Int. J. Appl. Math. Comput. Sci., *2014, Vol. 24, No. 3, 567–577*
DOI: 10.2478/amcs-2014-0041

MULTIPLE–INSTANCE LEARNING WITH PAIRWISE INSTANCE SIMILARITY

LIMING YUAN, JIAFENG LIU, XIANGLONG TANG

School of Computer Science and Technology
Harbin Institute of Technology, Harbin 150001, China
e-mail: yuanleeming@163.com,{jefferyliu,tangxl}@hit.edu.cn

Multiple-Instance Learning (MIL) has attracted much attention of the machine learning community in recent years and many real-world applications have been successfully formulated as MIL problems. Over the past few years, several Instance Selection-based MIL (ISMIL) algorithms have been presented by using the concept of the embedding space. Although they delivered very promising performance, they often require long computation times for instance selection, leading to a low efficiency of the whole learning process. In this paper, we propose a simple and efficient ISMIL algorithm based on the similarity of pairwise instances within a bag. The basic idea is selecting from every training bag a pair of the most similar instances as instance prototypes and then mapping training bags into the embedding space that is constructed from all the instance prototypes. Thus, the MIL problem can be solved with the standard supervised learning techniques, such as support vector machines. Experiments show that the proposed algorithm is more efficient than its competitors and highly comparable with them in terms of classification accuracy. Moreover, the testing of noise sensitivity demonstrates that our MIL algorithm is very robust to labeling noise.

Keywords: multiple-instance learning, instance selection, similarity, support vector machines.

1. Introduction

The term *Multiple-Instance Learning* (MIL) was first coined by Dietterich *et al.* (1997). In this learning framework, the training set is composed of labeled *bags*, and each of which consists of one or more unlabeled instances. A bag is positively labeled if it contains at least one positive instance, otherwise it is negatively labeled. The aim of an MIL predictor is to learn some target concepts from the training set for correctly labeling unseen bags.

The superiority of MIL over the traditional supervised learning ascribes to the fact that it only requires the label information of bags rather than that of individual instances in them for training. Compared with supervised learning, MIL is thus more suitable for particular applications, such as drug activity prediction (Dietterich *et al.*, 1997), stock selection (Maron and Lozano-Pérez, 1998), natural scene classification (Maron and Ratan, 1998), computer aided diagnosis (Fung *et al.*, 2007; Raykar *et al.*, 2008), Content-Based Image Retrieval (CBIR) (Yang and Lozano-Pérez, 2000; Zhang *et al.*, 2002; Rahmani *et al.*, 2008; Zha *et al.*, 2008), semantic segmentation (Vezhnevets and Buhmann, 2010), action recognition (Ali and Shah, 2010), as well as object detection (Viola *et al.*, 2006; Dollár *et al.*, 2008; Babenko *et al.*, 2011a) and tracking (Babenko *et al.*, 2009; 2011b; Li *et al.*, 2010). Take the CBIR problem as an example. An image is often represented as a set of localized regions extracted from the image. However, only those carrying category-specific information are regions of interest for the purpose of classification, whereas other regions have random features and thus possess no discriminative ability. Hence the CBIR issue can be naturally formalized as an MIL problem, where each image corresponds to a bag and each image region corresponds to an instance in the bag.

In recent years, several Instance Selection-based MIL (ISMIL) algorithms have been presented by using the concept of the embedding space, namely, DD-SVM (Chen and Wang, 2004), MILES (Chen *et al.*, 2006), MILD (Li and Yeung, 2010) and MILIS (Fu *et al.*, 2011). The main idea can be summarized as follows. First, an ISMIL algorithm applies an instance selection approach to select some representative instance prototypes from the training set. These instance prototypes thus form a new feature space named the embedding space. Then, a training bag is represented as a single feature vector by mapping it into the embedding space, and thus any standard supervised learning technique can be employed for classification. Finally, the ISMIL algorithm uses the new bag-level feature vectors to learn a standard Support Vector Machine (SVM). Even though these ISMIL algorithms can convey promising performance, they usually require much computation time to complete the instance selection process, especially for large-scale data sets, leading to high computational complexity for the whole learning process. As we know, computational efficiency is an important issue for practical applications (Czarnowski and Jędrzejowicz, 2011), so it is necessary to design an efficient instance selection approach to speed up the learning process of the ISMIL algorithm while not sacrificing its generalization ability.

In this paper, we propose a novel ISMIL algorithm based on an efficient instance selection approach, which is inspired by the similarity of pairwise instances within a bag. We call it Multiple-Instance Learning with Pairwise Instance Similarity (MILPIS). The basic idea is choosing from every training bag a pair of the most similar instances as instance prototypes and using all the instance prototypes to form the embedding space. Then a standard SVM is learned using all the new bag-level features for training bags derived by mapping theses bags into the embedding space. The main contributions of this paper can be summarized as follows. First, MILPIS searches for small clusters of target concepts in the feature space, so it has provided highly comparative classification accuracy with other ISMIL algorithms. Second, MILPIS performs instance selection by only considering the structure within a bag, so it can accomplish the instance selection process more quickly, and thus the whole learning process is faster. Finally, the testing of noise sensitivity shows that our MILPIS algorithm is very robust to labeling noise. This is also due to the fact that MILPIS focuses on the inner structure of a bag and is thus not influenced by the label of the bag.

The rest of this paper is organized as follows. In Section 2 we give an overview of some work related to our research. In Section 3 we present our MILPIS algorithm and analyze the computational complexities of MILPIS and four previous ISMIL algorithms including DD-SVM, MILES, MILD and MILIS. Section 4 provides a comparative analysis of MILPIS with other ISMIL algorithms using three MIL tasks, i.e., drug activity prediction, automatic image annotation and region-based image categorization. We give conclusions and outline some future work in the last section of the paper.

2. Related work

MIL has become an important learning framework in the machine learning community since it was first proposed by Dietterich *et al*. (1997). Many algorithms have been presented to tackle this new learning problem. Dietterich *et al*. (1997) developed the first MIL algorithm named the Axis-Parallel Rectangle (APR). They assume that there may exist an APR in the feature space that includes at least one instance from each positive bag and excludes all instances from the negative bags. A bag will be labeled positive if one of its instances falls within the APR, otherwise it will be labeled negative. Later on, Maron and Lozano-Pérez (1998) used a new concept of Diverse Density (DD) to solve MIL problems. Similarly to the APR, DD can be used to evaluate how many different positive bags possess instances near a point in the feature space and how far the negative instances are from that point. Following their work, Zhang and Goldman (2002) incorporated the concept of DD into the Expectation Maximization (EM) framework in order to learn the target concept in a more efficient manner. Rahmani *et al*. (2008) employed the quasi-Newton method to iteratively locate multiple target concepts from diverse initial locations, since some real-world applications own the characteristics of multi-modal distributions.

Together with these classical MIL algorithms, many researchers focused on the adaptation of the standard supervised learning techniques to the MIL scenario. Ramon and De Raedt (2000) adapted neural networks (Trawiński *et al*., 2012) to the MIL setting via taking into account the relation of a bag to its instances. Zhang and Zhou (2004) later derived a similar framework. Wang and Zucker (2000) presented two MIL versions of k-Nearest Neighbor (kNN) algorithms by using the Hausdorff distance to compute the distances between different bags, namely, citation-kNN and Bayesian-kNN. Gärtner *et al*. (2002) designed a special kernel function for multiple-instance data such that SVMs can be learned directly from the training bags. Several years later, Tao *et al*. (2008) explored specialized kernels for MIL and applied them to the generalized MIL setting.

Andrews *et al*. (2003) treated the unobservable instance labels as hidden variables and formulated MIL as mixed integer quadratic programs. They developed two MIL algorithms: mi-SVM, used for the instance-level classification, and MI-SVM, used for the bag-level one. Andrews and Hofmann (2004) proposed an algorithm based on a generalization of linear programming boosting. In the same year, Auer and Ortner (2004) proposed a boosting-based algorithm that built an ensemble of weak hypotheses, each of which is either a hyper-ball or a hyper-rectangle. Settles *et al*. (2008) presented a framework for active learning in the MIL scenario and demonstrated that learning from instance labels can significantly improve the performance of a basic MIL algorithm. Li and Sminchisescu (2010) made an attempt towards a convex formulation for MIL and introduced convex constraints on the likelihood ratio between the positive and negative classes for each instance. Thus, MIL can be converted to a convex joint estimation of the likelihood ratio function and the likelihood ratio values on training instances. Inspired by subgradient-based methods for SVMs, Bergeron *et al.* (2012) introduced a nonconvex bundle algorithm to optimize the multiple-instance objective directly. Li *et al.* (2013) assume that instances are drawn from a mixture distribution of the concept and non-concept. With this assumption the classification of a bag can be regarded as a classifier combining problem, which combines the classification results of all instances in the bag. Nguyen *et al.* (2013) provided a generic framework for transforming rule-based algorithms to solve MIL problems.

Over the past ten years, several researchers have combined the instance selection techniques with the concept of the embedding space to solve MIL problems and presented several ISMIL algorithms, namely, DD-SVM, MILES, MILD and MILIS. The main difference among them is how to select instance prototypes from the training set. Specifically, DD-SVM depends on the DD concept to identify instance prototypes. Those instances corresponding to local maximizers of the DD function in the feature space are chosen as instance prototypes. Then an SVM with a Gaussian kernel is learned in the embedding space. MILES regards all training instances as the initial instance prototypes and performs the instance selection implicitly by learning a 1-norm SVM with a linear kernel. MILD performs the instance selection based on a conditional probability model. The instance having the highest ability to distinguish between positive and negative training bags is chosen from each positive bag as an instance prototype. As in DD-SVM, MILD then learns a standard SVM with a Gaussian kernel using bag-level features for training bags. MILIS achieves the initial instance selection by modeling the distributions of the negative population with the Gaussian-kernel-based kernel density estimator. Then it depends on an iterative optimization framework to update instance prototypes and learn a linear SVM.

3. MILPIS: Multiple-instance learning with pairwise instance similarity

3.1. Motivation.

A general assumption made by many MIL algorithms is that positive instances often form one or more clusters in the feature space. Two such well-known algorithms are the APR and DD, which attempt to search the whole feature space for a target concept region or cluster that contains at least one instance from every positive bag and none of negative instances. Furthermore, the experimental analysis done by Maron and Lozano-Pérez (1998) as well as Zhang *et al.* (2002) revealed that the DD framework can learn certain simple concepts of nature scenes. From the perspective of instance selection, the DD concept can measure the co-occurrence of similar instances from different bags with the same label. In other words, an instance prototype learned from DD represents a certain simple concept or a class of instances that is more likely to appear in bags with the specific label than in other bags. This is one of the reasons why the DD-SVM algorithm has succeeded with regard to instance selection, although it tries to search for multiple instance prototypes (target concepts) starting from a new instance in each iteration. Nevertheless, DD-SVM needs to compute the distances between almost all pairs of instances in the training set for a single instance prototype, since it uses the quasi-Newton method to locate the target concept. Therefore, DD-SVM is very time-consuming, which has been demonstrated by the experimental results given by Chen *et al.* (2006) as well as Li and Yeung (2010).

Following the above analysis, one may argue that clustering methods could be applied for the purpose of capturing the target concepts or clusters of positive instances. As indicated by Fu *et al.* (2011), a principled way should be devoid of the clustering or quantization procedure for an ISMIL algorithm, since clustering and quantization do not consider the bag-level structure or discriminative information, and thus may discard small clusters in the feature space where informative features may be located.

Based on the above discussions, two main points can be summarized on how to search for the target concepts (instance prototypes) for the instance selection approach existent in an ISMIL algorithm. First, the searching process should be efficient enough since we are often faced with large-scale data sets in real-world applications. Second, small clusters should not be discarded as far as possible by the instance selection approach since they may contain informative features.

3.2. MILPIS.

To satisfy the above two constraints, we propose an MILPIS algorithm based on a similarity between pairwise instances within a bag. In the following, we first describe this algorithm and give the pseudo-code representation of the instance selection approach used by our MILPIS algorithm. Then we analyze why the MILPIS algorithm satisfies the above two constraints.

To describe the MILPIS algorithm, we need to introduce some notation. Let B represent all training bags and m represent the size of B. m^+ and m^- ($m^+ + m^- = m$) are the numbers of positive and negative training bags, respectively. We denote by B_i the i-th bag in B and by B_{ij} the j-th instance in that bag. The bag B_i is composed of n_i instances B_{ij}, $j = 1, 2, \ldots, n_i$. Without ambiguity, B_{ij} also stands for the feature vector of an instance depending on the context.

The MILPIS algorithm starts with selecting from every training bag a pair of instances with the highest similarity level. For this purpose, we need to evaluate a similarity measure between two instances within a bag. Here we adopt the Euclidean distance to determine the similarity between all pairwise instances B_{ij} and B_{ik} in a bag B_i, i.e., $\|B_{ij} - B_{ik}\|$, $i \in \{1, 2, \dots, m\}$, $j, k \in \{1, 2, \dots, n_i\}$ and $j \neq k$. Note that the smaller the distance between two instances, the higher the similarity level between them.

With the definition of the similarity between instances, we can now describe the details of our MILPIS algorithm. We first compute the Euclidean distances between all pairs of instances in every training bag, and select two closest instances from the bag as instance prototypes. Then we use all the instance prototypes to form the embedding space and map every training bag into this new feature space. Thus every training bag is represented by a single bag-level feature vector. Finally, we train a standard SVM with a Gaussian kernel using all the new bag-level features for training bags. The pseudo-code for the instance selection procedure of MILPIS is summarized in Algorithm 1.

Algorithm 1. Instance selection for MILPIS.

```
Input: Set of training bags $B$
Output: Set of instance prototypes $T$
$T = \{\}$
for $i = 1$ to $m$ do
    $d\_min = \infty$
    for $j = 1$ to $n_i - 1$ do
        for $k = j + 1$ to $n_i$ do
            $d = \|B_{ij} - B_{ik}\|$
            if $d < d\_min$ then
                $d\_min = d$
                $t_1 = B_{ij}$
                $t_2 = B_{ik}$
            end if
        end for
    end for
    $T \Leftarrow T \cup \{t_1, t_2\}$
end for
```

Algorithm 1 will output a set of instance prototypes $T = \{t_1, t_2, \dots, t_{2m}\}$, where m is the number of all training bags. Since we select two instances (most similar ones) from every training bag, the number of instance prototypes is the double of that of training bags. All these instance prototypes are then used to define the bag-level feature mapping mentioned above. Formally, given the set $T = \{t_1, t_2, \dots, t_{2m}\}$, the bag-level feature mapping between a bag and all the instance prototypes is defined as

$$D(B_i) = [\mathrm{H}(B_i, t_1), \mathrm{H}(B_i, t_2), \dots, \mathrm{H}(B_i, t_{2m})]^{\mathrm{T}}, \quad (1)$$

where $\mathrm{H}(B_i, t_k) = \min_{B_{ij} \in B_i} \|B_{ij} - t_k\|$ is the minimal Hausdorff distance between a bag and an instance (Wang and Zucker, 2000), and t_k is the k-th item in T. Equation (1) actually defines the global bag-level features for a bag, whose number is equal to the size of the set of instance prototypes T (i.e., $2m$) since every instance prototype determines one dimension of the bag-level features. Thus, given the training set $B = \{B_1, B_2, \dots, B_m\}$ and the set of instance prototypes $T = \{t_1, t_2, \dots, t_{2m}\}$, applying the mapping (1) yields the matrix representation for all the training bags as

$$\begin{aligned} &[D_1, D_2, \dots, D_m] \\ &= [D(B_1), D(B_2), \dots, D(B_m)] \\ &= \begin{bmatrix} \mathrm{H}(B_1, t_1) & \dots & \mathrm{H}(B_m, t_1) \\ \mathrm{H}(B_1, t_2) & \dots & \mathrm{H}(B_m, t_2) \\ \vdots & \ddots & \vdots \\ \mathrm{H}(B_1, t_{2m}) & \dots & \mathrm{H}(B_m, t_{2m}) \end{bmatrix}, \end{aligned} \quad (2)$$

where every column represents the bag-level features for a bag. This kind of distance-based bag-level feature mapping has been actually used by DD-SVM and MILD, although the former employs weighted bag-level features. With all the bag-level features for training bags, the MILPIS algorithm then trains a standard SVM with a Gaussian kernel, and thus the MIL problem has been converted to the standard supervised learning problem.

Note that apart from positive instance prototypes (selected from the positive training bags) the MILPIS algorithm has also selected negative instance prototypes from the negative training bags. Our empirical study shows that including the negative instance prototypes has improved the classification accuracy by an average amount of 2.8% for the drug activity prediction task. Actually, it is not the first time that an ISMIL algorithm includes the negative instance prototypes in the set of instance prototypes. This kind of strategy has been put into practice by several ISMIL algorithms, such as DD-SVM, MILES and MILIS.

Now we analyze why the MILPIS algorithm satisfies the two constraints given in Section 3.1. Algorithm 1 shows that MILPIS focuses on the structure within a bag. In other words, MILPIS puts emphasis on the relation of instances within a bag rather than that of instances between different bags. Specifically, MILPIS establishes the framework of instance selection on the basis of similarity between paired instances within a bag, while other ISMIL algorithms are often based on that between different instances from different bags. For example, DD-SVM needs to compute the distances between almost all pairs of instances from all training bags for a single instance prototype and MILD has to compute the distances from a candidate instance prototype to all instances in all training bags in order to acquire the discriminative ability of this candidate. Therefore, compared with other ISMIL algorithms, our MILPIS algorithm can accomplish the searching process for instance prototypes more quickly. A detailed analysis of computational complexities for different ISMIL algorithms will be presented in Section 3.3. Moreover, MILPIS tries hard to keep small clusters in the feature space. In general, there exists one or more clusters of positive instances in the feature space, which may be chosen from many small clusters. Unlike the clustering or quantization procedure, which does not consider the bag-level structure and may discard small clusters, MILPIS assumes that there may exist small clusters within a bag, which are near to the target concept regions or clusters. Furthermore, MILPIS considers that a small cluster within a bag is likely to be composed of the most similar instances in this bag.

3.3. Computational complexity. The ISMIL algorithm can be roughly divided into three phases, including instance selection, feature mapping and classifier learning. Meanwhile, the instance selection determines the further feature mapping and classifier learning. Thus, we focus on the analysis of the computational efficiency of the instance selection approaches used by various ISMIL algorithms, including our MILPIS algorithm, DD-SVM, MILES, MILD as well as MILIS. Note that the names of these algorithms also represent the corresponding instance selection approaches in the following discussions. To simplify the deliberations, we assume that every training bag contains n instances on the average.

DD-SVM starts from every instance in all positive training bags to search for an instance prototype, so the total number of iterations is m^+n. In each iteration, DD-SVM uses the quasi-Newton method to search for a candidate instance prototype, and thus it needs to compute the distances between almost all pairs of training instances, so the amount of computation produced in each iteration is approximately $(mn)^2$. Therefore, the computational complexity of DD-SVM is equal to $O((m^+n)(mn)^2)$, i.e., $O(m^2m^+n^3)$.

Since MILES initially regards all training instances as valid instance prototypes, it has to compute the distances between all pairs of training instances for the further instance pruning step. Hence, the computational complexity of MILES is equal to $O((mn)^2)$, that is, $O(m^2n^2)$. Note that this computational complexity does not consider the implicit instance pruning process via 1-norm SVM optimization.

In order to assess the discriminative ability of every instance in all positive training bags, MILD has to compute the distances from this instance to all training instances, and thus the computational complexity of MILD is equal to $O((m^+n)(mn))$, i.e., $O(mm^+n^2)$.

To achieve the density estimation for every training instance in the initial instance selection process, MILIS needs to search for its k-nearest negative instances and then evaluates the probability of it being generated from the negative population. The amount of computation consumed on searching for its k-nearest negative instances is m^-n, while that consumed on evaluating the probability is km^-n since for this purpose MILIS has to compute the distances from every instance inside its k-nearest negative instances to all negative training instances. Thus, the total amount of computation is $(k+1)m^-n$ for every training instance. Since k is usually less than m^-, the computational complexity of MILIS equals to $O((mn)(m^-n))$, that is, $O(mm^-n^2)$. This complexity is only related to the initial instance selection process, and the very time-consuming iterative optimization framework for instance updating and classifier learning is not considered. The computational complexities of all the instance selection approaches discussed above are summarized in Table 1, together with the complexity of MILPIS introduced below.

Table 1. Computational complexities for the instance selection approaches used by various ISMIL algorithms.

Algorithm	Complexity
MILPIS	$O(mn^2)$
DD-SVM (Chen and Wang, 2004)	$O(m^2m^+n^3)$
MILES (Chen *et al.*, 2006)	$O(m^2n^2)$
MILD (Li and Yeung, 2010)	$O(mm^+n^2)$
MILIS (Fu *et al.*, 2011)	$O(mm^-n^2)$

Table 1 shows that MILD is the most efficient one among all the above algorithms. Thus, we only need to compare MILPIS with MILD in terms of computational efficiency. From Algorithm 1, we can easily see that the total number of the outermost iterations is equal to that of training bags m. In each iteration, MILPIS needs to compute the distances between all pairs of instances in a bag. Hence, the computational complexity of MILPIS is $O(mn^2)$, which is less than the complexity of MILD ($O(mm^+n^2)$). The analysis of computational efficiency for various ISMIL algorithms was validated by experiments on the MUSK and ANIMAL data sets, the details of which can be seen in Section 4. The higher efficiency of our MILPIS algorithm may be very promising, since we are often faced with large-scale data sets in real-world applications.

4. Experiments

4.1. Drug activity prediction. The MUSK data sets, MUSK1 and MUSK2, are standard benchmarks for MIL, which are publicly available from the UCI Machine Learning Repository (Blake and Merz, 1998). These data sets consist of descriptions of molecules and the task is to predict whether a given molecule is active or inactive. Each molecule is viewed as a bag whose instances are the different low-energy conformations of the molecule.

Surface properties of a conformation are extracted as its feature vector that has 166 dimensions. If one of the conformations of a molecule binds well to the target protein, the molecule is said to be active, and otherwise it is inactive. MUSK1 contains 47 positive bags and 45 negative bags, with an average of 5.17 instances per bag. MUSK2 contains 39 positive bags and 63 negative bags, with 64.69 instances per bag on average. MUSK2 shares 72 molecules with MUSK1, but contains more conformations for those shared molecules.

We used ten random runs of tenfold cross-validation to test our MILPIS algorithm. LIBSVM (Chang and Lin, 2011) was used to train all the SVMs for MILPIS, and thus the regularization parameter C and the Gaussian kernel parameter γ need to be specified. In our experiments, both of them were chosen from $\{2^{-10}, 2^{-8}, \dots, 2^{10}\}$, and a pair of values giving the minimum twofold cross-validation error on the training examples (from nine of ten folds) were selected to set the two parameters. As for other ISMIL algorithms, including DD-SVM, MILES, MILD and MILIS, we used the same setup to determine the corresponding parameters.[1] All the experiments were performed on a 3.1 GHz PC with four cores.

4.1.1. Classification results. The prediction accuracy in ten runs varies from 84.8% to 88.2% for MUSK1, and from 84.3% to 91.4% for MUSK2. Table 2 thus reports the mean and the 95% confidence interval of the results of ten runs of tenfold cross-validation for MILPIS. We also listed some other results on the same data sets for comparison. Table 2 shows that the APR algorithm achieves the best performance on both MUSK1 and MUSK2 data sets in terms of the classification accuracy. However, the APR algorithm chooses the parameters to maximize the performance on the test set, rather than the training set, and thus the superiority of the APR should not be interpreted as a failure. Table 2 also shows that our MILPIS algorithm gives the second best overall prediction accuracy on the MUSK1 and MUSK2 data sets. In particular, MILPIS outperforms all other ISMIL algorithms in terms of the classification accuracy on either MUSK1 or MUSK2, which confirms that our instance selection approach is very effective. This is mainly due to the fact that the MILPIS algorithm can make instances representing small clusters of target concepts remain in the set of instance prototypes.

4.1.2. Computation time. The analysis of computational complexity indicated that our MILPIS algorithm is superior to other ISMIL algorithms with respect to computation time. To validate this conclusion, we give in Table 3 the computation time for various ISMIL algorithms on both MUSK1 and MUSK2 data sets. The computation time for various algorithms on either of the two data sets is the total training and testing time consumed on tenfold cross-validation. Remember that we use ten runs of tenfold cross-validation to evaluate the performance of various ISMIL algorithms on the MUSK data sets. From Table 3, we can easily see that MILPIS performs on a par with MILD in terms of computation time but slightly better than MILD. Meanwhile, they consume much less computation time than other ISMIL algorithms. The speedup of MILPIS over other ISMIL algorithms for the MUSK2 data set is more obvious due to the large number of instances in this data set. The results herein demonstrate that our MILPIS algorithm is more efficient than other ISMIL algorithms, which mainly ascribes to the very fast instance selection scheme used by MILPIS. As mentioned above, MILPIS focuses on the structure within a training bag, specifically, the similarity between instances in every individual training bag. In contrast, other ISMIL algorithms often take into account the relation of instances between different training bags.

4.2. Automatic image annotation.

Automatic image annotation data sets (Andrews *et al.*, 2003) concern identification of three kinds of animals in images: Elephant, Fox and Tiger. For each data set, 100 images containing the target animal are used as positive bags, and 100 images randomly drawn from a set of photos of other animals as negative bags. Each image is represented by a set of segments and each segment is described by a 230-dimensional feature vector characterizing color, texture and shape. The number of instances in a bag ranges from 1 to 13, with an average of 7, 6.6, and 6.1 instances per bag for Elephant, Fox and Tiger, respectively. We used the same experimental setup and parameter selection approach as in Section 4.1 to test the performance of various ISMIL algorithms on the three data sets.

4.2.1. Annotation results. We applied tenfold cross-validation with 10 different random runs to estimate the prediction accuracy of our MILPIS algorithm on the three data sets. The prediction accuracy range of MILPIS over 10 runs is $[82.0\%, 85.5\%]$ for Elephant, $[64.0\%, 69.0\%]$ for Fox and $[79.0\%, 84.0\%]$ for Tiger. Therefore, we report in Table 4 the average and 95% confidence interval of the results of 10 runs of tenfold cross-validation for MILPIS. Table 4 also summarizes the results of other ISMIL algorithms on Elephant, Fox and Tiger for comparison. Table 4 shows that the MILPIS and MILES algorithms are highly comparable with each other

[1] We noticed that several ISMIL algorithms, including DD-SVM, MILES, MILD and MILIS, tuned the SVM parameters on the whole data set. For a fair comparison, we implemented these algorithms and run them based on our experimental setup.

Table 2. Classification accuracies (in %) for various MIL algorithms on the MUSK data sets.

Algorithm	MUSK1	MUSK2	Avg
MILPIS	86.9:[86.2,87.5]	88.5:[87.1,89.8]	87.7
DD-SVM (Chen and Wang, 2004)	85.6:[83.9,87.2]	87.3:[86.3,88.2]	86.5
MILES (Chen *et al.*, 2006)	86.6:[84.9,88.4]	88.3:[86.8,89.9]	87.5
MILD (Li and Yeung, 2010)	85.0:[82.8,87.1]	85.0:[83.6,86.5]	85.0
MILIS (Fu *et al.*, 2011)	86.4:[84.6,88.2]	88.3:[87.2,89.5]	87.4
APR (Dietterich *et al.*, 1997)	92.4	89.2	90.8
DD (Maron and Lozano-Pérez, 1998)	88.9	82.5	85.7
EM-DD (Zhang and Goldman, 2002)	84.8	84.9	84.9
MI-SVM (Andrews *et al.*, 2003)	77.9	84.3	81.1
mi-SVM (Andrews *et al.*, 2003)	87.4	83.6	85.5

Table 3. Computation time (in minutes) for various ISMIL algorithms on the MUSK data sets.

Algorithm	MUSK1	MUSK2
MILPIS	0.04	0.28
DD-SVM (Chen and Wang, 2004)	8.74	122.57
MILES (Chen *et al.*, 2006)	0.13	4.42
MILD (Li and Yeung, 2010)	0.04	0.46
MILIS (Fu *et al.*, 2011)	8.17	3091.39

in terms of classification accuracy, and superior to other ISMIL algorithms (i.e., DD-SVM, MILD and MILIS), which once again demonstrates the effectiveness of the instance selection method used by our MILPIS algorithm. As stated by Chen *et al.* (2006), the MILES algorithm essentially uses the linear SVM feature selection method, which can take into account the correlations between features (instances). However, this is not the case for DD-SVM, MILD and MILIS. For example, DD-SVM focuses on the individual points (instances) corresponding to local maxima of DD, not several instances that together determine a candidate target concept region. Like MILES, our MILPIS algorithm takes advantage of the relation between instances, specifically, the similarity between pairwise instances in every bag. This may be why MILES and MILPIS perform better in the experiments when compared with other algorithms.

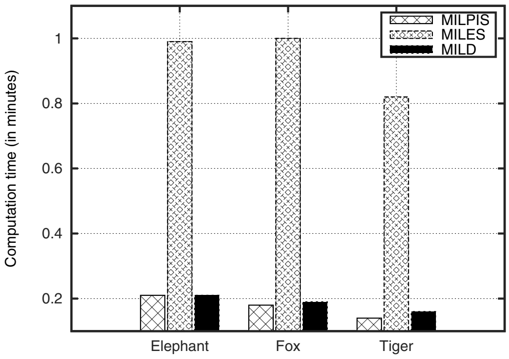

Fig. 1. Computation time for various ISMIL algorithms on the Elephant, Fox and Tiger data sets.

4.2.2. Computation time. Figure 1 illustrates the computation time consumed by various ISMIL algorithms for the automatic image annotation task. Since the computation time for DD-SVM on each of the three data sets is nearly one hour and that for MILIS is more than two hours, we do not show them in Fig. 1. From the figure, we can see that MILPIS and MILD are highly competitive with each other in terms of computation time and much better than MILES for all the three data sets. Although the MILES algorithm performs on a par with our MILPIS algorithm in terms of classification accuracy (see Table 4), MILPIS is superior to MILES with respect to both effectiveness and efficiency, since it needs much less computation time for the whole learning process. The reason resides in the fact that MILPIS considers the correlations between instances from the perspective of local scope, while MILES considers this problem from the global point of view. Consequently, MILPIS can accomplish the instance selection process very quickly, whereas MILPIS needs a great amount of time to learn a linear SVM for feature selection.

4.3. Region-based image categorization. The COREL data sets have been widely used for region-based image categorization. The data sets contain 20 thematically diverse image categories with 100 images of size 384×256 or 256×384 in each category.

Table 4. Classification accuracies (in %) for various ISMIL algorithms on the Elephant, Fox and Tiger data sets.

Algorithm	Elephant	Fox	Tiger	Avg
MILPIS	83.8:[83.1,84.5]	67.3:[66.3,68.2]	81.9:[81.0,82.8]	77.7
DD-SVM (Chen and Wang, 2004)	80.8:[80.1,81.4]	61.2:[59.7,62.7]	76.7:[76.0,77.4]	72.9
MILES (Chen *et al.*, 2006)	83.2:[82.6,83.8]	66.9:[65.8,68.0]	82.0:[81.2,82.8]	77.4
MILD (Li and Yeung, 2010)	77.3:[76.6,78.1]	60.1:[59.4,60.9]	76.5:[75.7,77.3]	71.3
MILIS (Fu *et al.*, 2011)	83.8:[82.5,85.1]	61.2:[58.8,63.6]	82.5:[81.1,84.0]	75.8

Each image is segmented into several local regions and features are extracted from each region. The data sets and extracted features are publicly available at www.cs.olemiss.edu/~ychen/ddsvm.html. Details of segmentation and feature extraction are beyond the scope of this paper and interested readers are referred to the work of Chen and Wang (2004) for further information.

For the COREL data sets, we conducted two tests for the 10-category and 20-category image categorizations. The first 10 categories in the COREL data sets were used in the first test while all 20 categories were used in the second one. For each category, we randomly selected half of images as training bags and the remaining half as test bags. We repeated each experiment for five different random partitions and reported the average of the results obtained over five different test tests. Since this is a multi-class classification problem, we simply applied the one-against-the-rest approach to train 10/20 binary SVMs. A test bag is assigned to the category with the largest decision value given by a specific SVM. Both the regularization parameter C and the Gaussian kernel parameter γ were chosen from $\{2^{-10}, 2^{-8}, \ldots, 2^{10}\}$, and a pair of values giving the minimum twofold cross-validation error on the training examples were selected to set the two parameters.

4.3.1. Categorization results. We compared the prediction accuracy of MILPIS with that of DD-SVM, MILES, MILD and MILIS. The average classification accuracies over five different random test sets and the corresponding 95% confidence intervals are provided in Table 5. The table shows that the overall prediction accuracy of MILPIS on both of COREL data sets is highly comparable with that of MILES (only 0.1% of the difference) and slightly higher than that of MILIS. As for DD-SVM and MILD, Table 5 shows that they are obviously inferior to MILPIS and the other two algorithms. The better performance of MILES and MILPIS with respect to prediction accuracy once again demonstrates that it is very important for an ISMIL algorithm to take into account the dependencies between instances for instance selection. Meanwhile, the similarity scheme used by MILPIS is an appropriate measure for the correlations between instances in that it uses paired instances to capture small concept regions in the feature space.

4.3.2. Sensitivity to labeling noise. Then we tested the sensitivity of MILPIS to labeling noise based on the COREL data sets. We used the same setting as that in MILES and MILD to perform this experiment. The tests were conducted using binary classification, and labeling noise is defined as the probability that an image is mislabeled. Category 3 (Buses) and Category 4 (Dinosaurs) were used in this experiment, since they can be almost perfectly separated by all the ISMIL algorithms, which makes them a good data set for testing noise sensitivity.[2] We randomly selected half of images from the data set to form a training set and the remaining half to form a test set. To introduce labeling noise for training images, we first randomly picked $d\%$ of positive images and $d\%$ of negative images from the training set, and then negated the labels of these images, i.e., labeling positive (negative) images as negative (positive) images. The training set thus contains $d\%$ of mislabeled images. Both the regularization parameter C and the Gaussian kernel parameter γ were still chosen from $\{2^{-10}, 2^{-8}, \ldots, 2^{10}\}$, and a pair of values giving the minimum twofold cross-validation error on the training set were selected to set them. Training and testing were repeated for five different random partitions and the average classification accuracy was computed. We compared various ISMIL algorithms at different levels of labeling noise ($d = 0 - 30$ with step size 2).

The average classification accuracies over five randomly generated test sets are shown in Fig. 2. Here we can see that MILPIS and MILD are better than other ISMIL algorithms when there exists noise in labels, and even so MILPIS still outperforms MILD slightly with respect to the overall prediction accuracy at all levels of labeling noise. Specifically, the classification accuracies of all the algorithms are almost 100% when there is no labeling noise ($d = 0$). When the noise level increases, the better prediction accuracy of MILPIS, MILD and MILES can be kept until $d = 24$ while that of MILIS deteriorates to a greater extent. Meanwhile, Fig. 2 also shows that

[2] Unlike with MILES and MILD, we did not use Category 2 (Historical buildings) and Category 7 (Horses) in this experiment, since they cannot be almost perfectly separated by all the ISMIL algorithms.

Table 5. Classification accuracies (in %) for various ISMIL algorithms on the COREL data sets.

Algorithm	COREL_{10}	COREL_{20}	Avg
MILPIS	82.2:[80.8,83.5]	69.5:[67.6,71.4]	75.9
DD-SVM (Chen and Wang, 2004)	73.0:[71.8,74.1]	54.3:[51.0,57.7]	63.7
MILES (Chen *et al.*, 2006)	82.0:[81.2,82.9]	69.9:[68.3,71.6]	76.0
MILD (Li and Yeung, 2010)	80.1:[77.9,82.3]	66.8:[65.5,68.1]	73.5
MILIS (Fu *et al.*, 2011)	81.2:[79.3,83.2]	69.7:[67.2,72.1]	75.5

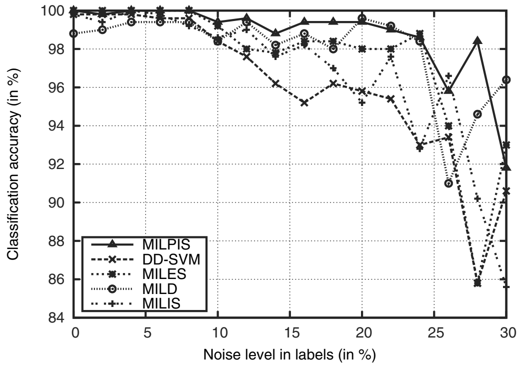

Fig. 2. Sensitivity of various ISMIL algorithms to labeling noise.

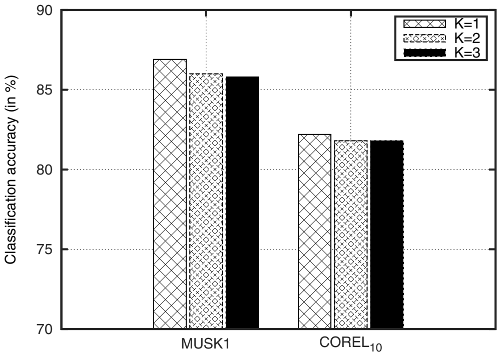

Fig. 3. Classification accuracies for MILPIS on MUSK1 and COREL_{10} with different numbers (K) of pairs of instances selected from every training bag.

DD-SVM is the most sensitive one among all the ISMIL algorithms, which has also been demonstrated by MILES and MILD. The better performance of MILPIS implies that its instance selection method is very robust to labeling noise, which is also due to the perspective adopted by MILPIS in instance selection. Since MILPIS focuses on the inner structure of a bag and does not consider the label of the bag, it can still choose the representative instance prototypes from the bag even though the bag is mislabeled. As a result, MILPIS is less sensitive to labeling noise.

4.4. Evaluation of similarity-based instance selection. Algorithm 1 shows that the MILPIS algorithm selects from every training bag a pair of the most similar instances as instance prototypes. One may wonder if the prediction accuracy of MILPIS will become higher when selecting several pairs of instances from every training bag. To address this issue, we performed the experiments on the MUSK1 and COREL_{10} data sets using a simple instance selection approach which extends Algorithm 1 to choose from every training bag K pairs of the most similar instances. The experimental setup and parameter selection method for MUSK1 and COREL_{10} were the same as those described in Sections 4.1 and 4.3, respectively. Figure 3 shows the corresponding classification results of MILPIS on both MUSK1 and COREL_{10} data sets.

In Fig. 3 we can see that, when $K = 1$ (corresponding to Algorithm 1), the MILPIS algorithm achieves the highest performance on both MUSK1 and COREL_{10} data sets. On the other hand, the performance deteriorates to a smaller extent when adding additional pairs of instances ($K = 2, 3$) into the set of instance prototypes. This is mainly due to the fact that some instances irrelevant to the target concepts were selected from the training bags when $K = 2, 3$. Moreover, adding more instance prototypes will make the further feature mapping and classifier learning become slower. Based on these results, we can assert that it is unnecessary to choose from every training bag several pairs of the most similar instances as instance prototypes, and Algorithm 1 is sufficient for common MIL tasks.

5. Conclusions and future work

Based on similarity between instances within a bag, we proposed a novel instance selection method for an ISMIL algorithm called MILPIS. The MILPIS algorithm tries to capture small clusters of target concepts in the feature space by considering the correlations between instances; specifically, the similarity between pairwise instances in a bag. Accordingly, MILPIS achieves results highly comparable with those of other ISMIL algorithms on three MIL tasks in the experiments, which confirms that our instance selection method is very effective. Since MILPIS focuses on the local scope of every training bag for instance selection, it showed the highest efficiency in the experiments compared with other similar algorithms. In addition, the testing of noise sensitivity demonstrates that our MILPIS algorithm is very robust to labeling noise. This mainly ascribes to the fact that MILPIS takes into account only the inner structure of a bag for instance selection and thus will not be affected by the label of the bag.

Following the descriptions of the MILPIS algorithm, we know that it will select all the instances in a bag as instance prototypes if the number of instances in this bag is equal to 2. If we adopt some strategy to choose one from the two instances, the generalization ability and efficiency of the algorithm may be improved to some extent. Moreover, as we know, every individual ISMIL algorithm performs instance selection from a different point of view. Thus, it may be interesting to investigate how to integrate different instance selection methods together, e.g., using ensemble feature selection techniques.

Acknowledgment

This research has been supported by the National Natural Science Foundation of China under the grants no. 61173087 and 61370162. The authors would like to thank the anonymous reviewers for their valuable suggestions which led to improvement of this paper.

References

Ali, S. and Shah, M. (2010). Human action recognition in videos using kinematic features and multiple instance learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**(2): 288–303.

Andrews, S. and Hofmann, T. (2004). Multiple instance learning via disjunctive programming boosting, *Proceedings of Advances in Neural Information Processing Systems 16, Vancouver and Whistler, BC, Canada*, pp. 65–72.

Andrews, S., Tsochantaridis, I. and Hofmann, T. (2003). Support vector machines for multiple-instance learning, *Proceedings of Advances in Neural Information Processing Systems 15, Vancouver, BC, Canada*, pp. 561–568.

Auer, P. and Ortner, R. (2004). A boosting approach to multiple instance learning, *Proceedings of the 15th European Conference on Machine Learning, Pisa, Italy*, pp. 63–74.

Babenko, B., Yang, M.-H. and Belongie, S. (2009). Visual tracking with online multiple instance learning, *Proceedings of the 22nd Conference on Computer Vision and Pattern Recognition, Miami, FL, USA*, pp. 983–990.

Babenko, B., Verma, N., Dollár, P. and Belongie, S. (2011a). Multiple instance learning with manifold bags, *Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA*, pp. 81–88.

Babenko, B., Yang, M.-H. and Belongie, S. (2011b). Robust object tracking with online multiple instance learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**(8): 1619–1632.

Bergeron, C., Moore, G., Zaretzki, J., Breneman, C.M. and Bennett, K. P. (2012). Fast bundle algorithm for multiple-instance learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(6): 1068–1079.

Blake, C.L. and Merz, C.J. (1998). UCI repository of machine learning databases, `http://archive.ics.uci.edu/ml/`.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology* **2**(27): 1–27, `www.csie.ntu.edu.tw/~cjlin/libsvm/`.

Chen, Y., Bi, J. and Wang, J. Z. (2006). MILES: Multiple-instance learning via embedded instance selection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(12): 1931–1947.

Chen, Y. and Wang, J. Z. (2004). Image categorization by learning and reasoning with regions, *Journal of Machine Learning Research* **5**: 913–939.

Czarnowski, I. and Jędrzejowicz, P. (2011). Application of agent-based simulated annealing and tabu search procedures to solving the data reduction problem, *International Journal of Applied Mathematics and Computer Science* **21**(1): 57–68, DOI: 10.2478/v10006-011-0004-3.

Dietterich, T. G., Lathrop, R. H. and Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles, *Artificial Intelligence* **89**(1–2): 31–71.

Dollár, P., Babenko, B., Belongie, S., Perona, P. and Tu, Z. (2008). Multiple component learning for object detection, *Proceedings of the 10th European Conference on Computer Vision, Marseille, France*, pp. 211–224.

Fu, Z., Robles-Kelly, A. and Zhou, J. (2011). MILIS: Multiple instance learning with instance selection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**(5): 958–977.

Fung, G., Dundar, M., Krishnapuram, B. and Rao, R.B. (2007). Multiple instance learning for computer aided diagnosis, *Proceedings of Advances in Neural Information Processing Systems 19, Vancouver, BC, Canada*, pp. 425–432.

Gärtner, T., Flach, P.A., Kowalczyk, A. and Smola, A.J. (2002). Multi-instance kernels, *Proceedings of the 19th International Conference on Machine Learning, Sydney, NSW, Australia*, pp. 179–186.

Li, F. and Sminchisescu, C. (2010). Convex multiple-instance learning by estimating likelihood ratio, *Proceedings of Advances in Neural Information Processing Systems 23, Vancouver, BC, Canada*, pp. 1360–1368.

Li, M., Kwok, J.T. and Lu, B.-L. (2010). Online multiple instance learning with no regret, *Proceedings of the 23rd Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA*, pp. 1395–1401.

Li, W.-J. and Yeung, D.-Y. (2010). MILD: Multiple-instance learning via disambiguation, *IEEE Transactions on Knowledge and Data Engineering* **22**(1): 76–89.

Li, Y., Tax, D. M.J., Duin, R.P.W. and Loog, M. (2013). Multiple-instance learning as a classifier combining problem, *Pattern Recognition* **46**(3): 865–874.

Maron, O. and Lozano-Pérez, T. (1998). A framework for multiple-instance learning, *Proceedings of Advances in Neural Information Processing Systems 10, Denver, CO, USA*, pp. 570–576.

Maron, O. and Ratan, A. L. (1998). Multiple-instance learning for natural scene classification, *Proceedings of the 15th International Conference on Machine Learning, Madison, WI, USA*, pp. 341–349.

Nguyen, D.T., Nguyen, C.D., Hargraves, R., Kurgan, L.A. and Cios, K.J. (2013). mi-DS: Multiple-instance learning algorithm, *IEEE Transactions on Cybernetics* **43**(1): 143–154.

Rahmani, R., Goldman, S.A., Zhang, H., Cholleti, S.R. and Fritts, J.E. (2008). Localized content-based image retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**(11): 1902–1912.

Ramon, J. and De Raedt, L. (2000). Multi instance neural networks, *Proceedings of the 17th International Conference on Machine Learning/Workshop on Attribute-Value and Relational Learning, Stanford, CA, USA*.

Raykar, V.C., Krishnapuram, B., Bi, J., Dundar, M. and Rao, R.B. (2008). Bayesian multiple instance learning: Automatic feature selection and inductive transfer, *Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland*, pp. 808–815.

Settles, B., Craven, M. and Ray, S. (2008). Multiple-instance active learning, *Proceedings of Advances in Neural Information Processing Systems 20, Vancouver, BC, Canada*, pp. 1289–1296.

Tao, Q., Scott, S.D., Vinodchandran, N.V., Osugi, T.T. and Mueller, B. (2008). Kernels for generalized multiple-instance learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**(12): 2084–2098.

Trawiński, B., Smętek, M., Telec, Z. and Lasota, T. (2012). Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms, *International Journal of Applied Mathematics and Computer Science* **22**(4): 867–881, DOI: 10.2478/v10006-012-0064-z.

Vezhnevets, A. and Buhmann, J. M. (2010). Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning, *Proceedings of the 23rd Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA*, pp. 3249–3256.

Viola, P.A., Platt, J.C. and Zhang, C. (2006). Multiple instance boosting for object detection, *Proceedings of Advances in Neural Information Processing Systems 18, Vancouver, BC, Canada*, pp. 1417–1424.

Wang, J. and Zucker, J.-D. (2000). Solving the multiple-instance problem: A lazy learning approach, *Proceedings of the 17th International Conference on Machine Learning, Stanford, CA, USA*, pp. 1119–1126.

Yang, C. and Lozano-Pérez, T. (2000). Image database retrieval with multiple-instance learning techniques, *Proceedings of the 16th International Conference on Data Engineering, San Diego, CA, USA*, pp. 233–243.

Zha, Z.-J., Hua, X.-S., Mei, T., Wang, J., Qi, G.-J. and Wang, Z. (2008). Joint multi-label multi-instance learning for image classification, *Proceedings of the 21st Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA*, pp. 1–8.

Zhang, M.-L. and Zhou, Z.-H. (2004). Improve multi-instance neural networks through feature selection, *Neural Processing Letters* **19**(1): 1–10.

Zhang, Q. and Goldman, S.A. (2002). EM-DD: An improved multiple-instance learning technique, *Proceedings of Advances in Neural Information Processing Systems 14, Vancouver, BC, Canada*, pp. 1073–1080.

Zhang, Q., Goldman, S.A., Yu, W. and Fritts, J.E. (2002). Content-based image retrieval using multiple-instance learning, *Proceedings of the 19th International Conference on Machine Learning, Sydney, NSW, Australia*, pp. 682–689.

Liming Yuan received the B.Sc. degree in computer science and technology from Harbin Normal University, China, in 2005, and the M.Sc. degree in computer applied technology from Harbin Engineering University, China, in 2009. He is currently working toward his Ph.D. degree at the School of Computer Science and Technology, Harbin Institute of Technology, China. His main research interest concentrates on using example selection, instance selection and feature selection to solve the multiple-instance learning problem.

Jiafeng Liu received his Ph.D. degree from the Harbin Institute of Technology, China, in 1996. He is currently an associate professor at the School of Computer Science and Technology, Harbin Institute of Technology. His research interests cover image and video analysis, optimal character recognition, pattern recognition, machine learning and artificial intelligence. He has published over 40 papers in refereed international journals.

Xianglong Tang received his Ph.D. degree from the Harbin Institute of Technology, China, in 1995. He is currently a professor at the School of Computer Science and Technology and the director of the Research Center of Pattern Recognition, both at the Harbin Institute of Technology. His main research interests are focused on Chinese character recognition, medical imaging and biometrics, computer vision and pattern recognition. He has published over 80 papers in refereed international journals.

Received: 1 July 2013
Revised: 1 December 2013