Under consideration for publication in Knowledge and Information
Systems

Solving Multi-Instance Problems with Classifier Ensemble Based on Constructive Clustering

Zhi-Hua Zhou, Min-Ling Zhang

National Laboratory for Novel Software Technology, Nanjing University, China

Abstract. In multi-instance learning, the training set is composed of labeled *bags* each consists of many unlabeled instances, that is, an object is represented by a set of feature vectors instead of only one feature vector. Most current multi-instance learning algorithms work through adapting single-instance learning algorithms to the multi-instance representation, while this paper proposes a new solution which goes at an opposite way, that is, adapting the multi-instance representation to single-instance learning algorithms. In detail, the instances of all the bags are collected together and clustered into d groups at first. Each bag is then re-represented by d binary features, where the value of the i-th feature is set to one if the concerned bag has instances falling into the i-th group and zero otherwise. Thus, each bag is represented by one feature vector so that single-instance classifiers can be used to distinguish different classes of bags. Through repeating the above process with different values of d, many classifiers can be generated and then they can be combined into an ensemble for prediction. Experiments show that the proposed method works well on standard as well as generalized multi-instance problems.

Keywords: Machine Learning; Multi-instance Learning; Classification; Clustering; Ensemble Learning; Knowledge Representation; Constructive Induction

1. Introduction

During the past decades, *learning from examples* becomes one of the most flourishing areas in machine learning. According to the *label ambiguity*, i.e. ambiguity of the labels of training examples, research in this area can be roughly categorized into three learning frameworks, i.e. supervised learning, unsupervised learning,

Received April 25, 2005
Revised September 17, 2005
Accepted January 14, 2006

and reinforcement learning (Maron, 1998). Supervised learning attempts to learn a concept for correctly labeling unseen instances, where the training instances are with known labels and therefore the label ambiguity is the 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 label 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 that can be viewed as delayed labels, and therefore its label ambiguity is between that of supervised learning and unsupervised learning.

The term *multi-instance learning* was coined by Dietterich et al. (1997) when they were investigating the problem of drug activity prediction. Here the training set is composed of labeled *bags* each consists of many unlabeled instances, and the goal is to learn some concept from the training set for correctly labeling unseen bags. A bag is positively labeled if it contains at least one positive instance and negatively labeled otherwise. Note that a positive bag may contain hundreds of instances, among which maybe only one is really positive. This implies that the *false positive noise* may be overwhelmingly high if a multi-instance problem were regarded as a typical supervised learning problem through simply assigning the label of a bag to the instances in the bag. Therefore, common single-instance learning algorithms can hardly obtain good performance when being applied to multi-instance problems directly. In fact, it has been shown that learning algorithms ignoring the characteristics of multi-instance problems, such as the traditional decision trees and neural networks, could not work well in this scenario (Dietterich et al., 1997).

Actually, multi-instance learning is quite unique if looking it from the aspect of label ambiguity. 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; in contrast to reinforcement learning where the labels of the training instances are delayed, in multi-instance learning there is not any time delay. Since multi-instance problems extensively exist but are unique to these addressed by previous learning frameworks, multi-instance learning has been regarded as a new learning framework (Maron, 1998), and attracted much attention of the machine learning community.

This paper deems that the difficulty of multi-instance learning mainly lies in that it is accompanied with an unusual representation. Actually, a bag corresponds to a real-world object while the instances in the bag are feature vectors describing the object. In contrast to typical machine learning settings where an object is represented by only one feature vector, in multi-instance learning an object is represented by a set of (more than one) feature vectors.

Zhou and Zhang (2003) showed that single-instance 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. In fact, most current multi-instance learning algorithms can be viewed as going along this way, that is, adapting single-instance learning algorithms to the multi-instance representation.

In this paper, a new way to the solution of multi-instance learning is proposed, that is, adapting the multi-instance representation to single-instance learning algorithms. In detail, the bags are re-represented by features generated with the help of a clustering process such that the multi-instance problem becomes a single-instance supervised learning problem, which is then solved by an ensemble of classifiers. Since the clustering process is used to help change the representation, it can be viewed as a specific scheme of constructive induction (Bloedorn and Michalski, 1998). Therefore, the proposed method is called CCE, i.e. Constructive Clustering based Ensemble. Experiments show that CCE can work well on not only standard multi-instance problems, but also generalized multi-instance problems (Weidmann et al., 2003).

The rest of this paper is organized as follows. Section 2 briefly introduces standard and generalized multi-instance learning. Section 3 presents CCE. Section 4 reports on the experiments. Section 5 discusses on some related issues. Finally, Section 6 concludes.

2. Multi-Instance Learning

Most drugs are small molecules working by binding to larger protein molecules such as enzymes and cell-surface receptors. For molecules qualified to make a drug, one of its low-energy shapes can tightly bind to the target area; while for molecules unqualified to make a drug, none of its low-energy shapes can tightly bind to the target area. In the middle of 1990s, Dietterich et al. (1997) investigated 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, through analyzing a collection of known molecules. The main difficulty of this task lies in that each molecule can 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. In order to solve this problem, Dietterich et al. (1997) regarded each molecule as a bag, and regarded the alternative low-energy shapes of the molecule as the instances in the bag, thereby formulated multi-instance learning. They then proposed three *axis-parallel rectangle* (abbreviated as APR) algorithms to solve the drug activity prediction problem, which attempt to search for appropriate axis-parallel rectangles constructed by the conjunction of the features.

Long and Tan (1998) 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. (1998) 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 (Auer, 1997). Blum and Kalai (1998) 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 the algorithm of Auer et al. (1998).

Maron and Lozano-Pérez (1998) proposed a practical multi-instance learning algorithm, Diverse Density. This algorithm attempts to search for a point in the feature space with the maximum diverse density, where the *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. Many other practical multi-instance learning algorithms have been developed during the past years, such as Wang and Zucker (2000)'s Citation-*k*NN and Bayesian-*k*NN, Ruffo (2000)'s multi-instance decision tree RELIC, Chevaleyre and Zucker (2001)'s multi-instance decision tree ID3-MI and multi-instance rule inducer RIPPER-MI, Zhou and Zhang (2002)'s multi-instance neural network BP-MIP, Zhang and Goldman (2002)'s EM-DD, Gärtner et al. (2002)'s MI Kernel, Andrews et al. (2003)'s MI SVM, Zhou and Zhang (2003)'s multi-instance ensemble, and Xu and Frank (2004)'s MIBOOST. It is noteworthy that almost all these algorithms attempt to adapt single-instance supervised learning algorithms to the multi-instance representation, through shifting their focuses from the discrimination on the instances to the discrimination on the bags (Zhou and Zhang, 2003). Nevertheless, multi-instance learning has already been applied to diverse applications including content-based image retrieval (Yang and Lozano-Pérez, 2000; Zhang et al., 2002; Huang et al., 2002; Zhou et al., 2003), scene classification (Maron and Ratan, 1998; Chen and Wang, 2004), stock selection (Maron and Lozano-Pérez, 1998), landmark matching (Goldman et al., 2001; Goldman and Scott, 2003), computer security (Ruffo, 2000), web mining (Zhou et al., 2005), etc.

In the early years of the research of multi-instance learning, most work were on multi-instance classification with discrete-valued outputs. Later, multi-instance regression with real-valued outputs was studied (Amar et al., 2001; Ray and Page, 2001). It is worth noting that multi-instance learning has also attracted the attention of the ILP community. It has been suggested that multi-instance problems could be regarded as a bias on inductive logic programming, 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 (De Raedt, 1998). Recently, Alphonse and Matwin (2004) 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. This work confirms that multi-instance learning can really act as a bridge between propositional and relational learning.

Recently, Weidmann et al. (2003) indicated that through employing different assumptions of how the instances' classifications determine their bag's label, different kinds of multi-instance problems can be defined. Formally, let χ denote the instance space and $\Omega = \{+, -\}$ denote the set of class labels. A multi-instance concept is a function on $2^\chi \rightarrow \Omega$. In standard multi-instance learning, this function is defined as Eq. 1, where $c_i \in \mathcal{C}$ is a specific concept from a concept space $\mathcal{C}$, and $X \subseteq \chi$ is a set of instances.

$$\nu_{MI}(X) \Leftrightarrow \exists x \in X : c_i(x) \tag{1}$$

Based on this recognition, Weidmann et al. (2003) defined three kinds of generalized multi-instance problems, i.e. *presence-based MI*[1], *threshold-based MI*, and *count-based MI*. Presence-based MI is defined in terms of the presence of

[1] Here "multi-instance" is abbreviated as MI.

instances of each concept in a bag. For example, an MI concept of this category is "only if instances of concept c_1 and instances of concept c_2 are present in the bag, the class is positive". Threshold-based MI requires a certain number of instances of each concept to be present simultaneously. For example, an MI concept of this category is "only if more than n_{c_1} number of instances of concept c_1 and n_{c_2} number of instances of concept c_2 are present in the bag, the class is positive". Count-based MI requires a maximum as well as a minimum number of instances of a certain concept in a bag. For example, an MI concept of this category is "only if at most max_{c_1} and at least min_{c_1} number of instances of concept c_1 and at most max_{c_2} and at least min_{c_2} number of instances of concept c_2 are present in the bag, the class is positive". The formal definitions of presence-based MI, threshold-based MI, and count-based MI are shown in Eqs. 2 to 4.

$$\nu_{PB}(X) \Leftrightarrow \forall c_i \in C : \Delta(X, c_i) \geq 1 \tag{2}$$

$$\nu_{TB}(X) \Leftrightarrow \forall c_i \in C : \Delta(X, c_i) \geq t_i \tag{3}$$

$$\nu_{CB}(X) \Leftrightarrow \forall c_i \in C : t_i \leq \Delta(X, c_i) \leq z_i \tag{4}$$

In Eqs. 2 to 4, ν_{PB}, ν_{TB} and ν_{CB} are functions defined on $2^{\mathcal{X}} \rightarrow \Omega$, $C \subset \mathcal{C}$ is a given set of concepts, Δ is a counting function $\Delta : 2^{\mathcal{X}} \times \mathcal{C} \rightarrow \mathrm{N}$ which counts the number of a given concept in a bag, $t_i \in \mathrm{N}$ and $z_i \in \mathrm{N}$ are respectively the lower and upper threshold for concept c_i.

It is worth mentioning that besides the presence-based MI, threshold-based MI, and count-based MI defined by Weidmann et al. (2003), there is another setting of generalized multi-instance learning, which was defined by Scott et al. (2003). In this setting, the target concept is a set of points $C = \{c_1, \cdots, c_k\}$, and the label for a bag $B = \{b_1, \cdots, b_n\}$ is positive if and only if there is a subset of r target points $C^{'} = \{c_1^{'}, \cdots, c_r^{'}\} \subseteq C$ such that each $c_j^{'} \in C^{'}$ is near some point in B. It is evident that this setting is close to that of threshold-based MI. Scott et al. proposed the Gmil-1 algorithm to solve this problem, which was then reformulated as a kernel algorithm (Tao et al., 2004), reducing the time complexity from exponential to polynomial. Later, this kernel was further generalized along the line of count-based MI (Tao et al., 2004a). This paper only considers Weidmann et al. (2003)'s settings of generalized multi-instance learning since it seems they are more general.

3. Cce

As mentioned before, since in multi-instance learning each bag is represented by a set of feature vectors, common supervised learning algorithms can hardly be applied directly to obtain good performance. Actually, most current multi-instance learning algorithms were derived in nature through enabling single-instance supervised learning algorithms deal with objects described by feature vector sets instead of a single feature vector, which goes the way of shifting the focuses of the algorithms from the discrimination on the instances to the discrimination on the bags (Zhou and Zhang, 2003). As such a strategy of adapting single-instance learning algorithms to meet the multi-instance representation has obtained some success, an opposite strategy, i.e. adapting the multi-instance representation to meet the requirement of existing single-instance supervised learning algorithms, can also be considered. This is really the start point of Cce.

In Cce, the instances contained in all the bags are collected together at first. Since the labels of the instances are unknown, a clustering algorithm is employed to cluster the instances into d groups. Intuitively, since clustering can help find the inherent structure of a data set, the clustered d groups might implicitly encode some information on the distribution of the instances of different bags. Therefore, Cce tries to re-represent the bags based on the clustering results. In detail, d features are generated in the way that if a bag has instance in the i-th group, then the value of the i-th feature is set to 1 and 0 otherwise. Thus, each bag is represented by a d-dimensional binary feature vector such that common single-instance supervised classifiers can be employed to distinguish the bags.

It is evident that various clustering results can be generated for a specific set of instances. Since there is no criterion available for judging which kind of clustering result is the best for the re-representation of the bags, a possible solution is to produce many classifiers based on different clustering results and then combine their predictions, which is adopted by Cce. Note that this is not a disadvantage but an advantage, because in this way, Cce can utilize the power of ensemble learning (Dietterich, 2000) to achieve strong generalization ability.

In general, for obtaining a good ensemble, the component learners should be as diverse as possible. In Cce, diverse classifiers can be easily obtained because they can be trained in different instance spaces. In fact, the clustering process in Cce can be repeated many times, each time the instances are clustered into different numbers of groups. These clustering results are then used to help represent the bags as binary feature vectors with different dimensions. Therefore, different classifiers can be trained with different dimensional feature vectors. This can be viewed as a specific process of manipulating the input features, which has been identified as an effective paradigm for generating diverse classifiers (Dietterich, 2000).

When an unseen bag is given for classification, Cce re-represents it through querying the clustering results, then feeds the generated feature vectors to their corresponding component classifiers, and finally obtains the classification from the ensemble. This implies that the clustering results, at least the center of each clustered group, should be stored such that the instances of the unseen bag can be assigned to appropriate groups through measuring their distances to different centers. Note that although Cce is not so efficient as multi-instance learning algorithms which do not query on training instances in prediction, such as Relic (Ruffo, 2000), it is more efficient than algorithms which require storing and querying all the training instances in prediction, such as Citation-kNN and Bayesian-kNN (Wang and Zucker, 2000).

In the definition of standard as well as generalized multi-instance learning, the label of a bag is actually determined by the relationship between the feature vector set describing the bag and the target points in the instance space. In Cce, such a relationship is implicitly encoded in the single binary feature vector describing the bag. For example, assume that the task is to learn the problem "only if instances of concept c_1 and instances of concept c_2 are present in the bag, the class is positive", and assume that the instances have been clustered into a number of groups where several groups belong to concept c_1 and several belong to concept c_2. Then, if the single binary feature vector of an unseen bag takes value 1 at a bit corresponding to any group belonging to concept c_1 as well as a bit corresponding to any group belonging to concept c_2, this unseen bag is positive because instances of concept c_1 and instances of concept c_2 are present in the bag. Therefore, it is evident that Cce can be applied to generalized multi-

Table 1. Pseudo-code describing the Cce algorithm

```
Cce($\mathcal{B}$, $\mathcal{D}$, $Cluster$, $Classifier$)

    Input: $\mathcal{B}$: A set of $l$ bags $\{X^1, X^2, \ldots, X^l\}$
           $\mathcal{D}$: A set of $m$ numbers $\{d_1, d_2, \ldots, d_m\}$
           $Cluster$: Clustering algorithm
           $Classifier$: Classifier training algorithm

    $Z \leftarrow \emptyset$
    for $i \in \{1..l\}$ do
        for $x \in X^i$ do $Z \leftarrow Z \cup \{x\}$
    end of for

    for $i \in \{1..m\}$ do
        $Cluster(Z, d_i)$      % cluster $Z$ into $d_i$ groups
        $S_i \leftarrow \emptyset$
        for $j \in \{1..l\}$ do
            for $k \in \{1..d_i\}$ do $y_k^j \leftarrow Overlap\left(X^j, group_k\right)$
                % $y_k^j$ is 1 if $X^j$ has instances in the $k$-th group, and 0 otherwise
            $ylabel^j \leftarrow Getlabel(X^j)$
            $S_i \leftarrow S_i \cup \{< y_1^j, \ldots, y_{d_i}^j, ylabel^j >\}$
        end of for
        $C_i \leftarrow Classifier(S_i)$
    end of for

    Output: $Label\,(X) \leftarrow \arg\max_{t \in \{+,-\}} \sum_{i:\, C_i\left(\widehat{X}_i\right)=t} 1$
                % $\widehat{X}_i$ is the corresponding feature vector of the bag $X$ for $C_i$
```

instance problems without any modification, which is a prominent advantage, while most current multi-instance learning algorithms cannot.

The pseudo-code of Cce is presented in Table 1. Many algorithms (Hinneburg and Keim, 2003; Ordonez and Omiecinski, 2004; Zhou et al., 2000; Abbass et al., 2001; Hodge and Austin, 2005) can be used to implement the clustering process and the classifier. In this paper, k-means is employed for clustering, while support vector machines are used as the classifiers. In combining the predictions of the classifiers, *majority voting* is used in this paper, but note that other schemes are also applicable.

4. Experiments

4.1. Musk Data Sets

Musk data is a real-world benchmark test data for standard multi-instance learning algorithms, which was generated in the research of drug activity prediction (Dietterich et al., 1997). There are two data sets, i.e. *Musk1* and *Musk2*, both publicly available at the UCI machine learning repository (Blake et al., 1998). *Musk1* contains 47 positive bags and 45 negative bags, and the number of instances contained in each bag ranges from 2 to 40. *Musk2* contains 39 positive bags and 63 negative bags, and the number of instances contained in each bag ranges from 1 to 1,044. Each instance in the bags is represented by 166 continuous attributes. Detailed information on the *Musk* data is tabulated in Table 2.

Table 2. The *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.
Musk1	166	92	47	45	476	2	40	5.17
Musk2	166	102	39	63	6,598	1	1,044	64.69

Leave-one-out test is performed on the *Musk* data sets. Each time Cce clusters the instances into five different numbers of groups, and then five classifiers are trained and combined. The best predictive error rates of Cce are compared with the best results reported in the literatures, as shown in Table 3. Note that these results were obtained with different experimental methodologies. For example, the results of MI Kernel [2] were average values of 1,000 runs each randomly leaves out 10 bags for testing while using the remaining bags to train the classifier (Gärtner et al., 2002), the results of the Tlc algorithms [3] were obtained with 10 runs of 10-fold cross-validation (Weidmann et al., 2003), while the results of Cce were obtained with leave-one-out test.

Table 3 shows that on *Musk1* Cce ranks the 3rd among all the 21 algorithms while on *Musk2* it ranks the 5th. Note that among all these algorithms, besides Cce, only MI Kernel and Tlc without AS can be used to tackle generalized multi-instance problems (Weidmann et al., 2003). It can be found from Table 3 that the performance of Cce is comparable to that of MI Kernel on *Musk1* but worse on *Musk2* [4], while much better than that of Tlc without AS on both *Musk1* and *Musk2*. These observations support the claim that Cce illustrates a new way to solve multi-instance problems, that is, adapting the multi-instance representation to common single-instance supervised learning algorithms.

Fig. 1 shows the performance of the single classifiers trained after the clustering process employed by Cce, where the number of clusters ranges from 2 to 80. Fig. 2 shows the best performance of the classifier ensembles generated by Cce, where the number of classifiers used ranges from 3 to 21 with interval 2.

Comparing Figs. 1 and 2, it can be found that the ensembles are strong although the single classifiers are not strong. Actually, the best predictive accuracy of the single classifier is 85.9% on *Musk1* and 80.4% on *Musk2*, both worse than that of Diverse Density (88.9% on *Musk1* and 82.5% on *Musk2*); but the best predictive accuracy of the Cce ensemble is 92.4% on *Musk1* and 87.3% on *Musk2*, both much better than that of Diverse Density. Moreover, Fig. 2 reveals that no matter which ensemble size is used, the performance of Cce ensemble is always higher than 89.1% on *Musk1* and 83.3% on *Musk2*, consistently better than that of Diverse Density. These observations tell that the performance of the single classifier is relatively sensitive to the clustering process, while that of the Cce ensemble is relatively robust owing to the contribution of the ensemble process.

[2] This method was called as "MI Svm" by Weidmann et al. (2003), but originally it was named as MI Kernel (Gärtner et al., 2002) and MI Svm is usually used to refer to Andrews et al. (2003)'s method.

[3] "Tlc with/without AS" means Tlc with/without attribute selection. The performance of Tlc with AS on *Musk* is not available (Weidmann et al., 2003).

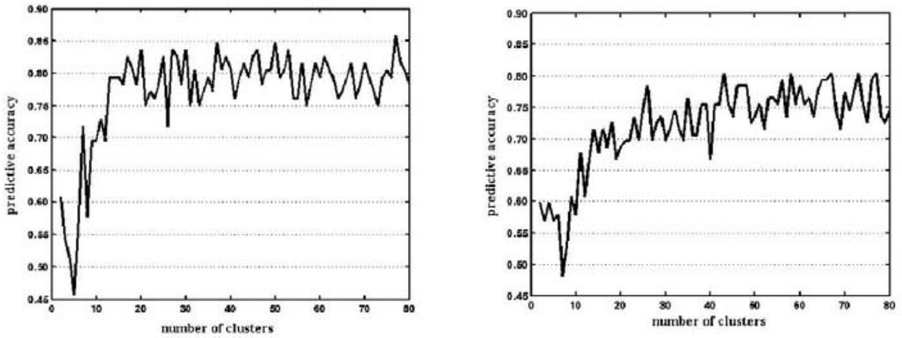
[4] The best predictive error rates of MI Kernel reported by Weidmann et al. (2003) were 13.6% on *Musk1* and 12.0% on *Musk2*. The latter is slightly better while the former is much worse than that of Cce.

Table 3. The best predictive error rates (%) on *Musk* data sets

Musk1	
Algorithm	Error
MI Ensemble	3.1 (Zhou and Zhang, 2003)
Em-dd	3.2 (Zhang and Goldman, 2002)
Cce	7.6
Iterated-discrim Apr	7.6 (Dietterich et al., 1997)
Citation-*k*NN	7.6 (Wang and Zucker, 2000)
MI Kernel	7.6 (Gärtner et al., 2002)
Gfs elim-kde Apr	8.7 (Dietterich et al., 1997)
Gfs elim-count Apr	9.8 (Dietterich et al., 1997)
Bayesian-*k*NN	9.8 (Wang and Zucker, 2000)
Diverse Density	11.1 (Maron and Lozano-Pérez, 1998)
Tlc without AS	11.3 (Weidmann et al., 2003)
Ripper-mi	12.0 (Chevaleyre and Zucker, 2001)
Bp-mip-pca	12.0 (Zhang and Zhou, 2004)
MiBoost	12.1 (Xu and Frank, 2004)
Bp-mip-dd	14.1 (Zhang and Zhou, 2004)
Relic	16.3 (Ruffo, 2000)
Bp-mip	16.3 (Zhou and Zhang, 2002)
MI Svm	22.1 (Andrews et al., 2003)
Multinst	23.3 (Auer, 1997)
BP	25.0 (Dietterich et al., 1997)
C4.5	31.5 (Dietterich et al., 1997)
Musk2	
Algorithm	Error
MI Ensemble	3.0 (Zhou and Zhang, 2003)
Em-dd	4.0 (Zhang and Goldman, 2002)
MI Kernel	7.8 (Gärtner et al., 2002)
Iterated-discrim Apr	10.8 (Dietterich et al., 1997)
Cce	12.7
Relic	12.7 (Ruffo, 2000)
Citation-*k*NN	13.7 (Wang and Zucker, 2000)
MI Svm	15.7 (Andrews et al., 2003)
Multinst	16.0 (Auer, 1997)
MiBoost	16.0 (Xu and Frank, 2004)
Bp-mip-pca	16.7 (Zhang and Zhou, 2004)
Tlc without AS	16.9 (Weidmann et al., 2003)
Diverse Density	17.5 (Maron and Lozano-Pérez, 1998)
Bayesian-*k*NN	17.6 (Wang and Zucker, 2000)
Gfs elim-kde Apr	19.6 (Dietterich et al., 1997)
Bp-mip	19.6 (Zhou and Zhang, 2002)
Bp-mip-dd	19.6 (Zhang and Zhou, 2004)
Ripper-mi	23.0 (Chevaleyre and Zucker, 2001)
Gfs elim-count Apr	24.5 (Dietterich et al., 1997)
BP	32.3 (Dietterich et al., 1997)
C4.5	41.2 (Dietterich et al., 1997)

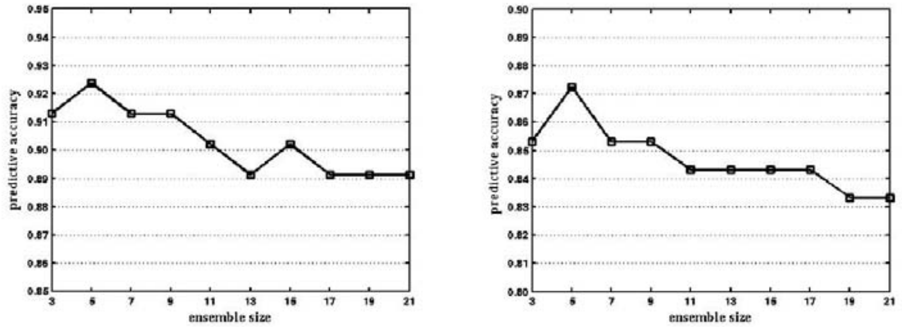
4.2. Generalized MI Data Sets

Weidmann et al. (2003) designed some methods for artificially generating generalized multi-instance data sets. In every data set they generated, there are five different training sets each containing 50 positive and 50 negative bags, and a big test set containing 5,000 positive and 5,000 negative bags. The average test set accuracy of the classifiers trained on each of these five training sets is recorded



(a) *Musk1* (b) *Musk2*

Fig. 1. Predictive accuracy of single classifiers trained from different number of clusters



(a) *Musk1* (b) *Musk2*

Fig. 2. The best predictive accuracy of CCE ensembles with different ensemble sizes

as the predictive accuracy of the concerned learning algorithm on that data set. Note that in this subsection, the results of MI Kernel and the TLC algorithms are from the literature (Weidmann et al., 2003).

In generating presence-based MI data sets, $|C|$ concepts were used. To generate a positive bag, the number of instances in a concept was chosen randomly from $\{1, \ldots, 10\}$ for each concept. The number of random instances was selected with equal probability from $\{10|C|, \ldots, 10|C| + 10\}$. Hence the minimal bag size in this data set was $|C| + 10|C|$ and the maximal bag size $20|C| + 10$. In this paper, four presence-based MI data sets are used. The results are shown in Table 4, where the numbers following '±' are standard deviations. Here the name of the data set '2-10-5' means this data set was generated with 2 concepts, 10 relevant and 5 irrelevant attributes.

In generating threshold-based MI data sets, Weidmann et al. (2003) also used $|C|$ concepts. They chose thresholds $t_1 = 4$ and $t_2 = 2$ for data sets with $|C| = 2$, and $t_1 = 2$, $t_2 = 7$ and $t_3 = 5$ for data sets with $|C| = 3$. For positive bags, the number of instances of concept c_i was chosen randomly from $\{t_i, \ldots, 10\}$. To form a negative bag, they replaced at least $(\Delta(X, c_i) - t_i + 1)$ instances of a concept c_i in a positive bag X by random instances. The minimal bag size in this data set is $\sum_i t_i + 10|C|$, the maximal size is $20|C| + 10$. In this paper,

Table 4. Predictive accuracy (%) on presence-based MI data sets

Data set	MI Kernel	Tlc without AS	Tlc with AS	Cce
2-10-5	82.00 ± 1.53	85.18 ± 10.07	96.67 ± 1.58	88.95 ± 0.31
3-5-5	82.12 ± 0.98	81.93 ± 2.90	99.98 ± 0.04	82.86 ± 0.46
3-5-10	81.43 ± 0.96	86.32 ± 6.48	98.49 ± 1.74	78.88 ± 1.03
3-10-5	84.27 ± 1.44	78.07 ± 0.91	87.41 ± 6.24	84.93 ± 0.90

Table 5. Predictive accuracy (%) on threshold-based MI data sets

Data set	MI Kernel	Tlc without AS	Tlc with AS	Cce
42-10-5	85.36 ± 0.92	88.65 ± 10.12	96.58 ± 1.91	87.64 ± 1.34
42-10-10	83.93 ± 0.36	84.59 ± 8.08	95.89 ± 2.05	88.04 ± 0.38
275-5-10	82.73 ± 0.85	86.42 ± 5.39	93.75 ± 6.63	79.59 ± 0.26
275-10-5	87.05 ± 0.75	86.92 ± 6.56	90.44 ± 4.63	84.39 ± 0.65

four threshold-based MI data sets are used. The results are shown in Table 5, where the numbers following '±' are standard deviations. Here the name of the data set '42-10-5' means this data set has at least 4 instances of the first concept and 2 instances of the second concept in a positive bag, with 10 relevant and 5 irrelevant attributes.

In generating count-based MI data sets, Weidmann et al. (2003) still used $|C|$ concepts. They used the same value for both thresholds t_i and z_i. Hence, the number of instances of concept c_i is exactly z_i in a positive bag. They set $z_1 = 4$ and $z_2 = 4$ for data sets with $|C| = 2$, and $z_1 = 2$, $z_2 = 7$ and $z_3 = 5$ for data sets with $|C| = 3$. A negative bag can be created by either increasing or decreasing the required number z_i of instances for a particular c_i. They chose a new number from $\{0, \ldots, z_i - 1\} \cup \{z_i + 1, \ldots, 10\}$ with equal probability. If this number was less than z_i, they replaced instances of concept c_i by random instances; if it was greater, they replaced random instances by instances of concept c_i. The minimal bag size in this data set is $\sum_i z_i + 10|C|$, and the maximal possible bag size is $\sum_i z_i + 10|C| + 10$. In this paper, four count-based MI data sets are used. The results are shown in Table 6, where the numbers following '±' are standard deviations. Here the name of the data set '42-10-0' means this data set requires exactly 4 instances of the first concept and 2 instances of the second concept in a positive bag, with 10 relevant and 0 irrelevant attributes.

Tables 4 to 6 show that although the performance of Cce is not so good as that of Tlc with AS, it is comparable to that of MI Kernel and Tlc without AS. The fact that the performance of Cce on count-based MI is not well might because the binary feature vectors used by Cce are not sufficient for representing the exact number of instances in a cluster. Note that the Tlc methods were

Table 6. Predictive accuracy (%) on count-based MI data sets

Data set	MI Kernel	Tlc without AS	Tlc with AS	Cce
42-10-0	55.21 ± 1.76	90.89 ± 6.25	92.76 ± 1.64	57.69 ± 1.69
42-10-10	55.59 ± 2.81	51.05 ± 1.60	65.10 ± 20.35	57.79 ± 1.72
275-5-10	52.34 ± 0.50	50.33 ± 0.72	56.94 ± 11.64	52.74 ± 0.81
275-10-0	54.52 ± 1.54	87.85 ± 4.26	89.86 ± 3.40	53.39 ± 1.24

specifically designed for generalized multi-instance problems. Therefore, the experimental results tell that CCE can work well, at least not bad, on generalized multi-instance problems.

5. Discussion

Weidmann et al. (2003) proposed TLC to tackle generalized multi-instance problems, which constructs a *meta-instance* for each bag and then passes the meta-instance and the class label of the corresponding bag together to a common classifier. TLC uses a standard decision tree for imposing a structure on the instance space, which is trained on the set of all instances contained in all bags where the instances are labeled with their bag's class label, such that a meta-instance is generated for each bag. It is obvious that the role of the decision tree in TLC can be taken place by some other supervised learning algorithms such as rule induction algorithms. Nevertheless, it is worth noting that TLC generates only one meta-instance for each bag. In contrast to TLC, CCE employs clustering to impose different structures on the instance space, in each structure a meta-instance can be generated for each bag. That is, CCE generates multiple meta-instances for each bag, therefore it can utilize the power of ensemble learning in making predictions with an ensemble instead of a single classifier.

Zhou and Zhang (2003) proposed to build multi-instance ensembles to solve multi-instance problems. In detail, they used a popular ensemble learning method to generate ensembles of multi-instance learners including Iterated-discrim APR, Diverse Density, Citation-kNN, and EM-DD, and obtained better results than single learners. In contrast to CCE, the multi-instance ensemble method does not change the representation of the bags. Moreover, since the base learners it employed were designed for standard multi-instance problems, these multi-instance ensembles could only be applied to standard multi-instance problems.

Data with complex structures are usually difficult to learn with traditional machine learning paradigms. Although the structure of multi-instance data is not so complex as that of multimedia data, it is really more complex than traditionally used feature vector structure. In order to learn multi-instance data, a usually adopted way is to modify common learning algorithms to meet complex representations (Zhou and Zhang, 2003), while CCE goes an opposite way, i.e. simplifying complex representations to meet common learning algorithms. In fact, such an idea of changing the representation has been studied in the area of constructive induction as early as (Michalski, 1983).

Constructive induction is a general approach for dealing with inadequate features found in original data. Commonly, it improves the representation by constructing new features from the instance space, so that the learning tasks become easier to be performed or the learning results are improved. Roughly speaking, there are three kinds of constructive induction schemes classified according to the information used in searching for the best representation space (Bloedorn and Michalski, 1998), that is, data-driven constructive induction that exploits input examples, hypothesis-driven constructive induction that exploits intermediate hypotheses, and knowledge-driven constructive induction that exploits domain knowledge. The clustering process employed by CCE is actually used to help construct new features from the original instances, which can be viewed as a data-driven constructive induction process. Therefore, the success of CCE also indicates that, although constructive induction is not so hot as before, this technique might be well proven useful in learning data with complex structures.

6. Conclusion

Most current multi-instance learning algorithms were derived from supervised learning algorithms through shifting their focuses from the discrimination on the instances to the discrimination on the bags, that is, adapting single-instance algorithms to the multi-instance representation. The main contribution of this paper is the illustration of the feasibility of an opposite way to the solution of multi-instance learning, that is, adapting the multi-instance representation to the single-instance algorithms.

The Cce method proposed in this paper employs a clustering process to help construct new features, on which common supervised learning algorithms can work. Besides, Cce utilizes the power of ensemble learning paradigms to achieve strong generalization ability. Experiments show that Cce can work well on standard multi-instance problems. Moreover, experiments show that Cce can be applied to generalized multi-instance problems without any modification, which is difficult for most current multi-instance learning algorithms.

There are many possible ways for modifying the multi-instance representation. For example, the number of instances of a bag belonging to the clustered groups can be used as feature values such that each bag is represented by an integer instead of a binary feature vector. Exploring other schemes for adapting multi-instance representation to single-instance algorithms is an interesting issue for future work.

Moreover, the success of Cce discloses that in learning data with complex structures, constructive induction techniques might be useful. Trying to apply these techniques to tasks involving complex structures of data, such as multimedia stream data, is also an interesting issue to be explored in the future.

Acknowledgements. The authors wish to thank Eibe Frank for constructive discussions. The comments and suggestions from the anonymous reviewers significantly improved the paper. This work was supported by the National Science Fund for Distinguished Young Scholars of China under the Grant No. 60325207, the National Science Foundation of China under the Grant No. 60496327, the Fok Ying Tung Education Foundation under the Grant No. 91067, the Excellent Young Teachers Program of MOE of China, and the National 973 Fundamental Research Program of China under the Grant No. 2002CB312002.

References

Abbass HA, Towsey M, Finn G (2001) C-Net: A method for generating non-deterministic and dynamic multivariate decision trees. Knowledge and Information Systems 3(2):184–197

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

Amar RA, Dooly DR, Goldman SA, Zhang Q (2001) Multiple-instance learning of real-valued data. In Proceedings of the 18th International Conference on Machine Learning, Williamstown, MA, 2001, pp 3–10

Andrews S, Tsochantaridis I, Hofmann T (2003) Support vector machines for multiple-instance learning. In Becker S, Thrun S, Obermayer K (eds). Advances in Neural Information Processing Systems 15, MIT Press, Cambridge, MA, pp 561–568

Auer P (1997) On learning from multi-instance examples: Empirical evaluation of a theoretical approach. In Proceedings of the 14th International Conference on Machine Learning, Nashville, TN, 1997, pp 21–29

Auer P, Long PM, Srinivasan A (1998) Approximating hyper-rectangles: Learning and pseudorandom sets. Journal of Computer and System Sciences 57(3):376–388

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

Bloedorn E, Michalski RS (1998) Data-driven constructive induction. IEEE Intelligent Systems 13(2):30–37

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

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

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

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

Dietterich TG (2000) Ensemble methods in machine learning. In Kittler J, Roli F (eds). Lecture Notes in Computer Science 1867, Springer, Berlin, pp 1–15

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

Gärtner T, Flach PA, Kowalczyk A, Smola AJ (2002) Multi-instance kernels. In Proceedings of the 19th International Conference on Machine Learning, Sydney, Australia, 2002, pp 179–186

Goldman SA, Kwek SS, Scott SD (2001) Agnostic learning of geometric patterns. Journal of Computer and System Sciences 62(1):123–151

Goldman SA, Scott SD (2003) Multiple-instance learning of real-valued geometric patterns. Annals of Mathematics and Artificial Intelligence 39(3):259–290

Hinneburg A, Keim DA (2003) A general approach to clustering in large databases with noise. Knowledge and Information Systems 5(4):387–415

Hodge VJ, Austin J (2005) A binary neural k-nearest neighbour technique. Knowledge and Information Systems 8(3):276–309

Huang X, Chen S-C, Shyu M-L, Zhang C (2002) Mining high-level user concepts with multiple instance learning and relevance feedback for content-based image retrieval. In Zaïane OR, Simoff SJ, Djeraba C (eds). Lecture Notes in Artificial Intelligence 2797, Springer, Berlin, pp 50–67

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

Maron O (1998) Learning from ambiguity. PhD dissertation, Department of Electrical Engineering and Computer Science, MIT, Jun. 1998

Maron O, Lozano-Pérez T (1998) A framework for multiple-instance learning. In Jordan MI, Kearns MJ, Solla SA (eds). Advances in Neural Information Processing Systems 10, MIT Press, Cambridge, MA, pp 570–576

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

Michalski RS (1983) A theory and methodology of inductive learning. In Michalski RS, Carbonell JG, Mitchell TM (eds). Machine Learning: An Artificial Intelligence Approach, Tioga, Palo Alto, CA, pp 83–134

Ordonez C, Omiecinski E (2004) Accelarating EM clustering to find high-quality solutions. Knowledge and Information Systems 7(2):135–157

Ray S, Page D (2001) Multiple instance regression. In Proceedings of the 18th International Conference on Machine Learning, Williamstown, MA, 2001, pp 425–432

Ruffo G (2000) Learning single and multiple instance decision trees for computer security applications. PhD dissertation, Department of Computer Science, University of Turin, Torino, Italy, 2000

Scott SD, Zhang J, Brown J (2003) On generalized multiple-instance learning. Technical Report UNL-CSE-2003-5, Department of Computer Science, University of Nebraska, Lincoln, NE, 2003

Tao Q, Scott S, Vinodchandran NV, Osugi TT (2004) SVM-based generalized multiple-instance learning via approximate box counting. In Proceedings of the 21st International Conference on Machine Learning, Banff, Canada, 2004, pp 779–806

Tao Q, Scott S, Vinodchandran NV, Osugi TT, Mueller B (2004) An extended kernel for generalized multiple-instance learning. In Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, Boca Raton, FL, 2004, pp 272–277

Wang J, Zucker J-D (2000) Solving the multiple-instance problem: A lazy learning approach. In Proceedings of the 17th International Conference on Machine Learning, San Francisco, CA, 2000, pp 1119–1125

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

Xu X, Frank E (2004) Logistic regression and boosting for labeled bags of instances. In Dai H, Srikant R, Zhang C (eds). Lecture Notes in Artificial Intelligence 3056, Springer, Berlin, pp 272–281

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

Zhang Q, Goldman SA (2002) EM-DD: An improved multi-instance learning technique. In Dietterich TG, Becker S, Ghahramani Z (eds). Advances in Neural Information Processing Systems 14, MIT Press, Cambridge, MA, pp 1073–1080

Zhang Q, Yu W, Goldman SA, Fritts JE (2002) Content-based image retrieval using multiple-instance learning. In Proceedings of the 19th International Conference on Machine Learning, Sydney, Australia, 2002, pp 682–689

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

Zhou Z-H, Chen S, Chen Z (2000) FANNC: A fast adaptive neural network classifier. Knowledge and Information Systems 2(1):115–129

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

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

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

Zhou Z-H, Zhang M-L, Chen K-J (2003) A novel bag generator for image database retrieval with multi-instance learning techniques. In Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, Sacramento, CA, 2003, pp 565–569

Author Biographies

Zhi-Hua Zhou is currently Professor in the Department of Computer Science & Technology and head of the LAMDA group at Nanjing University. His main research interests include machine learning, data mining, information retrieval, and pattern recognition. He is associate editor of *Knowledge and Information Systems* and on the editorial boards of *Artificial Intelligence in Medicine*, *International Journal of Data Warehousing and Mining*, *Journal of Computer Science & Technology*, and *Journal of Software*. He has also been involved in various conferences.

Min-Ling Zhang received his B.Sc. and M.Sc. degrees in computer science from Nanjing University, China, in 2001 and 2004, respectively. Currently he is a Ph.D. candidate in the Department of Computer Science & Technology at Nanjing University and a member of the LAMDA group. His main research interests include machine learning and data mining, especially in multi-instance learning and multi-label learning.

Correspondence and offprint requests to: Zhi-Hua Zhou, National Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China. Email: zhouzh@nju.edu.cn