Multi-Instance Learning from Supervised View

Zhi-Hua Zhou (周志华)

National Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, P.R. China

E-mail: zhouzh@nju.edu.cn

Received March 13, 2006; revised June 29, 2006.

Abstract In multi-instance learning, the training set comprises labeled *bags* that are composed of unlabeled instances, and the task is to predict the labels of unseen bags. This paper studies multi-instance learning from the view of supervised learning. First, by analyzing some representative learning algorithms, this paper shows that multi-instance learners can be derived from supervised learners by shifting their focuses from the discrimination on the instances to the discrimination on the bags. Second, considering that ensemble learning paradigms can effectively enhance supervised learners, this paper proposes to build multi-instance ensembles to solve multi-instance problems. Experiments on a real-world benchmark test show that ensemble learning paradigms can significantly enhance multi-instance learners.

Keywords machine learning, multi-instance learning, supervised learning, ensemble learning, multi-instance ensemble

1 Introduction

During the past years, *learning from examples* has become one of the most flourishing areas in machine learning. According to the *ambiguity* of the labels of training examples, previous research in this area can be roughly categorized into three learning frameworks, i.e., supervised learning, unsupervised learning, and reinforcement learning[1]. Supervised learning attempts to learn a concept for correctly labeling unseen instances, where the training instances are with known labels and therefore the ambiguity is minimum; unsupervised learning attempts to learn the structure of the underlying sources of instances, where the training instances are without known labels and therefore the ambiguity is the maximum; reinforcement learning attempts to learn a mapping from states to actions, where the instances are with no labels but with delayed rewards which can be viewed as delayed labels and therefore the ambiguity is between that of supervised learning and unsupervised learning.

The term *multi-instance learning* was coined by Dietterich *et al.*[2] when they were investigating the problem of drug activity prediction. In multi-instance learning, the training set is composed of many *bags* each contains many instances. A bag is positively labeled if it contains at least one positive instance and negative otherwise. The task is to learn some concept from the training set for correctly labeling unseen bags.

In contrast to supervised learning where all training instances are with known labels, in multi-instance learning the labels of the training instances are unknown; in contrast to unsupervised learning where all training instances are without known labels, in multi-instance learning the labels of the training bags are known; and in contrast to reinforcement learning where the labels of the training instances are delayed, in multi-instance learning there is no any delay. It has been shown that learning algorithms ignoring the characteristics of multi-instance problems, such as popular decision trees and neural networks, cannot work well in this scenario[2].

Since multi-instance problems extensively exist but are unique to these addressed by previous learning frameworks, multi-instance learning was regarded as a new learning framework[1], and has attracted much attention of the machine learning community.

The contribution of this paper lies in two aspects. First, illustrated by the analyses on some representative multi-instance learning algorithms, this paper shows that supervised learning algorithms can be adapted to multi-instance learning as long as their focuses are shifted from the discrimination on the instances to the discrimination on the bags. This insight provides a general solution to the open problem raised by Dietterich *et al.*[2], that is, *how to design multi-instance modifications for popular machine learning algorithms*. Second, considering that ensemble learning paradigms can effectively improve the generalization ability in supervised learning[3] and the first contribution of this paper has revealed that multi-instance learning has close connection with supervised learning, this paper proposes to build *multi-instance ensembles* to solve multi-instance problems. Experiments on a real-world benchmark data set show that current multi-instance learners can be significantly enhanced by ensemble learning paradigms.

The rest of this paper is organized as follows. Section 2 briefly reviews the advances in the area of multi-instance learning. Section 3 shows that multi-instance learners can be derived by shifting the focuses of supervised learners. Section 4 proposes to build multi-instance ensembles and reports on the experimental results. Finally, Section 5 concludes and raises several issues for future work.

2 Multi-Instance Learning

In the middle of 1990s, Dietterich *et al.*[2] investigated

Regular Paper

Supported by the National Natural Science Foundation of China under Grant Nos. 60105004 and 60325207.

the problem of *drug activity prediction.* The goal was to endow learning systems with the ability of predicting that whether a new molecule was qualified to make some drug or not, by analyzing a collection of known molecules.

Most drugs are small molecules working by binding to larger protein molecules such as enzymes and cell-surface receptors. For a molecule qualified to make a drug, at least one of its low-energy shapes can tightly bind to the target area; while for a molecule unqualified to make a drug, none of its low-energy shapes can tightly bind to the target area. The main difficulty of drug activity prediction lies in that each molecule could have many alternative low-energy shapes, but currently biochemists only know that whether a molecule is qualified to make a drug or not, instead of knowing that which of its alternative low-energy shapes responses for the qualification.

An intuitive solution is to exploit traditional supervised learning algorithms by regarding all the low-energy shapes of the "good" molecules as positive training instances, while regarding all the low-energy shapes of the "bad" molecules as negative training instances. However, as shown by Dietterich *et al.*[2], such a method can hardly work due to the high false positive noise, which is caused by the fact that a "good" molecule may have hundreds of low-energy shapes but maybe only one of them is really a "good" shape.

In order to solve this problem, Dietterich *et al.*[2] regarded each molecule as a *bag*, and the alternative low-energy shapes of the molecule as the instances in the bag, thereby formulated multi-instance learning.

Formally, let $\mathcal{X}$ denote the instance space and $\mathcal{Y}$ the set of class labels. The task of multi-instance learning is to learn a function $f : 2^{\mathcal{X}} \to \{-1, +1\}$ from a given data set $\{(X_1, y_1), (X_2, y_2), \ldots, (X_m, y_m)\}$, where $X_i \subseteq \mathcal{X}$ is a set of instances $\{\boldsymbol{x}_1^{(i)}, \boldsymbol{x}_2^{(i)}, \ldots, \boldsymbol{x}_{n_i}^{(i)}\}$, $\boldsymbol{x}_j^{(i)} \in \mathcal{X}$ $(j = 1, \ldots, n_i)$, and $y_i \in \{-1, +1\}$ is the known label of X_i. In contrast, the task of traditional supervised learning is to learn a function $f : \mathcal{X} \to \mathcal{Y}$ from a given data set $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$, where $\boldsymbol{x}_i \in \mathcal{X}$ is an instance and $y_i \in \mathcal{Y}$ is the known label of $\boldsymbol{x}_i$.

Then, Dietterich *et al.*[2] proposed three Axis-Parallel Rectangle (abbreviated as Apr) algorithms, which attempt to search for appropriate axis-parallel rectangles constructed by the conjunction of the features, to tackle the problem. Among these algorithms the Iterated-Discrim Apr algorithm has achieved the best performance on the *Musk* data, a concrete test data for the drug activity prediction task which becomes a popular real-world benchmark test for multi-instance learning algorithms since then.

Note that multi-instance problems do not emerge suddenly from drug activity prediction. Actually, they extensively exist in real-world applications[4,5], but unfortunately the uniqueness of these problems has not been particularly distinguished until Dietterich *et al.*'s work[2].

Long and Tan[6] initiated the investigation of the Pac-learnability of Apr under the multi-instance learning framework. They showed that if the instances in the bags are independently drawn from product distribution, then the Apr is Pac-learnable. Auer *et al.*[7] showed that if the instances in the bags are not independent then Apr learning under the multi-instance learning framework is NP-hard. Moreover, they presented a theoretical algorithm that does not require product distribution but with smaller sample complexity than that of Long and Tan's algorithm, which was transformed to a practical algorithm named Multinst later[8]. Blum and Kalai[9] described a reduction from Pac-learning under the multi-instance learning framework to Pac-learning with one-sided random classification noise. They also presented a theoretical algorithm with smaller sample complexity than that of Auer *et al.*'s algorithm[7].

It is noteworthy that almost all of these theoretical analyses were made under strong assumptions, such as *the number of instances in the bags is a constant* and *all the instances are independent.* These assumptions are unfortunately hard to be met in real-world problems. For example, in drug activity prediction it is not the fact that different molecules have the same number of alternative low-energy shapes, and it is not reasonable to assume that the different alternative low-energy shapes of a molecule are independent.

Maron and Lozano-Pérez[10] proposed a practical multi-instance learning algorithm, Diverse Density, which has been applied to diverse tasks including natural scene classification[11], stock selection[10], subgoal discovery[12], content-based image retrieval[13,14], image categorization[15], etc. Through combining Diverse Density with EM[16], Zhang and Goldman[17] proposed the EM-DD algorithm, which has been applied to content-based image retrieval[18]. Wang and Zucker[19] proposed the multi-instance k-nearest neighbor algorithms Citation-$k$nn and Bayesian-$k$nn. Later, Zhou *et al.*[20] developed Fretcit-$k$nn, a variant of Citation-$k$nn, and applied it to web mining. Zhou and Zhang[21] proposed the multi-instance neural network BP-MIP, and by incorporating feature scaling and feature reduction mechanisms into BP-MIP, they developed BP-MIP-DD and BP-MIP-PCA[22]. There are also many other practical multi-instance learning algorithms, such as Ruffo's multi-instance decision tree Relic[23], Chevaleyre and Zucker's multi-instance decision tree ID3-MI and multi-instance rule inducer Ripper-mi[24], Gärtner *et al.*'s multi-instance kernels[25], Andrews *et al.*'s multi-instance support vector machines[26], Zhou and Zhang's multi-instance ensembles[27], Xu and Frank's multi-instance logistic regression algorithm MiLogisticRegression and multi-instance ensemble algorithm MiBoosting[28], Zhang and Zhou's multi-instance neural network RBF-MIP[29], etc. Some of these algorithms will be analyzed in Section 3.

In the early years of the research of multi-instance learning, most work was on multi-instance classification with discrete-valued outputs. Later, multi-instance regression with real-valued outputs caught the attention of many researchers[30−33].

Recently, Weidmann *et al.*[34] indicated that by employing different assumptions of how the instances' classifications determine their bag's label, different kinds of multi-instance problems can be defined. Based on this recognition, they defined three kinds of generalized multi-instance problems, i.e., *presence-based MI*, *threshold-based MI*, and *count-based MI*, and proposed the TLC algorithm to tackle these problems. Scott *et al.*[35] defined another kind of generalized multi-instance problem which is close to threshold-based MI, and proposed the GMIL-1 algorithm to solve this problem, which was then reformulated as a kernel algorithm[36], reducing the time complexity from exponential to polynomial. Later, this kernel was further generalized along the line of count-based MI[37].

It is worth noting that multi-instance learning has also attracted the attention of the inductive logic programming (abbreviated as ILP) community. It has been suggested that multi-instance problems could be regarded as a bias on ILP, and the multi-instance paradigm could be the key between the propositional and relational representations, being more expressive than the former, and much easier to learn than the latter[38]. Recently, Alphonse and Matwin[39] successfully employed multi-instance learning to help relational learning. At first, the original relational learning problem is approximated by a multi-instance problem. Then, the resulting data is passed to feature selection techniques adapted from propositional representations. Finally, the filtered data is transformed back to relational representation for a relational learner to learn. In this way, the expressive power of relational representation and the ease of feature selection on propositional representation are gracefully combined. It is also worth noting that although multi-instance learning was proposed initially based on propositional representation, a recent modification developed by McGovern and Jensen[40] allows multi-instance techniques to be used on relational representation. These works confirm that multi-instance learning can really act as a bridge between propositional and relational learning.

3 Adapt Supervised Learners to Multi-Instance Learning

When proposing the notion of multi-instance learning, Dietterich *et al.*[2] raised an open problem, i.e., *how to design multi-instance modifications for popular machine learning algorithms*. This open problem greatly pushes the development of this area. Actually, multi-instance versions of many machine learning algorithms have been proposed during the past years. However, these algorithms were developed in a one by one manner, since there was no general rule indicating how to do such a modification.

Generally speaking, the focus of a supervised learner is to discriminate the instances, which is feasible since the labels of all the training instances are known in supervised scenario. But in multi-instance learning it is very difficult, if not infeasible, to discriminate training instances because the labels of the training instances are unknown. If the label of a bag is simply used as the label of its instances, then the learning task will become a difficult one although every training instance holds a label now, because the false positive noise may be extremely high, as indicated by Dietterich *et al.*[2] Therefore, whether it is possible to discriminate the training instances or not is the principal difference between supervised learning and multi-instance learning.

Illustrated by the analyses on some representative algorithms including Diverse Density, Citation-$k$nn, ID3-MI, Ripper-mi, and BP-MIP, this section shows that *supervised learners can be adapted to multi-instance learning as long as their focuses are shifted from the discrimination on the instances to the discrimination on the bags*, which provides a general solution to Dietterich *et al.*'s open problem.

3.1 Diverse Density

The Diverse Density algorithm[10] regards each bag as a manifold, which is composed of many instances, i.e., feature vectors. If a new bag is positive then it is believed to intersect all positive feature-manifolds without intersecting any negative feature-manifolds. Intuitively, *diverse density* at a point in the feature space is defined to be a measure of how many different positive bags have instances near that point, and how far the negative instances are from that point. Thus, the task of multi-instance learning is transformed to the search for a point in the feature space with the *maximum diverse density*.

It is evident that the key of the Diverse Density algorithm lies in the formal definition of the *maximum diverse density*, which is the objective to be optimized by the algorithm. Below we show that such a definition can be attained by modifying standard Bayesian classifier according to the rule, i.e., shifting the focus from discriminating the instances to discriminating the bags.

Given data set D and a set of class labels $C = \{c_1, c_2, \ldots, c_t\}$ to be predicted, the posterior probability of the class can be estimated according to the Bayes rule as shown in (1).

$$\Pr(C|D) = \frac{\Pr(D|C)\Pr(C)}{\Pr(D)}. \tag{1}$$

What we want is the class label with the maximum posterior probability, as indicated in (2), where *Obj* denotes the objective.

$$Obj = \mathop{\arg\max}_{1 \leqslant k \leqslant t} \Pr(c_k|D)$$

$$= \arg\max_{1\leqslant k\leqslant t} \frac{\Pr(D|c_k)\Pr(c_k)}{\Pr(D)}. \quad (2)$$

Considering that $\Pr(D)$ is a constant which can be dropped, and $\Pr(c_k)$ can also be dropped if we assume uniform prior, (2) can be simplified to (3).

$$\mathit{Obj} = \arg\max_{1\leqslant k\leqslant t} \Pr(D|c_k). \quad (3)$$

(3) is fine when the goal is to discriminate the instances, but for discriminating the bags it is helpful to consider $D = \{B_1^+,\ldots,B_m^+,B_1^-,\ldots,B_n^-\}$ where B_i^+ denotes the i-th positive bag while B_j^- denotes the j-th negative bag. Then, assuming that the bags are conditionally independent, (3) can be re-written into (4).

$$\begin{aligned}\mathit{Obj} &= \arg\max_{1\leqslant k\leqslant t} \Pr(\{B_1^+,\ldots,B_m^+,B_1^-,\ldots,B_n^-\}|c_k)\\ &= \arg\max_{1\leqslant k\leqslant t} \prod_{1\leqslant i\leqslant m}\Pr(B_i^+|c_k)\prod_{1\leqslant j\leqslant n}\Pr(B_j^-|c_k).\end{aligned} \quad (4)$$

(5) can be obtained by applying Bayes rule to (4).

$$\begin{aligned}\mathit{Obj} = \arg\max_{1\leqslant k\leqslant t} &\prod_{1\leqslant i\leqslant m}\frac{\Pr(c_k|B_i^+)\Pr(B_i^+)}{\Pr(c_k)}\\ &\cdot\prod_{1\leqslant j\leqslant n}\frac{\Pr(c_k|B_j^-)\Pr(B_j^-)}{\Pr(c_k)}.\end{aligned} \quad (5)$$

Considering $\prod_{1\leqslant i\leqslant m}\Pr(B_i^+)\prod_{1\leqslant j\leqslant n}\Pr(B_j^-)$ is a constant which can be dropped, and reminding that $\Pr(c_k)$ can be dropped as that has been done in (3) since we assume uniform prior, (5) can be simplified into (6).

$$\mathit{Obj} = \arg\max_{1\leqslant k\leqslant t} \prod_{1\leqslant i\leqslant m}\Pr(c_k|B_i^+)\prod_{1\leqslant j\leqslant n}\Pr(c_k|B_j^-). \quad (6)$$

(6) is the general expression for the class label with the maximum posterior probability. Concretely, the class label for a specific point $\boldsymbol{x}$ in the feature space can be expressed as (7), where $(\boldsymbol{x} = c_k)$ means the label of $\boldsymbol{x}$ is c_k.

$$\begin{aligned}\mathit{Obj}^{\boldsymbol{x}} = \arg\max_{1\leqslant k\leqslant t} &\prod_{1\leqslant i\leqslant m}\Pr(\boldsymbol{x} = c_k|B_i^+)\\ &\cdot\prod_{1\leqslant j\leqslant n}\Pr(\boldsymbol{x} = c_k|B_j^-).\end{aligned} \quad (7)$$

If we want to identify a single point in the feature space where the maximum posterior probability of a specific class label, say c_h, is the biggest, then the point can be located according to (8).

$$\begin{aligned}\hat{\boldsymbol{x}} &= \arg\max_{\boldsymbol{x}} \Pr(\mathit{Obj}^{\boldsymbol{x}} = c_h)\\ &= \arg\max_{\boldsymbol{x}} \prod_{1\leqslant i\leqslant m}\Pr(\boldsymbol{x} = c_h|B_i^+)\\ &\quad\cdot\prod_{1\leqslant j\leqslant n}\Pr(\boldsymbol{x} = c_h|B_j^-).\end{aligned} \quad (8)$$

It is noteworthy that (8) is exactly the formal definition of the general *maximum diverse density*[10] optimized by the DIVERSE DENSITY algorithm. This implies that multi-instance learner can be obtained by shifting the focus of the standard Bayesian classifier.

3.2 CITATION-*k*NN

CITATION-*k*NN[19] is a nearest neighbor style algorithm, which borrows the notion of *citation* and *reference* of scientific literatures in the way that a bag is labeled by analyzing not only its neighboring bags but also the bags that regard the concerned bag as a neighbor.

It is evident that for any nearest neighbor style algorithm, the key lies in the definition of the distance metric which is utilized to measure the distance between different objects. Below we show that the key of CITATION-*k*NN, i.e., the definition of the *minimal Hausdorff distance*, can be attained by modifying standard k-nearest neighbor algorithm according to the rule, i.e., shifting the focus from discriminating the instances to discriminating the bags.

In standard k-nearest neighbor algorithm, each object, or instance, is regarded as a feature vector in the feature space. For two different feature vectors, i.e., $\boldsymbol{a}$ and $\boldsymbol{b}$, the distance between them can be written as (9). Usually $\|\boldsymbol{a}-\boldsymbol{b}\|$ is realized by the Euclidean distance.

$$\mathrm{Dist}(\boldsymbol{a},\boldsymbol{b}) = \|\boldsymbol{a}-\boldsymbol{b}\|. \quad (9)$$

(9) is fine to be instantiated when the goal is to discriminate the instances, but for discriminating the bags (9) must be extended because now we should measure the distance between different bags.

Suppose there are two different bags, i.e., $\boldsymbol{A} = \{\boldsymbol{a}_1,\boldsymbol{a}_2,\ldots,\boldsymbol{a}_m\}$ and $\boldsymbol{B} = \{\boldsymbol{b}_1,\boldsymbol{b}_2,\ldots,\boldsymbol{b}_n\}$ where $\boldsymbol{a}_i$ $(1\leqslant i\leqslant m)$ and $\boldsymbol{b}_j$ $(1\leqslant j\leqslant n)$ are the instances. It is obvious that they can be regarded as two feature vector sets, where each $\boldsymbol{a}_i$ $(1\leqslant i\leqslant m)$ or $\boldsymbol{b}_j$ $(1\leqslant j\leqslant n)$ is a feature vector in the feature space. Therefore, the problem of measuring the distance between different bags is in fact the problem of measuring the distance between different feature vector sets.

Geometrically, a feature vector set can be viewed as a group of points enclosed in a contour in the feature space. Thus, an intuitive way to measure the distance between two feature vector sets is to define their distance as the distance between their nearest feature vectors, as illustrated in Fig.1.

Formally, such a distance metric can be written as (10).

$$\mathrm{Dist}(\boldsymbol{A},\boldsymbol{B}) = \min_{\substack{1\leqslant i\leqslant m\\1\leqslant j\leqslant n}}(\mathrm{Dist}(\boldsymbol{a}_i,\boldsymbol{b}_j)) = \min_{\boldsymbol{a}\in\boldsymbol{A}}\min_{\boldsymbol{b}\in\boldsymbol{B}}\|\boldsymbol{a}-\boldsymbol{b}\|. \quad (10)$$

It is noteworthy that (10) is exactly the formal definition of the *minimum Hausdorff distance*[19] employed by the CITATION-kNN algorithm to measure the distance between different bags. This implies that multi-instance learner can be obtained by shifting the focus of the standard k-nearest neighbor algorithm.

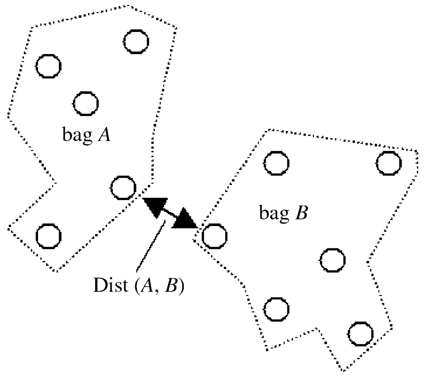

Fig.1. An Intuitive way to define the distance between bags.

Note that although Wang and Zucker admitted that using the *minimal Hausdorff distance* does allow k-nearest neighbor algorithm to be adapted to multi-instance learning, they also indicated that it is not sufficient[19]. This is because the common prediction-generating scheme employed by k-nearest neighbor algorithms, i.e., *majority voting*, may be confused by false positive instances in positive bags in some cases. Therefore as mentioned before, the notion of citation and reference is introduced for obtaining the optimal performance.

However, the utilization of the notion of citation and reference does not change the fact that the *minimal Hausdorff distance* is the key in adapting k-nearest neighbor algorithms to multi-instance learning. This is because the notion of citation and reference can also be introduced to improve the performance of k-nearest neighbor algorithms dealing with supervised learning tasks. More importantly, a k-nearest neighbor algorithm employing common distance metrics such as the Euclidean distance cannot work in multi-instance scenarios, even though it were facilitated with the notion of citation and reference; while a k-nearest neighbor algorithm employing the *minimal Hausdorff distance* can work in multi-instance scenarios, even though it does not take citation and reference into account.

In fact, by analyzing the experimental data presented in the Appendix of Wang and Zucker's paper[19], it could be found that when k is 3, the performance of the k-nearest neighbor algorithm employing the *minimal Hausdorff distance* without utilizing citation and reference is already comparable to or even better than that of some multi-instance learning algorithms such as RELIC[23] and MULTINST[8] on *Musk*1, and RIPPER-MI[24] and GFS ELIM-COUNT APR[2] on *Musk*2. Moreover, if the fact that the occurrence of positive bags is much smaller than that of negative bags has been considered so that a new bag is negatively labeled when tie occurs in determining its label, the performance of the k-nearest neighbor algorithm employing the *minimal Hausdorff distance* without utilizing citation and reference would be 90.2% on *Musk*1 and 82.4% on *Musk*2, respectively, when k is 2. It is interesting that this reaches the best performance of another multi-instance k-nearest neighbor algorithm, i.e., BAYESIAN-kNN[19].

3.3 ID3-MI

ID3-MI[24] is a decision tree algorithm, which follows the divide-and-conquer way of popular decision trees, i.e., training data falling into a tree node will be split into different subnodes unless almost all the data on the concerning node belonging to the same class, if pruning is not considered.

Roughly speaking, a decision tree algorithm has two important components, i.e., the strategies of how to choose *tests* to split the tree nodes and how to make predictions using the tree. Since the ID3-MI algorithm makes predictions in the same way as standard decision tree does, i.e., the label of an unseen bag is determined by the label of the leaf node into which the bag falls, it is evident that the key of ID3-MI lies in the formal definition of the *multi-instance entropy*, i.e., the criterion used by ID3-MI to select candidate tests to split the tree nodes. Below we show that such a definition can be attained by modifying standard decision tree according to the rule, i.e., shifting the focus from discriminating the instances to discriminating the bags.

Given data set D containing p positive instances and n negative instances, the entropy of D corresponding to the classification is shown as (11).

$$\mathrm{Info}(D) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right). \tag{11}$$

Assuming attribute V is chosen as the test, which partitions D into $\{D_1, D_2, \ldots, D_l\}$, then the entropy of D after partitioning with V is shown as (12), where $|\mathcal{X}|$ denotes the size of the set $\mathcal{X}$, i.e., the numbers of instances contained in $\mathcal{X}$.

$$\mathrm{Info}(D, V) = \sum_{i=1}^{l} \frac{|D_i|}{|D|}\,\mathrm{Info}\,(D_i). \tag{12}$$

The *information gain* of V on D can be computed according to (13).

$$\begin{aligned}\mathrm{Gain}(D, V) &= \mathrm{Info}(D) - \mathrm{Info}(D, V)\\ &= \mathrm{Info}(D) - \sum_{i=1}^{l} \frac{|D_i|}{|D|}\mathrm{Info}(D_i).\end{aligned} \tag{13}$$

(13) is fine for choosing appropriate tests for a decision tree when the goal is to discriminate the instances, but for discriminating the bags it is necessary to count the number of positive bags and negative bags instead of that of positive instances and negative instances contained in D and D_i.

Let $\pi(\mathcal{X})$ and $\nu(\mathcal{X})$ respectively denote the numbers of positive and negative bags which have instances appearing in data set $\mathcal{X}$. Then the entropies of D defined at the bag level before and after partitioning with V are shown in (14) and (15), respectively.

$$\mathrm{Info}_{\mathrm{multi}}(D) = -\frac{\pi(D)}{\pi(D)+\nu(D)}\log_2\Big(\frac{\pi(D)}{\pi(D)+\nu(D)}\Big) - \frac{\nu(D)}{\pi(D)+\nu(D)}\log_2\Big(\frac{\nu(D)}{\pi(D)+\nu(D)}\Big), \qquad (14)$$

$$\mathrm{Info}_{\mathrm{multi}}(D, V) = \sum_{i=1}^{l}\frac{\pi(D_i)+\nu(D_i)}{\pi(D)+\nu(D)}\mathrm{Info}_{\mathrm{multi}}(D_i). \qquad (15)$$

The information gain of V on D defined at the bag level is computed according to (16).

$$\mathrm{Gain}_{\mathrm{multi}}(D, V) = \mathrm{Info}_{\mathrm{multi}}(D) - \mathrm{Info}_{\mathrm{multi}}(D, V) = \mathrm{Info}_{\mathrm{multi}}(D) - \sum_{i=1}^{l}\frac{\pi(D_i)+\nu(D_i)}{\pi(D)+\nu(D)}\mathrm{Info}_{\mathrm{multi}}(D_i). \qquad (16)$$

It is noteworthy that (16) is exactly the formal definition of the *multi-instance entropy*[24] employed by the ID3-MI algorithm to select candidate tests. This implies that multi-instance learner can be obtained by shifting the focus of the standard decision tree algorithm.

Note that there is another way to modify standard decision tree. As mentioned before, besides the criterion for choosing the tests, a decision tree algorithm has another important component, i.e., the strategy of how to make predictions using the tree. Through shifting the focus of this strategy from discriminating the instances to discriminating the bags, standard decision tree can also be adapted to multi-instance learning, which is exactly the way adopted by Ruffo's Relic algorithm[23].

3.4 Ripper-mi

Ripper-mi[24] is a rule induction algorithm, which follows the separate-and-conquer way of popular rule inducers, i.e., the rules are induced one by one and all training data covered by a rule will be removed after the rule is found.

In general, a rule is grown on a growing data set and then pruned on a pruning data set, where the definition of *coverage* that expresses the numbers of instances covered by the rule is very important no matter whether the rule is being grown or pruned. In detail, when a rule is being grown, rule conditions can be ceaselessly added to the rule until the rule does not cover any negative instances in the growing data set; while when a rule is being pruned, rule conditions can be ceaselessly removed from the rule to maximize some evaluation function, such as the one shown in (17), where p and n respectively denote the numbers of positive and negative instances in the pruning data set covered by the rule.

$$v = \frac{p-n}{p+n}. \qquad (17)$$

Since the only difference between Ripper-mi and its corresponding supervised learner, i.e., Ripper, is on the definition of coverage[24], it is evident that the key of Ripper-mi lies in the formal definition of the *multi-instance coverage*, i.e., the function used by Ripper-mi to measure the coverage of a rule. Below we show that such a definition can be attained by modifying standard rule inducer according to the rule, i.e., shifting the focus from discriminating the instances to discriminating the bags.

Given data set D, the coverage of rule R can be measured according to (18), where $\mathrm{Cover}(R, instance_i)$ means that the i-th instance in D is covered by R, that is, R is more general than $instance_i$ if the latter is being regarded as a rule.

$$\mathrm{Coverage}(R) = |\{instance_i|\mathrm{Cover}(R, instance_i)\}|. \qquad (18)$$

(18) is fine when the goal is to discriminate the instances, but for discriminating the bags the coverage function must be extended. For this purpose we should define in which situation a bag can be regarded as being covered by rule R. If we adopt the definition shown as (19), then the coverage function at the bag level is shown as (20), where bag_i denotes the i-th bag in D.

$$\mathrm{Cover}_{\mathrm{multi}}(R, bag) = \exists(instance \in bag)\mathrm{Cover}(R, instance), \qquad (19)$$

$$\mathrm{Coverage}_{\mathrm{multi}}(R) = |\{bag_i|\mathrm{Cover}_{\mathrm{multi}}(R, bag_i)\}|. \qquad (20)$$

It is noteworthy that (20) is exactly the formal definition of the *multi-instance coverage*[24] employed by the Ripper-mi algorithm to measure the coverage of rules. This implies that multi-instance learner can be obtained by shifting the focus of the standard rule induction algorithm.

3.5 BP-MIP

BP-MIP[21] is a feedforward neural network algorithm, which compares the actual output of the network with the desired output, and then backpropagates the error and updates the weights of the connections and the thresholds of the units.

It is evident that the key of the BP-MIP algorithm lies in the formal definition of the *multi-instance error function*[①], which is the function used to measure the error of the neural network and therefore is the objective to be optimized by the algorithm. Below we show that such a definition can be attained by modifying standard feedforward neural network algorithm according to

[①] This error function was not named in Zhou and Zhang's paper[21]. Here we call it *multi-instance error function* for convenience.

the rule, i.e., shifting the focus from discriminating the instances to discriminating the bags.

Given data set D comprising l instances, the error of the neural network is usually computed according to (21), where o_i and d_i is the actual output and desired output on the i-th instance, respectively.

$$E = \sum_{i=1}^{l} E_i = \sum_{i=1}^{l} \frac{1}{2}(o_i - d_i)^2. \tag{21}$$

(21) is fine when the goal is to discriminate the instances, but for discriminating the bags the error function must be extended. For this purpose we must define what is the actual output of a bag. If we adopt the definition shown as (22), where o_{ij} denotes the actual output of the j-th instance of the i-th bag in D and m_i denotes the total number of instances in the i-th bag, then the error function at the bag level can be defined as (23). Note that instances belonging to the same bag are input to the network successively.

$$o_i = \max_{1 \leqslant j \leqslant m_i} o_{ij}, \tag{22}$$

$$E = \sum_{i=1}^{l} E_i = \sum_{i=1}^{l} \frac{1}{2}\big(\max_{1 \leqslant j \leqslant m_i} o_{ij} - d_i\big)^2. \tag{23}$$

It is noteworthy that (23) is exactly the formal definition of the *multi-instance error function*[21] which is optimized by the BP-MIP algorithm. This implies that multi-instance learner can be obtained by shifting the focus of the standard feedforward neural network algorithm.

In this section we have shown that supervised learners can be adapted to multi-instance learning according to the general rule, i.e., *shifting the focuses of supervised learners from the discrimination on the instances to the discrimination on the bags.* This general rule provides an insight into the working mechanism of many multi-instance learners, and illuminates the design of new multi-instance learning algorithms. Actually, so far as we identify some important components of a given supervised learner, we can derive a multi-instance learner by picking one of these components to modify according to the general rule. Note that this implies that from a supervised learner we may derive several multi-instance learners, which has been implicitly shown in Subsection 3.3 where different multi-instance decision trees were derived from the standard decision tree algorithm. Moreover, the general rule establishes a bridge connecting multi-instance learning with supervised learning, which suggests that some mechanisms useful in supervised learning may also be useful in multi-instance learning. This will be explored in the next section.

4 Multi-Instance Ensemble

Ensemble learning paradigms train multiple versions of a base learner to solve a problem. Since ensembles are usually more accurate than single learners, one of the most active areas of research in supervised learning has been ensemble learning[3].

Since it has been shown in Section 3 that multi-instance learning has close connection with supervised learning, a consequent exciting idea is to see whether ensemble learning paradigms can be used to enhance multi-instance learners, as it is well-known that they can be used to enhance supervised learners. Here we call ensembles of multi-instance learners as *multi-instance ensembles*.

4.1 Method

A lot of ensemble learning algorithms have been developed, such as AdaBoost[41], Bagging[42], Gasen[43], etc. In this subsection, we use a relatively simple algorithm, i.e., Bagging, to build multi-instance ensembles.

Bagging employs *bootstrap sampling*[44] to generate multiple training sets from the original training set and then trains component learners, i.e., different versions of the base learner, from each generated training set. The predictions of the component learners are combined via *majority voting*, where the class label receiving the biggest number of votes is regarded as the final prediction.

In this paper we try to build multi-instance ensembles of seven different base learners, including Diverse Density, Citation-$k$nn, Ripper-mi, BP-MIP, Relic, Iterated-Discrim Apr and EM-DD. It is obvious that these base learners span a wide spectrum including Bayesian learner, nearest neighbor learner, rule inducer, decision tree, neural network, etc. The first four learners have been analyzed in Section 3. Relic is a multi-instance decision tree, which has been briefly discussed in Subsection 3.3. We use Relic instead of ID3-MI here simply because we have got its code. The reason for choosing the other two base learners are briefly explained as follows.

Iterated-Discrim Apr is the best among the Apr algorithms proposed by Dietterich *et al.*[2] It works quite well although it is one of the earliest multi-instance learning algorithms. Actually, Dietterich *et al.*[2] indicated that since the Apr algorithms had been optimized to the *Musk* data, the performance of Iterated-Discrim Apr might be the upper bound of this benchmark test.

EM-DD[17] incorporates Diverse Density into an EM framework. It converts the multi-instance problem to a single-instance setting by using EM to estimate the instance which is responsible for the label of the bag. Zhang and Goldman[17] reported that the predictive error rates of EM-DD are 3.2% and 4.0% on *Musk*1 and *Musk*2, respectively, which is the best performance on the *Musk* data before our work[27]. Note that the performance of EM-DD reported in [17] has already exceeded the upper bound of this benchmark test anticipated by Dietterich *et al.*[2]

Table 1. *Musk* Data (72 molecules are shared in both data sets)

Data Set	Dim.	Bags			Instances	Instances per Bag		
		Total	Musk	Non-Musk		Min	Max	Ave.
*Musk*1	166	92	47	45	476	2	40	5.17
*Musk*2	166	102	39	63	6,598	1	1,044	64.69

Table 2. Predictive Error Rates (%) of Multi-Instance Ensembles and Corresponding Single Learners

Learner	*Musk*1		*Musk*2	
	Single	Ensemble	Single	Ensemble
Diverse Density	11.1	8.2	17.5	11.0
Citation-*k*nn	7.6	5.2	13.7	12.9
Ripper-mi	12.0	9.1	23.0	22.6
BP-MIP	16.3	13.0	19.6	15.7
Relic	16.3	15.9	12.7	11.0
Iterated-Discrim Apr	7.6	7.2	10.8	6.9
EM-DD	3.2	3.1	4.0	3.0

4.2 Experimental Results

The experiments are performed on the *Musk* data, which is a popularly used real-world benchmark for multi-instance learners.

The *Musk* data were generated in Dietterich *et al.*'s research on drug activity prediction[2]. Here each molecule is regarded as a bag, and its alternative low-energy shapes are regarded as the instances in the bag. A positive bag corresponds to a molecule qualified to make a certain drug while a negative bag corresponds to a molecule unqualified to make the drug. In order to represent the shapes, a molecule was placed at a standard position and orientation, and then a set of 162 rays emanating from the origin was constructed such that the molecular surface was sampled approximately uniformly. There were also four features that represented the position of an oxygen atom on the molecular surface. Therefore each instance in the bags is represented by 166 numerical attributes.

There are two data sets, i.e., *Musk*1 and *Musk*2, both of which are publicly available at the UCI Machine Learning Repository[45]. *Musk*1 contains 47 positive bags and 45 negative bags, and the number of instances contained in each bag ranges from 2 to 40. *Musk*2 contains 39 positive bags and 63 negative bags, and the number of instances contained in each bag ranges from 1 to 1,044. Detailed information on the *Musk* data is tabulated in Table 1.

Ten-fold cross validation is performed on each *Musk* data set. In each fold, Bagging is employed to build an ensemble for each of the seven base multi-instance learners. Each ensemble comprises five versions of the base learner. The predictive error rates of the ensembles are shown in Table 2. For comparison, the best results of the single multi-instance learners reported in the literatures[2,10,17,19,21,23,24] ② are also included in Table 2.

Table 2 shows that Bagging can significantly improve the generalization ability of all the investigated multi-instance learners. It is impressive that even for the strongest multi-instance learner, i.e., EM-DD, the performance can also be enhanced by such a relatively simple ensemble learning algorithm. In fact, the EM-DD ensemble achieves the best performance up to date on both the *Musk* data sets, i.e., predictive error rate 3.1% on *Musk*1 and 3.0% on *Musk*2.

Since the process of building multi-instance ensembles has not being geared to any specific data, such a paradigm can be applied to any multi-instance problems. It is reasonable to anticipate that such a paradigm may return more profit on difficult problems where no single multi-instance learners works very well. Moreover, the experiments reported in this section also suggest ensemble learning paradigms be investigated in more scenarios, not to be limited in supervised learning.

5 Conclusion

When formalizing multi-instance learning, Dietterich *et al.*[2] raised an open problem, i.e., *how to design multi-instance modifications for popular machine learning algorithms*. This paper presents a general rule, i.e., *supervised learners can be adapted to multi-instance learning by shifting their focuses from the discrimination on the instances to the discrimination on the bags*, which gives an answer to the open problem. Indeed, so far as we identify some important components of a given supervised learner, we can derive a multi-instance learner by modifying one of these components of the supervised learner according to the general rule.

The general rule also establishes a bridge connecting multi-instance learning with supervised learning, which suggests that some mechanisms useful in supervised learning may also be useful in multi-instance learning. Based on this recognition, this paper tries to build

②Here the performance of EM-DD reported in the paper which proposed EM-DD[17] is included. Another result of EM-DD was reported in [26].

multi-instance ensembles to solve multi-instance problems. Experiments show that all the studied multi-instance learners can be enhanced by a relatively simple ensemble learning algorithm, and the best result up to date on the *Musk* benchmark test is achieved by EM-DD ensemble. These results not only validate the strength of multi-instance ensembles but also suggest ensemble learning paradigms be investigated in more scenarios, not to be limited in supervised learning.

Note that although this paper reveals that multi-instance learners can be derived by shifting the focuses of supervised learners from the discrimination on the instances to the discrimination on the bags, that is, adapting single-instance algorithms to the multi-instance representation, a recent work shows that there is an opposite way to the solution of multi-instance problems, that is, adapting the multi-instance representation to the single-instance algorithms[46]. Exploring the cons and pros of these two opposite ways is an interesting issue for future work.

Most current multi-instance learners can only predict the labels of the bags. In many applications it will be more desirable if the labels of the instances in the bags can be predicted. This has attracted some attention recently. Blockeel *et al.*[47] proposed MITI, a multi-instance decision tree algorithm, for this purpose. Zhou *et al.*[48] proposed the C*k*NN-ROI algorithm, a variant of CITATION-*k*NN, and applied it to content-based image retrieval. It is evident that designing other algorithms for predicting the labels of instances in multi-instance learning is another interesting future issue.

Acknowledgements The author wants to thank Min-Ling Zhang for running the experiments, Giancarlo Ruffo for providing the code of RELIC, and Nicolas Bredèche for providing the code of RIPPER-MI. A preliminary version of this paper has been presented at ECML'03 (the 14th European Conference on Machine Learning).

References

[1] Maron O. Learning from ambiguity [Dissertation]. Department of Electrical Engineering and Computer Science, MIT, Jun. 1998.

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

[3] Dietterich T G. Machine learning research: Four current directions. *AI Magazine*, 1997, 18(4): 97–136.

[4] Lindsay R, Buchanan B, Feigenbaum E, Lederberg J. Applications of Artificial Intelligence to Organic Chemistry: The DENDRAL Project. New York: McGraw-Hill, 1980.

[5] Zucker J-D, Ganascia J-G. Changes of representation for efficient learning in structural domains. In *Proc. the 13th Int. Conf. Machine Learning*, Bary, Italy, 1996, pp.543–551.

[6] Long P M, Tan L. PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. *Machine Learning*, 1998, 30(1): 7–21.

[7] Auer P, Long P M, Srinivasan A. Approximating hyper-rectangles: Learning and pseudo-random sets. *Journal of Computer and System Sciences*, 1998, 57(3): 376–388.

[8] Auer P. On learning from multi-instance examples: Empirical evaluation of a theoretical approach. In *Proc. the 14th Int. Conf. Machine Learning*, Nashville, TN, 1997, pp.21–29.

[9] Blum A, Kalai A. A note on learning from multiple-instance examples. *Machine Learning*, 1998, 30(1): 23–29.

[10] Maron O, Lozano-Pérez T. A Framework for Multiple-Instance Learning. Advances in Neural Information Processing Systems 10, Jordan M I, Kearns M J, Solla S A (eds.), Cambridge, MA: MIT Press, 1998, pp.570–576.

[11] Maron O, Ratan A L. Multiple-instance learning for natural scene classification. In *Proc. the 15th Int. Conf. Machine Learning*, Madison, WI, 1998, pp.341–349.

[12] McGovern A, Barto A G. Automatic discovery of subgoals in reinforcement learning using diverse density. In *Proc. the 18th Int. Conf. Machine Learning*, Williamstown, MA, 2001, pp.361–368.

[13] Yang C, Lozano-Pérez T. Image database retrieval with multiple-instance learning techniques. In *Proc. the 16th Int. Conf. Data Engineering*, San Diego, CA, 2000, pp.233–243.

[14] Zhou Z-H, Zhang M-L, Chen K-J. A novel bag generator for image database retrieval with multi-instance learning techniques. In *Proc. the 15th IEEE Int. Conf. Tools with Artificial Intelligence*, Sacramento, CA, 2003, pp.565–569.

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

[16] Dempster A P, Laird N M, Rubin D B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistics Society* — B, 1977, 39(1): 1–38.

[17] Zhang Q, Goldman S A. EM-DD: An Improved Multi-Instance Learning Technique. Advances in Neural Information Processing Systems 14, Dietterich T G, Becker S, Ghahramani Z (eds.), Cambridge, MA: MIT Press, 2002, pp.1073–1080.

[18] Zhang Q, Yu W, Goldman S A, Fritts J E. Content-based image retrieval using multiple-instance learning. In *Proc. the 19th Int. Conf. Machine Learning*, Sydney, Australia, 2002, pp.682–689.

[19] Wang J, Zucker J-D. Solving the multiple-instance problem: A lazy learning approach. In *Proc. the 17th Int. Conf. Machine Learning*, San Francisco, CA, 2000, pp.1119–1125.

[20] Zhou Z-H, Jiang K, Li M. Multi-instance learning based web mining. *Applied Intelligence*, 2005, 22(2): 135–147.

[21] Zhou Z-H, Zhang M-L. Neural networks for multi-instance learning. Technical Report, AI Lab, Computer Science & Technology Department, Nanjing University, Nanjing, China, Aug. 2002.

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

[23] Ruffo G. Learning single and multiple instance decision tree for computer security applications [Dissertation]. Department of Computer Science, University of Turin, Torino, Italy, Feb. 2000.

[24] Chevaleyre Y, Zucker J-D. Solving multiple-instance and multiple-part learning problems with decision trees and decision rules. Application to the mutagenesis problem. *Lecture Notes in Artificial Intelligence 2056*, Stroulia E, Matwin S (eds.), Berlin: Springer, 2001, pp.204–214.

[25] Gärtner T, Flach P A, Kowalczyk A, Smola A J. Multi-instance kernels. In *Proc. the 19th Int. Conf. Machine Learning*, Sydney, Australia, 2002, pp.179–186.

[26] Andrews S, Tsochantaridis I, Hofmann T. Support Vector Machines for Multiple-Instance Learning. Advances in Neural Information Processing Systems 15, Becker S, Thrun S, Obermayer K (eds.), Cambridge, MA: MIT Press, 2003, pp.561–568.

[27] Zhou Z-H, Zhang M-L. Ensembles of multi-instance learners. *Lecture Notes in Artificial Intelligence 2837*, Lavrač N, Gamberger D, Blockeel H, Todorovski L (eds.), Berlin: Springer, 2003, pp.492–502.

[28] Xu X, Frank E. Logistic regression and boosting for labeled bags of instances. *Lecture Notes in Artificial Intelligence*

3056, Dai H, Srikant R, Zhang C (eds.), Berlin: Springer, 2004, pp.272–281.

[29] Zhang M-L, Zhou Z-H. Adapting RBF neural networks to multi-instance learning. *Neural Processing Letters*, 2006, 23(1): 1–26.

[30] Amar R A, Dooly D R, Goldman S A, Zhang Q. Multiple-instance learning of real-valued data. In *Proc. the 18th Int. Conf. Machine Learning*, Williamstown, MA, 2001, pp.3–10.

[31] Ray S, Page D. Multiple instance regression. In *Proc. the 18th Int. Conf. Machine Learning*, Williamstown, MA, 2001, pp.425–432.

[32] Dooly D R, Zhang Q, Goldman S A, Amar R A. Multiple-instance learning of real-valued data. *Journal of Machine Learning Research*, 2002, 3: 651–678.

[33] Goldman S A, Scott S D. Multiple-instance learning of real-valued geometric patterns. *Annals of Mathematics and Artificial Intelligence*, 2003, 39(3): 259–290.

[34] Weidmann N, Frank E, Pfahringer B. A two-level learning method for generalized multi-instance problem. *Lecture Notes in Artificial Intelligence 2837*, Lavrač N, Gamberger D, Blockeel H, Todorovski L (eds.), Berlin: Springer, 2003, pp.468–479.

[35] Scott S D, Zhang J, Brown J. On generalized multiple-instance learning. Technical Report UNL-CSE-2003-5, Department of Computer Science, University of Nebraska, Lincoln, NE, 2003.

[36] Tao Q, Scott S, Vinodchandran N V, Osugi T T. SVM-based generalized multiple-instance learning via approximate box counting. In *Proc. the 21st Int. Conf. Machine Learning*, Banff, Canada, 2004, pp.779–806.

[37] Tao Q, Scott S, Vinodchandran N V, Osugi T T, Mueller B. An extended kernel for generalized multiple-instance learning. In *Proc. the 16th IEEE Int. Conf. Tools with Artificial Intelligence*, Boca Raton, FL, 2004, pp.272–277.

[38] De Raedt L. Attribute-value learning versus inductive logic programming: The missing links. *Lecture Notes in Artificial Intelligence 1446*, Page D (ed.), Berlin: Springer, 1998, pp.1–8.

[39] Alphonse É, Matwin S. Filtering multi-instance problems to reduce dimensionality in relational learning. *Journal of Intelligent Information Systems*, 2004, 22(1): 23–40.

[40] McGovern A, Jensen D. Identifying predictive structures in relational data using multiple instance learning. In *Proc. the 20th Int. Conf. Machine Learning*, Washington DC, 2003, pp.528–535.

[41] Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting. *Lecture Notes in Computer Science 904*, Vitányi P M B (ed.), Berlin: Springer, 1995, pp.23–37.

[42] Breiman L. Bagging predictors. *Machine Learning*, 1996, 24(2): 123–140.

[43] Zhou Z-H, Wu J, Tang W. Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, 2002, 137(1-2): 239–263.

[44] Efron B, Tibshirani R. An Introduction to the Bootstrap. New York: Chapman & Hall, 1993.

[45] Blake C, Keogh E, Merz C J. UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, CA, 1998, http://www.ics.uci.edu/~mlearn/MLRepository.html.

[46] Zhou Z-H, Zhang M-L. Solving multi-instance problems with classifier ensemble based on constructive clustering. *Knowledge and Information Systems*, 2006, in press.

[47] Blockeel H, Page D, Srinivasan A. Multi-instance tree learning. In *Proc. the 22nd Int. Conf. Machine Learning*, Bonn, Germany, 2005.

[48] Zhou Z-H, Xue X-B, Jiang Y. Locating regions of interest in CBIR with multi-instance learning techniques. *Lecture Notes in Artificial Intelligence 3809*, Zhang S, Jarvis R (eds.), Berlin: Springer, 2005, pp.92–101.

Zhi-Hua Zhou received the B.Sc., M.Sc. and Ph.D. degrees in computer science from Nanjing University, China, in 1996, 1998 and 2000, respectively, all with the highest honor. He joined the Department of Computer Science & Technology of Nanjing University as a lecturer in 2001, and at present he is a professor and head of the LAMDA group. His research interests are in artificial intelligence, machine learning, data mining, information retrieval, pattern recognition, neural computing, and evolutionary computing. In these areas he has published over 70 technical papers in refereed international journals or conference proceedings. He has won various awards. He is an associate editor of *Knowledge and Information Systems*, on the editorial boards of *Artificial Intelligence in Medicine*, *International Journal of Data Warehousing and Mining*, *Journal of Computer Science and Technology* and *Journal of Software*, and guest editor/co-editor of *ACM/Springer Multimedia Systems*, *The Computer Journal*, etc. He served as program committee member for various international conferences and chaired a number of native conferences. He is a senior member of China Computer Federation (CCF) and the vice chair of CCF Artificial Intelligence & Pattern Recognition Society, an executive committee member of Chinese Association of Artificial Intelligence (CAAI), the vice chair and chief secretary of CAAI Machine Learning Society, a member of AAAI and ACM, and a senior member of IEEE and IEEE Computer Society.