# Multi-Instance Learning by Treating Instances As Non-I.I.D. Samples

**Zhi-Hua Zhou** ZHOUZH@LAMDA.NJU.EDU.CN
**Yu-Yin Sun** SUNYY@LAMDA.NJU.EDU.CN
**Yu-Feng Li** LIYF@LAMDA.NJU.EDU.CN
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China

## Abstract

Previous studies on multi-instance learning typically treated instances in the *bags* as *independently and identically distributed*. The instances in a bag, however, are rarely independent in real tasks, and a better performance can be expected if the instances are treated in an non-i.i.d. way that exploits relations among instances. In this paper, we propose two simple yet effective methods. In the first method, we explicitly map every bag to an undirected graph and design a graph kernel for distinguishing the positive and negative bags. In the second method, we implicitly construct graphs by deriving affinity matrices and propose an efficient graph kernel considering the clique information. The effectiveness of the proposed methods are validated by experiments.

## 1. Introduction

In multi-instance learning (Dietterich et al., 1997), each training example is a *bag* of instances. A bag is positive if it contains at least one positive instance, and negative otherwise. Although the labels of the training bags are known, however, the labels of the instances in the bags are unknown. The goal is to construct a learner to classify unseen bags. Multi-instance learning has been found useful in diverse domains such as image categorization (Chen et al., 2006; Chen & Wang, 2004), image retrieval (Zhang et al., 2002) , text categorization (Andrews et al., 2003; Settles et al., 2008), computer security (Ruffo, 2000), face detection (Viola et al., 2006; Zhang & Viola, 2008), computer-aided medical diagnosis (Fung et al., 2007), etc.

A prominent advantage of multi-instance learning mainly lies in the fact that many real objects have inherent structures, and by adopting the multi-instance representation we are able to represent such objects more naturally and capture more information than simply using the flat single-instance representation. For example, suppose we can partition an image into several parts. In contrast to representing the whole image as a single-instance, if we represent each part as an instance, then the partition information is captured by the multi-instance representation; and if the partition is meaningful (e.g., each part corresponds to a region of saliency), the additional information captured by the multi-instance representation may be helpful to make the learning task easier to deal with.

It is obviously not a good idea to apply multi-instance learning techniques everywhere since if the single-instance representation is sufficient, using multi-instance representation just gilds the lily. Even on tasks where the objects have inherent structures, we should keep in mind that the power of multi-instance representation exists in its ability of capturing some structure information. However, as Zhou and Xu (2007) indicated, previous studies on multi-instance learning typically treated the instances in the bags as independently and identically distributed; this neglects the fact that the relations among the instances convey important structure information. Considering the above image task again, treating the different image parts as inter-correlated samples is evidently more meaningful than treating them as unrelated samples. Actually, the instances in a bag are rarely independent, and a better performance can be expected if the instances are treated in an non-i.i.d. way that exploits the relations among instances.

In this paper, we propose two multi-instance learning methods which do not treat the instances as i.i.d.

Appearing in *Proceedings of the 26$^{th}$ International Conference on Machine Learning*, Montreal, Canada, 2009. 

samples. Our basic idea is to regard each bag as an entity to be processed as a whole, and regard instances as inter-correlated components of the entity. Experiments show that our proposed methods achieve performances highly competitive with state-of-the-art multi-instance learning methods.

The rest of this paper is organized as follows. We briefly review related work in Section 2, propose the new methods in Section 3, report on our experiments in Section 4, conclude the paper finally in Section 5.

## 2. Related Work

Many multi-instance learning methods have been developed during the past decade. To name a few, Diverse Density (Maron & Lozano-Pérez, 1998), $k$-nearest neighbor algorithm Citation-$k$NN (Wang & Zucker, 2000), decision trees RELIC (Ruffo, 2000) and MITI (Blockeel et al., 2005), neural networks BP-MIP and RBF-MIP (Zhang & Zhou, 2006), rule learning algorithm RIPPER-MI (Chevaleyre & Zucker, 2001), ensemble algorithms MIBoosting (Xu & Frank, 2004) and MILBoosting (Auer & Ortner, 2004), logistic regression algorithm MI-LR (Ray & Craven, 2005), etc.

Kernel methods for multi-instance learning have been studied by many researchers. Gärtner et al. (2002) defined the MI-Kernel by regarding each bag as a set of feature vectors and then applying *set kernel* directly. Andrews et al. (2003) proposed mi-SVM and MI-SVM. mi-SVM tries to identify a maximal margin hyperplane for the instances with subject to the constraints that at least one instance of each positive bag locates in the positive half-space while all instances of negative bags locate in the negative 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. Cheung and Kwok (2006) argued that the sign instead of value of the margin of the most positive instance was important. They defined a loss function which allowed bags as well as instances to participate in the optimization process, and used the well-formed constrained concave-convex procedure to perform the optimization. Later, Kwok and Cheung (2007) designed marginalized multi-instance kernels by incorporating generative model into the kernel design. Chen and Wang (2004) proposed the DD-SVM method which employed Diverse Density (Maron & Lozano-Pérez, 1998) to learn a set of instance prototypes and then maps the bags to a feature space based on the instance prototypes. Zhou and Xu (2007) proposed the 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. (2008) proposed the PPMM kernel by representing each bag as some aggregate posteriors of a mixture model derived based on unlabeled data.

In addition to classification, multi-instance regression has also been studied (Amar et al., 2001; Ray & Page, 2001), and different versions of generalized multi-instance learning have been defined (Weidmann et al., 2003; Scott et al., 2003). The main difference between standard multi-instance learning and generalized multi-instance learning is that in standard multi-instance learning there is a single concept, and a bag is positive if it has an instance satisfies this concept; while in generalized multi-instance learning (Weidmann et al., 2003; Scott et al., 2003) there are multiple concepts, and a bag is positive only when all concepts are satisfied (i.e., the bag contains instances from every concept). Recently, research on multi-instance semi-supervised learning (Rahmani & Goldman, 2006), multi-instance active learning (Settles et al., 2008) and multi-instance multi-label learning (Zhou & Zhang, 2007) have also been reported. In this paper we mainly work on standard multi-instance learning (Dietterich et al., 1997) and will show that our methods are also applicable to multi-instance regression. Actually it is also possible to extend our proposal to other variants of multi-instance learning.

Zhou and Xu (2007) indicated that instances in a bag should not be treated as i.i.d. samples, and this paper provides a solution. Our basic idea is to regard every bag as an entity to be processed as a whole. There are alternative ways to realize the idea, while in this paper we work by regarding each bag as a graph. McGovern and Jensen (2003) have taken multi-instance learning as a tool to handle relational data where each instance is given as a graph. Here, we are working on propositional data and there is no natural graph. In contrast to having instances as graphs, we regard every bag as a graph and each instance as a node in the graph.

## 3. The Proposed Methods

In this section we propose the MIGraph and miGraph methods. The MIGraph method explicitly maps every bag to an undirected graph and uses a new graph kernel to distinguish the positive and negative bags. The miGraph method implicitly constructs graphs by deriving affinity matrices and defines an efficient graph kernel considering the clique information.

Before presenting the details, we give the formal definition of multi-instance learning as following. Let $\mathcal{X}$ denote the instance space. Given a data set

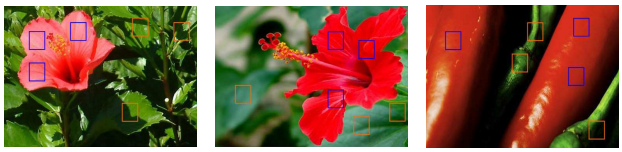

*Figure 1.* Example images with six marked patches each corresponding to an instance

$\{(X_1, y_1), \cdots, (X_i, y_i), \cdots, (X_N, y_N)\}$, where $X_i = \{\boldsymbol{x}_{i1}, \cdots, \boldsymbol{x}_{ij}, \cdots, \boldsymbol{x}_{i,n_i}\} \subseteq \mathcal{X}$ is called a *bag* and $y_i \in \mathcal{Y} = \{-1, +1\}$ is the label of $X_i$, the goal is to generate a learner to classify unseen bags. Here $\boldsymbol{x}_{ij} \in \mathcal{X}$ is an instance $[x_{ij1}, \cdots, x_{ijl}, \cdots, x_{ijd}]'$, $x_{ijl}$ is the value of $\boldsymbol{x}_{ij}$ at the $l$th attribute, $N$ is the number of training bags, $n_i$ is the number of instances in $X_i$, and $d$ is the number of attributes. If there exists $g \in \{1, \cdots, n_i\}$ such that $\boldsymbol{x}_{ig}$ is a positive instance, then $X_i$ is a positive bag and thus $y_i = +1$; otherwise $y_i = -1$. Yet the concrete value of the index $g$ is unknown.

We first explain our intuition of the proposed methods. Here, we use the three example images shown in Figure 1 for illustration. For simplicity, we show six marked patches in each figure, and assume that each image corresponds to a bag, each patch corresponds to an instance in the bag, and the marked patches with the same color are very similar (real cases are of course more complicated, but the essentials are similar as the illustration). If the instances were treated as independent samples then Figure 1 can be abstracted as Figure 2, which is the typical way taken by previous multi-instance learning studies, and obviously the three bags are similar to each other since they contain identical number of very similar instances. However, if we consider the relations among the instances, we can find that in the first two bags the blue marks are very close to each other while in the third bag the blue marks scatter among orange marks, and thus the first two bags should be more similar than the third bag. In this case, Figure 1 can be abstracted by Figure 3. It is evident that the abstraction in Figure 3 is more desirable than that in Figure 2. Here the essential is that, the relation structures of bags belonging to the same class are relatively more similar, while those belonging to different classes are relatively more dissimilar.

Now we describe the MIGraph method. The first step is to construct a graph for each bag. Inspired by (Tenenbaum et al., 2000) which shows that $\epsilon$-graph is helpful for discovering the underlying manifold structure of data, here we establish an $\epsilon$-graph for every bag. The process is quite straightforward. For a bag $X_i$, we regard every instance of it as a node. Then, we compute the distance of every pair of nodes, e.g., $\boldsymbol{x}_{iu}$ and $\boldsymbol{x}_{iv}$. If the distance between

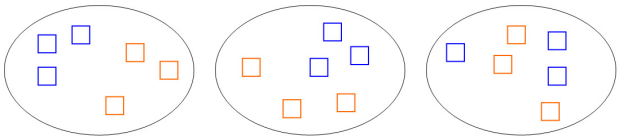

*Figure 2.* If we do not consider the relations among the instances, the three bags are similar to each other since they have identical number of very similar instances

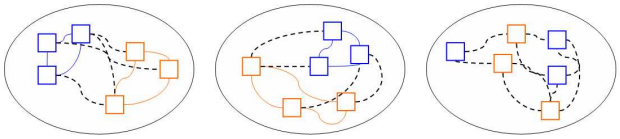

*Figure 3.* If we consider the relations among the instances, the first two bags are more similar than the third bag. Here, the solid lines highlight the high affinity among similar instances

$\boldsymbol{x}_{iu}$ and $\boldsymbol{x}_{iv}$ is smaller than a pre-set threshold $\epsilon$, then an edge is established between these two nodes, where the weight of the edge expresses the affinity of the two nodes (in experiments we use the normalized reciprocal of non-zero distance as the affinity value). Many distance measures can be used to compute the distances. According to the manifold property (Tenenbaum et al., 2000), i.e., a small local area is approximately an Euclidean space, we use Euclidean distance to establish the $\epsilon$-graph. When categorical attributes are involved, we use VDM (Value Difference Metric) (Stanfill & Waltz, 1986) as a complement. In detail, suppose the first $j$ attributes are categorical while the remaining $(d - j)$ ones are continuous attributes normalized to $[0, 1]$. We can use $(\sum_{h=1}^{j} VDM(\boldsymbol{x}_{1,h}, \boldsymbol{x}_{2,h}) + \sum_{h=j+1}^{d} |\boldsymbol{x}_{1,h} - \boldsymbol{x}_{2,h}|^2)^{1/2}$ to measure the distance between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Here the VDM distance between two values $z_1$ and $z_2$ on categorical attribute $Z$ can be computed by

$$VDM(z_1, z_2) = \sum_{c=1}^{C} \left| \frac{N_{Z,z_1,c}}{N_{Z,z_1}} - \frac{N_{Z,z_2,c}}{N_{Z,z_2}} \right|^2, \quad (1)$$

where $N_{Z,z}$ denotes the number of training examples holding value $z$ on $Z$, $N_{Z,z,c}$ denotes the number of training examples belonging to class $c$ and holding value $z$ on $Z$, and $C$ denotes the number of classes.

After mapping the training bags to a set of graphs, we can have many options to build a classifier. For example, we can build a $k$-nearest neighbor classifier that employs graph edit distance (Neuhaus & Bunke, 2007), or we can design a graph kernel (Gärtner, 2003) to capture the similarity among graphs and then solve classification problems by kernel machines such as SVM. The MIGraph method takes the second way, and the idea of our graph kernel is illustrated in Figure 4.

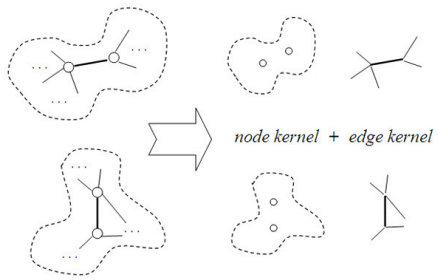


*Figure 4.* Illustration of the graph kernel in MIGraph

Briefly, to measure the similarity between the two bags shown in the left part of Figure 4, we use a *node kernel* (i.e., $k_{node}$) to incorporate the information conveyed by the nodes, use an *edge kernel* (i.e., $k_{edge}$) to incorporate the information conveyed by the edges, and aggregate them to obtain the final graph kernel (i.e., $k_G$). Formally, we define $k_G$ as follows.

**Definition 1** *Given two multi-instance bags $X_i$ and $X_j$ which are presented as graphs $G_h(\{\boldsymbol{x}_{hu}\}_{u=1}^{n_h}, \{\boldsymbol{e}_{hv}\}_{v=1}^{m_h})$, $h = i, j$, where $n_h$ and $m_h$ are the number of nodes and edges in $G_h$, respectively.*

$$k_G(X_i, X_j) = \sum_{a=1}^{n_i}\sum_{b=1}^{n_j} k_{node}(\boldsymbol{x}_{ia}, \boldsymbol{x}_{jb}) + \sum_{a=1}^{m_i}\sum_{b=1}^{m_j} k_{edge}(\boldsymbol{e}_{ia}, \boldsymbol{e}_{jb}), \quad (2)$$

*where $k_{node}$ and $k_{edge}$ are positive semidefinite kernels. To avoid numerical problem, $k_G$ is normalized to*

$$k_G(X_i, X_j) = \frac{k_G(X_i, X_j)}{\sqrt{k_G(X_i, X_i)}\sqrt{k_G(X_j, X_j)}}. \quad (3)$$

The $k_{node}$ and $k_{edge}$ can be defined in many ways. Here we simply define $k_{node}$ using Gaussian RBF kernel as

$$k_{node}(\boldsymbol{x}_{ia}, \boldsymbol{x}_{jb}) = \exp(-\gamma||\boldsymbol{x}_{ia} - \boldsymbol{x}_{jb}||^2), \quad (4)$$

and so the first part of Eq. 2 is exactly the MI-Kernel using Gaussian RBF kernel (Gärtner et al., 2002). $k_{edge}$ is also defined in a form as similar as Eq. 4, by replacing $\boldsymbol{x}_{ia}$ and $\boldsymbol{x}_{jb}$ with $\boldsymbol{e}_{ia}$ and $\boldsymbol{e}_{jb}$, respectively.

Here a key is how to define the feature vector describing an edge. In this paper, for the edge connecting the nodes $\boldsymbol{x}_{iu}$ and $\boldsymbol{x}_{iv}$ of the bag $X_i$, we define it as $[d_u, p_u, d_v, p_v]'$, where $d_u$ is the degree of the node $\boldsymbol{x}_{iu}$, that is, the number of edges connecting $\boldsymbol{x}_{iu}$ with other nodes. Note that it has been normalized through dividing it by the total number of edges in the graph corresponding to $X_i$. $d_v$ is the degree of the node $\boldsymbol{x}_{iv}$, which is defined similarly. $p_u$ is defined as $p_u = w_{uv}/\sum w_{u,*}$, where the numerator is the weight of the edge connecting $\boldsymbol{x}_{iu}$ to $\boldsymbol{x}_{iv}$; $w_{u,*}$ is the weight of the edge connecting $\boldsymbol{x}_{iu}$ to any nodes in $X_i$, thus the denominator is the sum of all the weights connecting with $\boldsymbol{x}_{iu}$. It is evident that $p_u$ conveys information on how important (or unimportant) the connection with the node $\boldsymbol{x}_{iv}$ is for the node $\boldsymbol{x}_{iu}$. $p_v$ is defined similarly for the node $\boldsymbol{x}_{iv}$. The intuition here is that, edges are similar if properties of their ending nodes (e.g., high-degree nodes or low-degree nodes) are similar.

The $k_G$ defined in Eq. 2 is a positive definite kernel and it can be used for any kinds of graphs. The computational complexity of $k_G(X_i, X_j)$ is $O(n_i n_j + m_i m_j)$. The $k_G$ clearly satisfies all the four major properties that should be considered for a graph kernel definition (Borgwardt & Kriegel, 2005).[1] Our above design is very simple, but in the next section we can see that the proposed MIGraph method is quite effective.

A deficiency of MIGraph is that the computational complexity of $k_G$ is $O(n_i n_j + m_i m_j)$, dominated by the number of edges. For bags containing a lot of instances, there will exist a large number of edges and MIGraph will be hard to execute. So, it is desired to have a method with smaller computational cost. For this purpose, we propose the miGraph method which is simple, efficient but effective.

For bag $X_i$, we can calculate the distance between its instances and derive an affinity matrix $W^i$ by comparing the distances with a threshold $\delta$. For example, if the distance between the instances $\boldsymbol{x}_{ia}$ and $\boldsymbol{x}_{iu}$ is smaller than $\delta$, $W^i$'s element at the $a$th row and $u$th column, $w^i_{au}$, is set to 1, and 0 otherwise. There are many ways to derive $W^i$ for $X_i$. In this paper we calculate the distances using Gaussian distance, and set $\delta$ to the average distance in the bag. The key of miGraph, the kernel $k_g$, is defined as follows.

**Definition 2** *Given two multi-instance bags $X_i$ and $X_j$ which contains $n_i$ and $n_j$ instances, respectively.*

$$k_g(X_i, X_j) = \frac{\sum_{a=1}^{n_i}\sum_{b=1}^{n_j} W_{ia} W_{jb} k(\boldsymbol{x}_{ia}, \boldsymbol{x}_{jb})}{\sum_{a=1}^{n_i} W_{ia} \sum_{b=1}^{n_j} W_{jb}}, \quad (5)$$

*where $W_{ia} = 1/\sum_{u=1}^{n_i} w^i_{au}$, $W_{jb} = 1/\sum_{v=1}^{n_j} w^j_{bv}$, and $k(\boldsymbol{x}_{ia}, \boldsymbol{x}_{jb})$ is defined as similar as Eq. 4.*

To understand the intuition of $k_g$, it is helpful to consider that once we have got a good graph, instances in one clique can be regarded as belonging to one concept. To find the cliques is generally expensive for large graphs, while $k_g$ can be viewed as an efficient

[1] We have tried to apply some existing graph kernels directly but unfortunately the results were not good. Due to the page limit, comparison with different graph kernels will be reported in a longer version.

soft version of clique-based graph kernel, where the following principles are evidently satisfied:

- 1) When $W^i = I$, i.e., every two instances do not belong to the same concept, all instances in a bag should be treated equally, i.e., $W_{ia} = 1$ for every instance $\boldsymbol{x}_{ia}$;

- 2) When $W^i = E$ ($E$ is all-one matrix), i.e., all instances belong to the same concept, each bag can be view as one instance and each instance contributes identically, i.e., $W_{ia} = 1/n_i$;

- 3) When $W^i$ is a block matrix, i.e., instances are clustered into cliques each belongs to a concept, $W_{ia} = 1/n_{ia}$ where $n_{ia}$ is size of clique to which $\boldsymbol{x}_{ia}$ belongs. In this case, $k_g$ is exactly an clique-based graph kernel;

- 4) When the value of any entries of $W^i$ increases, for example $w^i_{ab}$, $W_{ia}$ and $W_{ib}$ should decrease since they become more similar, while other $W_{iq}$ ($q \neq a, b$) should not be affected.

It is evident that the computational complexity of $k_g$ is as similar as that of the multi-instance kernel shown in Eq. 4, i.e., $O(n_i n_j)$. Note that once the multi-instance kernel is obtained, the Gaussian distances between every pair of instances have already been calculated, and it is easy to get the $W^i$'s.

## 4. Experiments

### 4.1. Benchmark Tasks

First, we evaluate the proposed MIGraph and miGraph methods on five benchmark data sets popularly used in 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, each of the other three data sets contains 100 positive and 100 negative bags. More details of the data sets can be found in (Dietterich et al., 1997; Andrews et al., 2003).

We compare MIGraph, miGraph with MI-Kernel (Gärtner et al., 2002) via ten times 10-fold cross validation (i.e., we repeat 10-fold cross validation for ten times with different random data partitions). All these methods use Gaussian RBF Kernel and the parameters are determined through cross validation on training sets. The average test accuracy and standard deviations are shown in Table 1[2]. The table also shows the performance of several other multi-instance kernel methods, including MI-SVM and mi-SVM (Andrews et al., 2003), MissSVM (Zhou & Xu, 2007) and PPMM kernel (Wang et al., 2008), and the famous Diverse Density algorithm (Maron & Lozano-Pérez, 1998) and its improvement EM-DD (Zhang & Goldman, 2002). The results of all methods except Diverse Density were obtained via ten times 10-fold cross validation; they were the best results reported in literature and since they were obtained in different studies and the standard deviations were not available, these results are only for reference instead of a rigorous comparison. The best performance on each data set is bolded.

*Table 1.* Accuracy (%) on benchmark tasks

| Algorithm | *Musk1* | *Musk2* | *Elept* | *Fox* | *Tiger* |
|---|---|---|---|---|---|
| MIGraph | 90.0 ±3.8 | 90.0 ±2.7 | 85.1 ±2.8 | 61.2 ±1.7 | 81.9 ±1.5 |
| miGraph | 88.9 ±3.3 | **90.3 ±2.6** | **86.8 ±0.7** | **61.6 ±2.8** | **86.0 ±1.6** |
| 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 |
| EM-DD | 84.8 | 84.9 | 78.3 | 56.1 | 72.1 |

Table 1 shows that the performance of MIGraph and miGraph are quite good. On *Musk1* they are only worse than PPMM kernel; note that the results of PPMM kernel were obtained through an exhaustive search that may be prohibitive in practice (Wang et al., 2008). On *Musk2*, *Elephant* and *Fox* miGraph and MIGraph are respectively the best and second-best algorithms. Pairwise $t$-tests at 95% significance level indicate that miGraph is significantly better than MI-Kernel on all data sets except that on *Musk2* there is no significant difference.

### 4.2. Image Categorization

Image categorization is one of the most successful applications of multi-instance learning. The data sets *1000-Image* and *2000-Image* contain ten and twenty categories of COREL images, respectively, where 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. More details of these data sets can be found in (Chen & Wang, 2004; Chen et al., 2006).

We use the same experimental routine as that described in (Chen et al., 2006). On each data set, we randomly partition the images within each category in half, and use one subset for training while the other

[2]We have re-implemented MI-Kernel since the comparison with MI-Kernel will clearly show whether it is helpful to treat instances as non-i.i.d. samples (this is the only difference between our methods and MI-Kernel). Note that the performance of MI-Kernel in our implementation is better than that reported in (Gärtner et al., 2002).

*Table 2.* Accuracy (%) on image categorization

| Algorithm | *1000-Image* | *2000-Image* |
|---|---|---|
| MIGraph | **83.9** : [**81.2**, **85.7**] | **72.1** : [**71.0**, **73.2**] |
| miGraph | 82.4 : [80.2, 82.6] | 70.5 : [68.7, 72.3] |
| 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] |
| $k$means-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] |

for testing. The experiment is repeated for five times with five random splits, and the average results are recorded. One-against-one strategy is used by MIGraph, miGraph and MI-Kernel for this multi-class task. Following the style of (Chen & Wang, 2004; Chen et al., 2006), we present the overall accuracy as well as 95% confidence intervals in Table 2. For reference, the table also shows the best results of some other multi-instance learning methods reported in literature, including MI-SVM (Andrews et al., 2003; Chen & Wang, 2004), DD-SVM (Chen & Wang, 2004), $k$means-SVM (Csurka et al., 2004), MissSVM (Zhou & Xu, 2007) and MILES (Chen et al., 2006).

It can be found from Table 2 that on the image categorization task our proposed MIGraph and miGraph are highly competitive with state-of-the-art multi-instance learning methods. In particular, MIGraph is the best performed method. This confirms our intuition that MIGraph is a good choice when each bag contains a few instances, and miGraph is better when each bag contains a lot of instances.

By examining the detail results on *1000-Image*, we found that both MIGraph and miGraph or at least one of them are better than MI-Kernel on most categories, except on *African* and *Dinosaurs*. This might owe to the fact that the structure information of examples belonging to these complicated concepts[3] is too difficult to be captured by the simple schemes used in MIGraph and miGraph, while using incorrect structure information is worse than conservatively treating the instances as i.i.d. samples.

For all three methods the largest errors occur between *Beach* and *Mountains* (the full name of this category is *Mountains & glaciers*). This phenomenon has been observed before (Chen & Wang, 2004; Chen et al., 2006; Zhou & Xu, 2007), owing to the fact that many images of these two categories contain semantically related and visually similar regions such as those corresponding to mountain, river, lake and ocean.

[3] *Dinosaurs* is complicated since it contains many different kinds of imaginary animals, toys and even bones.

*Table 3.* Accuracy (%) on text categorization

| Data set | MI-Kernel | miGraph |
|---|---|---|
| *alt.atheism* | $60.2 \pm 3.9$ | $\mathbf{65.5 \pm 4.0}$ |
| *comp.graphics* | $47.0 \pm 3.3$ | $\mathbf{77.8 \pm 1.6}$ |
| *comp.os.ms-windows.misc* | $51.0 \pm 5.2$ | $\mathbf{63.1 \pm 1.5}$ |
| *comp.sys.ibm.pc.hardware* | $46.9 \pm 3.6$ | $\mathbf{59.5 \pm 2.7}$ |
| *comp.sys.mac.harware* | $44.5 \pm 3.2$ | $\mathbf{61.7 \pm 4.8}$ |
| *comp.window.x* | $50.8 \pm 4.3$ | $\mathbf{69.8 \pm 2.1}$ |
| *misc.forsale* | $51.8 \pm 2.5$ | $\mathbf{55.2 \pm 2.7}$ |
| *rec.autos* | $52.9 \pm 3.3$ | $\mathbf{72.0 \pm 3.7}$ |
| *rec.motorcycles* | $50.6 \pm 3.5$ | $\mathbf{64.0 \pm 2.8}$ |
| *rec.sport.baseball* | $51.7 \pm 2.8$ | $\mathbf{64.7 \pm 3.1}$ |
| *rec.sport.hockey* | $51.3 \pm 3.4$ | $\mathbf{85.0 \pm 2.5}$ |
| *sci.crypt* | $56.3 \pm 3.6$ | $\mathbf{69.6 \pm 2.1}$ |
| *sci.electronics* | $50.6 \pm 2.0$ | $\mathbf{87.1 \pm 1.7}$ |
| *sci.med* | $50.6 \pm 1.9$ | $\mathbf{62.1 \pm 3.9}$ |
| *sci.space* | $54.7 \pm 2.5$ | $\mathbf{75.7 \pm 3.4}$ |
| *sci.religion.christian* | $49.2 \pm 3.4$ | $\mathbf{59.0 \pm 4.7}$ |
| *talk.politics.guns* | $47.7 \pm 3.8$ | $\mathbf{58.5 \pm 6.0}$ |
| *talk.politics.mideast* | $55.9 \pm 2.8$ | $\mathbf{73.6 \pm 2.6}$ |
| *talk.politics.misc* | $51.5 \pm 3.7$ | $\mathbf{70.4 \pm 3.6}$ |
| *talk.religion.misc* | $55.4 \pm 4.3$ | $\mathbf{63.3 \pm 3.5}$ |

### 4.3. Text Categorization

The twenty text categorization data sets were derived from the *20 Newsgroups* corpus popularly used in text categorization. Fifty positive and fifty negative bags were generated for each of the 20 news categories. Each positive bag contains 3% posts randomly drawn from the target category and the other instances (and all instances in negative bags) randomly and uniformly drawn from other categories. Each instance is a post represented by the top 200 TFIDF features.

On each data set we run ten times 10-fold cross validation (i.e., we repeat 10-fold cross validation for ten times with different random data partitions). MIGraph does not return results in a reasonable time, and so we only present the average accuracy with standard deviations of miGraph and MI-Kernel in Table 3, where the best result on each data set is bolded.

Pairwise $t$-tests at 95% significance level indicate that, miGraph is significantly better than MI-Kernel on all the text categorization data sets. It is impressive that, by examining the detail results we found that if we consider each time of the ten times 10-fold cross validation, the number of win/tie/lose of miGraph versus MI-Kernel is 10/0/0 on 16 out of the 20 data sets, 9/0/1 on two data sets (*talk.politics.guns* and *talk.religion.misc*), and 7/2/1 on the other two data sets (*alt.atheism* and *misc.forsale*).

### 4.4. Multi-Instance Regression

We also compare MIGraph, miGraph and MI-Kernel on four multi-instance regression data sets, includ-

*Table 4.* Squared loss on multi-instance regression tasks

| Algorithm | LJ160.1 | LJ160.1S | LJ80.1 | LJ80.1S |
|---|---|---|---|---|
| MIGraph | **0.0080** | 0.0112 | **0.0111** | 0.0154 |
| miGraph | 0.0084 | **0.0094** | 0.0118 | **0.0113** |
| MI-Kernel | 0.0116 | 0.0127 | 0.0174 | 0.0219 |
| DD | 0.0852 | 0.0052 | N/A | 0.1116 |
| BP-MIP | 0.0398 | 0.0731 | 0.0487 | 0.0752 |
| RBF-MIP | 0.0108 | 0.0075 | 0.0167 | 0.0448 |

ing LJ-160.166.1, LJ-160.166.1-S, LJ-80.166.1 and LJ-80.166.1-S (abbreviated as LJ160.1, LJ160.1S, LJ80.1 and LJ80.1S, respectively). In the name LJ-*r.f.s*, $r$ is the number of relevant features, $f$ is the number of features, and $s$ is the number of *scale factors* used for the relevant features that indicate the importance of the features. The suffix $S$ indicates that the data set uses only labels that are not near 1/2. More details of these data sets can be found in (Amar et al., 2001).

We perform leave-one-out tests and report the results in Table 4. For reference, the table also shows the leave-one-out results of some other methods reported in literature, including Diverse Density (Maron & Lozano-Pérez, 1998; Amar et al., 2001), BP-MIP and RBF-MIP (Zhang & Zhou, 2006). In Table 4 the best performance on each data set is bolded. It is evident that our proposed miGraph and MIGraph methods also work well on multi-instance regression tasks.

## 5. Conclusion

Previous studies on multi-instance learning typically treated instances in the bags as i.i.d. samples, neglecting the fact that instances within a bag are extracted from the same object, and therefore the instances are rarely i.i.d. intrinsically and the relations among instances may convey important information. In this paper, we propose two methods which treat the instances in an non-i.i.d. way. Experiments show that our proposed methods are simple yet effective, with performances highly competitive with the best performing methods on several multi-instance classification and regression tasks. Note that our methods can also handle i.i.d. samples by using identity matrix.

An interesting future issue is to design a better graph kernel to capture more useful structure information of multi-instance bags. Applying graph edit distance or metric learning methods to the graphs corresponding to multi-instance bags is also worth trying. The success of our proposed methods also suggests that it is possible to improve other multi-instance learning methods by incorporating mechanisms to exploit the relations among instances, which opens a promising future direction. Moreover, it is possible to extend our proposal to other settings such as generalized multi-instance learning, multi-instance semi-supervised learning, multi-instance active learning, multi-instance multi-label learning, etc.

## Acknowledgments

Supported by NSFC (60635030, 60721002), JiangsuSF (BK2008018) and Jiangsu 333 Program.

## References

Amar, R. A., Dooly, D. R., Goldman, S. A., & Zhang, Q. (2001). Multiple-instance learning of real-valued data. *Proc. 18th Intl. Conf. Mach. Learn.* (pp. 3–10).

Andrews, S., Tsochantaridis, I., & Hofmann, T. (2003). Support vector machines for multiple-instance learning. In *Adv. Neural Inf. Process. Syst. 15*, 561–568. Cambridge, MA: MIT Press.

Auer, P., & Ortner, R. (2004). A boosting approach to multiple instance learning. *Proc. 15th Eur. Conf. Mach. Learn.* (pp. 63–74).

Blockeel, H., Page, D., & Srinivasan, A. (2005). Multi-instance tree learning. *Proc. 22nd Intl. Conf. Mach. Learn.* (pp. 57–64).

Borgwardt, K. M., & Kriegel, H.-P. (2005). Shortest-path kernels on graphs. *Proc. 5th IEEE Intl. Conf. Data Min.* (pp. 74–81).

Chen, Y., Bi, J., & Wang, J. Z. (2006). MILES: Multiple-instance learning via embedded instance selection. *IEEE Trans. Patt. Anal. Mach. Intell.*, *28*, 1931–1947.

Chen, Y., & Wang, J. Z. (2004). Image categorization by learning and reasoning with regions. *J. Mach. Learn. Res.*, *5*, 913–939.

Cheung, P.-M., & Kwok, J. T. (2006). A regularization framework for multiple-instance learning. *Proc. 23rd Intl. Conf. Mach. Learn.* (pp. 193–200).

Chevaleyre, Y., & Zucker, J.-D. (2001). A framework for learning rules from multiple instance data. *Proc. 12th Eur. Conf. Mach. Learn.* (pp. 49–60).

Csurka, G., Bray, C., Dance, C., & Fan, L. (2004). Visual categorization with bags of keypoints. *ECCV Workshop on Statistical Learning in Computer Vision* (pp. 59–74).

Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple-instance problem with axis-parallel rectangles. *Artif. Intell.*, *89*, 31–71.

Fung, G., Dundar, M., Krishnappuram, B., & Rao, R. B. (2007). Multiple instance learning for computer aided diagnosis. In *Adv. Neural Inf. Process. Syst. 19*, 425–432. Cambridge, MA: MIT Press.

Gärtner, T. (2003). A survey of kernels for structured data. *SIGKDD Explorations*, *5*, 49–58.

Gärtner, T., Flach, P. A., Kowalczyk, A., & Smola, A. J. (2002). Multi-instance kernels. *Proc. 19th Intl. Conf. Mach. Learn.* (pp. 179–186).

Kwok, J. T., & Cheung, P.-M. (2007). Marginalized multi-instance kernels. *Proc. 20th Intl. J. Conf. Artif. Intell.* (pp. 901–906).

Maron, O., & Lozano-Pérez, T. (1998). A framework for multiple-instance learning. In *Adv. Neural Inf. Process. Syst. 10*, 570–576. Cambridge, MA: MIT Press.

McGovern, A., & Jensen, D. (2003). Identifying predictive structures in relational data using multiple instance learning. *Proc. 20th Intl. Conf. Mach. Learn.* (pp. 528–535).

Neuhaus, M., & Bunke, H. (2007). A quadratic programming approach to the graph edit distance problem. *Proc. 6th IAPR Workshop on Graph-based Represent. in Patt. Recogn.* (pp. 92–102).

Rahmani, R., & Goldman, S. A. (2006). MISSL: Multiple-instance semi-supervised learning. *Proc. 23rd Intl. Conf. Mach. Learn.* (pp. 705–712).

Ray, S., & Craven, M. (2005). Supervised versus multiple instance learning: An empirical comparison. *Proc. 22nd Intl. Conf. Mach. Learn.* (pp. 697–704).

Ray, S., & Page, D. (2001). Multiple instance regression. *Proc. 18th Intl. Conf. Mach. Learn.* (pp. 425–432).

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

Scott, S. D., Zhang, J., & Brown, J. (2003). *On generalized multiple-instance learning* (Technical Report UNL-CSE-2003-5). CS Dept., Univ. Nebraska, Lincoln, NE.

Settles, B., Craven, M., & Ray, S. (2008). Multiple-instance active learning. In *Adv. Neural Inf. Process. Syst. 20*, 1289–1296. Cambridge, MA: MIT Press.

Stanfill, C., & Waltz, D. (1986). Toward memory-based reasoning. *Comm. ACM*, *29*, 1213–1228.

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*, 2319–2323.

Viola, P., Platt, J., & Zhang, C. (2006). Multiple instance boosting for object detection. In *Adv. Neural Inf. Process. Syst. 18*, 1419–1426. Cambridge, MA: MIT Press.

Wang, H.-Y., Yang, Q., & Zha, H. (2008). Adaptive p-posterior mixture-model kernels for multiple instance learning. *Proc. 25th Intl. Conf. Mach. Learn.* (pp. 1136–1143).

Wang, J., & Zucker, J.-D. (2000). Solving the multi-instance problem: A lazy learning approach. *Proc. 17th Intl. Conf. Mach. Learn.* (pp. 1119–1125).

Weidmann, N., Frank, E., & Pfahringer, B. (2003). A two-level learning method for generalized multi-instance problem. *Proc. 14th Eur. Conf. Mach. Learn.* (pp. 468–479).

Xu, X., & Frank, E. (2004). Logistic regression and boosting for labeled bags of instances. *Proc. 8th Pac.-Asia Conf. Knowl. Discov. Data Min.* (pp. 272–281).

Zhang, C., & Viola, P. (2008). Multiple-instance pruning for learning efficient cascade detectors. In *Adv. Neural Inf. Process. Syst. 20*, 1681–1688. Cambridge, MA: MIT Press.

Zhang, M.-L., & Zhou, Z.-H. (2006). Adapting RBF neural networks to multi-instance learning. *Neural Process. Lett.*, *23*, 1–26.

Zhang, Q., & Goldman, S. A. (2002). EM-DD: An improved multi-instance learning technique. In *Adv. Neural Inf. Process. Syst. 14*, 1073–1080. Cambridge, MA: MIT Press.

Zhang, Q., Yu, W., Goldman, S. A., & Fritts, J. E. (2002). Content-based image retrieval using multiple-instance learning. *Proc. 19th Intl. Conf. Mach. Learn.* (pp. 682–689).

Zhou, Z.-H., & Xu, J.-M. (2007). On the relation between multi-instance learning and semi-supervised learning. *Proc. 24th Intl. Conf. Mach. Learn.* (pp. 1167–1174).

Zhou, Z.-H., & Zhang, M.-L. (2007). Multi-instance multi-label learning with application to scene classification. In *Adv. Neural Inf. Process. Syst. 19*, 1609–1616. Cambridge, MA: MIT Press.