chen, J. Bi, and J. Z. Wang. MILES: Multiple-instance learning via embedded instance selection. IEEE TPAMI, 28(12), 1931-1947, 2006.

[3] Y. Chen, and J. Z. Wang. Image categorization by learning and reasoning with regions. J. Mach. Learn. Res., 5, 913-939, 2004.

[4] Q. Zhang, W. Yu, S. A. Goldman, J. E. Fritts. Content-based image retrieval using multiple-instance learning. ICML, pages 682-689, 2002.

[5] S. Andrews, I. Tsochantaridis, T. Hofmann. Support vector machines for multiple instance learning. NIPS, pages 561-568, 2003.

[6] B. Settles, M. Craven, S.Ray. Multiple instance active learning. NIPS, pages 1289-1296, 2008.

[7] G. Ruffo. Learning single and multiple instance decision trees for computer security applications. Doctoral dissertation, CS Dept., Univ. Turin, Torino, Italy, 2000.

[8] P. Viola, J. Platt, C. Zhang. Multiple instance boosting for object detection. NIPS, pages 1419-1426, 2006.

[9] C. Zhang, P. Viola. Multiple-instance pruning for learning efficient cascade detectors. NIPS, pages 1681-1688, 2008.

[10] G. Fung, M. Dundar, B. Krishnappuram, R. B. Rao. Multiple instance learning for computer aided diagnosis.NIPS, pages 425-432, 2007.

[11] O. Maron, T. Lozano-Perez. A framework for multiple-instance learning. NIPS, pages 570-576, 1998.

[12] Q. Zhang, S. A. Goldman. EM-DD: An improved multi-instance learning technique. NIPS, pages 1073-1080, 2002.

[13] J. Wang, J. D.Zucker. Solving the multi-instance problem: A lazy learning approach. ICML, pages 1119-1125, 2000.

[14] H. Y. Wang, Q. Yang, H. Zha. Adaptive p-posterior mixture-model kernels for multiple instance learning. ICML, pages 1136C1143, 2008.

[15] T. Gartner,P. A. Flach, A. Kowalczyk, A. J. Smola. Multi-instance kernels. ICML, pages 179-186, 2002.

[16] Z. H. Zhou, J. M. Xu. On the relation between multi-instance learning and semi-supervised learning. ICML, pages 1167-1174, 2007.

[17] Z. Zhou, Y. Sun, and Y. Li. Multi-Instance Learning by Treating Instances As Non-I.I.D. Samples. ICML, pages 1249-1256, 2009.

[18] B. Babenko, Ming-Hsuan Yang, S. Belongie.Visual tracking with online Multiple Instance Learning. CVPR, pages 983-990, 2009.

[19] M. Li, J. Kwok, B. L. Lu. Online Multiple Instance Learning with No Regret. CVPR, pages 1395-1401, 2010.

[20] J. Wright,Y. Ma, J. Mairal, G. Sapiro. Sparse Representation for Computer Vision and Pattern Recognition. the Proceedings of the IEEE, June 2010.

[21] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma.Robust Face Recognition via Sparse Representation. IEEE TPAMI, 31(2), 2009.

[22] J. B. Tenenbaum, V. de Silva, J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. Science, 290, 2319-2323, 2000.

[23] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang. Learn-ing with ℓ^1-Graph for Image Analysis, IEEE TIP,19(4), 858-866, 2010.

[24] K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. NIPS, pages , 2009.

[25] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained Linear Coding for Image Classification, CVPR, pages 1063-6919, 2010.

[26] H. Lee, A. Battle, R. Raina, and Y. Ng. Andrew. Efficient sparse coding algorithms. NIPS, pages , 2006.

[27] A.Yang, J. Wright, Y. Ma, and S. Sastry. Feature selection in face recognition: A sparse representation perspective. UC Berkeley Tech Report UCB/EECS-2007-99, 2007.

[28] R. Polikar, L. Udpa, S. S. Udpa and V. Honavar.Learn++: An Incremental Learning Algorithm for Supervised Neural Networks. IEEE TNN, 31(4), 497-508, 2001.

[29] D. Donoho. For most large underdetermined systems of linear equations the minimal ℓ^1-norm solution is also the sparsest solution. Commun. Pure Appl. Math., 59(7), 797-829, 2004.