Multi-Instance Learning: A Survey

Zhi-Hua Zhou

National Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210093, China

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 provides a survey on this topic. At first, it introduces the origin of multi-instance learning. Then, developments on the study of learnability, learning algorithms, applications and extensions of multi-instance learning are reviewed. In particular, this paper employs a unified view to look into multi-instance learning algorithms. Some important issues to be addressed are also discussed.

Keywords Learning from Examples; Multi-Instance Learning; Supervised Learning

Corresponding author:
Zhi-Hua Zhou
Mail: National Laboratory for Novel Software Technology,
Nanjing University, Mailbox 419,
Hankou Road 22, Nanjing 210093, China
Tel.: +86-25-8368-6268
Fax: +86-25-8368-6268
E-mail: zhouzh@nju.edu.cn

Multi-Instance Learning: A Survey

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 provides a survey on this topic. At first, it introduces the origin of multi-instance learning. Then, developments on the study of learnability, learning algorithms, applications and extensions of multi-instance learning are reviewed. In particular, this paper employs a unified view to look into multi-instance learning algorithms. Some important issues to be addressed are also discussed.

Key words: Learning from Examples; Multi-Instance Learning; Supervised Learning

1 Introduction

During the past years, *learning from examples* is one of the most flourishing areas in machine learning. According to the *ambiguity* of training data, research in this area can be roughly categorized into three learning frameworks, i.e. supervised learning, unsupervised learning, and reinforcement learning [16]. 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 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 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 could be viewed as delayed labels, therefore the ambiguity is between that of supervised learning and unsupervised learning.

The term *multi-instance learning* was coined by Dietterich et al. [10] 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; otherwise it is labeled as a negative bag. The task is to learn some concept from the training set for correctly labeling unseen bags.

Different to supervised learning where all training instances are with known labels, in multi-instance learning the labels of the training instances are unknown; different to unsupervised learning where all training instances are without known labels, in multi-instance learning the labels of the training bags are known; different 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, could not work well in this scenario [10].

Since multi-instance problems extensively exist but are unique to those addressed by previous learning frameworks, multi-instance learning was regarded as a new learning framework [16], and has attracted much attention of the machine learning community. Since many developments on multi-instance learning have been made, it seems useful to have a survey on this topic, which is the purpose of this paper.

The rest of this paper is organized as follows. Section 2 introduces the origin of multi-instance learning. Section 3 presents the analyses on the learnability of multi-instance learning. Section 4 looks into multi-instance learning algorithms with a supervised view. Section 5 and Section 6 presents applications and extensions of multi-instance learning, respectively. Finally, Section 7 discusses on several issues to be addressed in this field.

2 Drug Activity Prediction

In the middle of 1990s, Dietterich et al. [10] 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.

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 could tightly bind to the target area; while for molecules unqualified to make a drug, none of its low-energy shapes could 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, as illustrated in Fig. 1, 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.

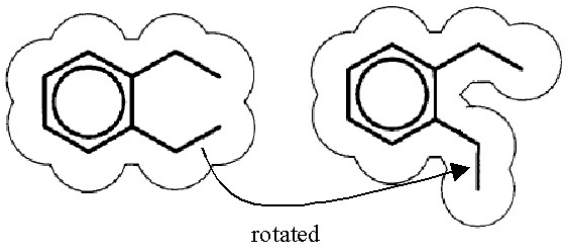


Fig. 1. The shape of a molecule changes as it rotates an internal bond

An intuitive solution is to utilize 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. [10], such a method can hardly work due to high false positive noise, which is caused by 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. [10] 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. In order to represent the shapes, the molecule was placed in a standard position and orientation and then a set of 162 rays emanating from the origin was constructed so that the molecular surface was sampled approximately uniformly, as illustrated in Fig. 2. There were also four features that represented the position of an oxygen atom on the molecular

surface. Therefore each instance in the bags was represented by a 166-dimensional numerical feature vector.

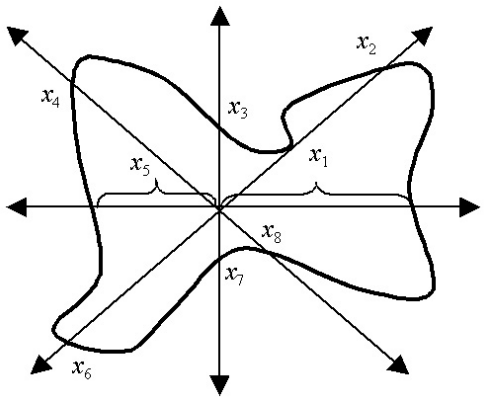


Fig. 2. The ray-based representation of the molecular shape

Dietterich et al. [10] proposed three *axis-parallel rectangle* (abbreviated as APR) algorithms to solve the drug activity prediction problem, which attempts to search for appropriate axis-parallel rectangles constructed by the conjunction of the features. The GFS elim-count APR algorithm identifies an APR covering all the instances from positive bags at first. Then, it gradually shrinks the APR through greedily eliminating instances from negative bags. A figure from Dietterich et al.'s paper [10] well illustrated this process, as shown in Fig. 3, where the white and dark points represented instances from positive and negative bags, respectively, different shapes represented different bags, and the initial APR was indicated with solid line. For each instance from native bags, the algorithm counts the minimum number of instances from positive bags that must be excluded from the APR in order to exclude the concerned instance from negative bag. In Fig. 3, these counts were shown next to small lines that indicated which side of the APR should be shrunk in order to exclude the instance from negative bag. The algorithm iteratively chooses to eliminate the instance from negative bag that is easiest to eliminate until all such instances are eliminated. The resulting APR was indicated with the dash line in Fig. 3. Then, the algorithm determines the bounds of relevant features with greedy feature selection, and therefore obtains the final APR.

The difference between the GFS kde APR algorithm and the GFS elim-count APR algorithm mainly lies in the fact that the former does not merely counting the

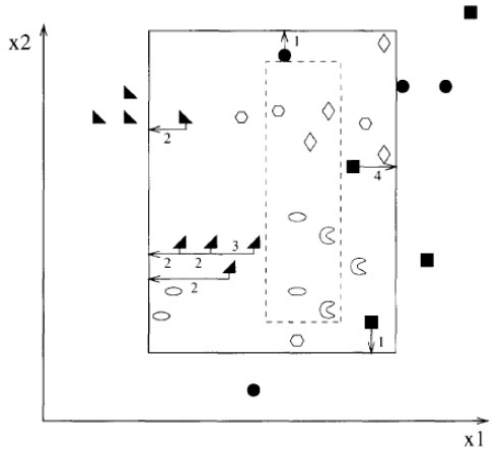


Fig. 3. An illustration of the GFS elim-count APR algorithm

number of instances from positive bags that must be excluded in order to exclude an instance from negative bag. Instead, GFS kde APR considers the number of instances from different positive bags that are covered by the initial APR, and uses a cost function to control the process of eliminating instances from negative bags so that each positive bag reserves at least one instance in the APR. The Iterated-discrim APR algorithm employs greedy backfitting algorithm to identify an APR that covers at least one instance from each positive bag. Then, it utilizes this APR to select the most discriminating features. Finally, kernel density estimation is exploited to help improve the generalization through expanding the bounds of the APR so that with high probability, new instances from positive bags will fall inside the APR.

Dietterich et al.'s experiments [10] showed that the Iterated-discrim APR algorithm achieved the best result on the *Musk* data [6], which is a concrete test data of drug activity prediction and the most popularly used benchmark in multi-instance learning. It is worth mentioning that Dietterich et al. [10] indicated that since the Iterated-discrim APR algorithm had been geared to the *Musk* data, its performance might be the upper bound on this data set.

Note that multi-instance problems do not emerge suddenly from drug activity prediction. In fact, they extensively exist in real-world applications [14][32]. But unfortunately, the uniqueness of these problems have not been particularly distinguished until Dietterich et al. [10].

3 Learnability

Long and Tan [15] 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. [5] 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 [4]. Blum and Kalai [7] 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 [5].

Table 1 summarizes these works, where $(1-\epsilon)$ and δ represent accuracy and confidence, d and n represent the APR dimensionality and the number of instances in bags, respectively. Here $\widetilde{O}$ denotes that logarithm term has been subsumed.

Table 1
Main results on the study of learnability

Work	Sample complexity	Running time	Main assumptions
Long & Tan [15]	$O(\frac{d^2n^6}{\varepsilon^{10}}\log\frac{nd}{\varepsilon\delta})$	$O(\frac{d^5n^{12}}{\varepsilon^{20}}\log^2\frac{nd}{\varepsilon\delta})$	independent instances, product distribution, n is constant
Auer et al. [5]	$O(\frac{d^2n^2}{\varepsilon^2}\log\frac{d}{\delta})$	$\widetilde{O}(\frac{d^3n^2}{\varepsilon^2})$	independent instances, n is constant
Blum & Kalai [7]	$\widetilde{O}(\frac{d^2n}{\varepsilon^2})$	$\widetilde{O}(\frac{d^3n^2}{\varepsilon^2})$	independent instances, n is constant

It is noteworthy that all these results were obtained under strong assumptions, such as the number of instances in the bags should be a constant and the instances

should be independent. Unfortunately, these assumptions are rarely satisfied in real-world problems. For example, it is not the fact that different molecules have the same number of low-energy shapes, and it is not reasonable to assume that different low-energy shapes of the same molecule are fully independent. Moreover, all the theoretical analyses focused on APR learning. Although many effective multi-instance learning algorithms have been proposed during the past years, it seems that these kinds of learning algorithms are difficult to be analyzed under the PAC framework at present. Nonetheless, these works have enriched the research scope of computational learning theory, and disclosed some insights for designing multi-instance learning algorithms. For example, Blum and Kalai [7] have indicated that any learner using statistical queries could be used to learn multi-instance concepts under the i.i.d. assumption.

4 Learning Algorithm

After Dietterich et al. [10], many multi-instance learning algorithms have been developed, mainly including Diverse Density [17], Citation-kNN and Bayesian-kNN [21], Relic [20], ID3-MI and RIPPER-MI [8], EM-DD [25], BP-MIP [29][27], MI kernels [11], MI SVMs [3], and multi-instance ensembles [30]. It seems not wise to give detailed descriptions for all these learning algorithms. Instead, this paper attempts to look into them through a unified view, which may help grasp the essentials of these multi-instance learning algorithms.

Generally speaking, the focus of a supervised learner is to discriminate the instances, which is feasible since all training instances are labeled in supervised scenario. But in multi-instance learning it is very difficult, if not infeasible, to discriminate training instances because none of them is labeled. Moreover, if the label of a bag is simply regarded as the label of its instances, that is, to believe that positive bag contains only positive instances while negative bag contains only negative instances, then the learning task may be very difficult although every training instance holds a label now. This is because the positive noise may be extremely high, as indicated by Dietterich et al. [10]. Therefore, whether it is pos-

sible to discriminate the training instances or not is the principal difference between supervised learning and multi-instance learning.

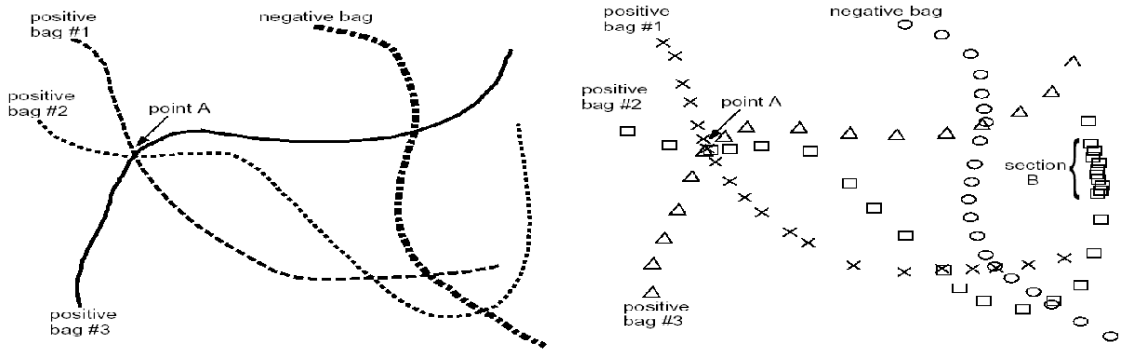
In fact, almost all the present multi-instance learning algorithms were modified from supervised learning algorithms according to a general rule, i.e. shifting the focuses of supervised learning algorithms from the discrimination on instances to the discrimination on bags. This rule has been implicitly utilized by the authors of most multi-instance learning algorithms, who have identified effective ways to substantialize the rule on different kind of supervised learners. Below we employ such a unified supervised view to examine Diverse Density, Citation-kNN, ID3-MI, RIPPER-MI, and BP-MIP.

4.1 Diverse Density

The Diverse Density algorithm [17] 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. A figure from Maron and Lozano-Pérez's paper well illustrated this heuristics, as shown in Fig. 4. 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 how such a definition can be achieved through 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, i.e. $C = \{c_1, c_2, \cdots, c_t\}$, to be predicted, the posterior probability of the class can be estimated according to the Bayes rule



(a) The different shapes that a molecule can take on are represented as a path. The inter-section of positive paths is where they took on the same shape.

(b) Samples taken along the paths. Section B is a high density area, but point A is a high Diverse Density area.

Fig. 4. The heuristics of Diverse Density

as shown in Eq. 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 Eq. 2, where *Obj* denotes the objective.

$$\begin{aligned} \text{Obj} &= \arg\max_{1\le k\le t} \Pr(c_k|D) \\ &= \arg\max_{1\le k\le t} \frac{\Pr(D|c_k)\Pr(c_k)}{\Pr(D)} \end{aligned} \tag{2}$$

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

$$\text{Obj} = \arg\max_{1\le k\le t} \Pr(D|c_k) \tag{3}$$

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

$$\text{Obj} = \arg\max_{1\le k\le t} \Pr\left(\{B_1^+, \cdots, B_m^+, B_1^-, \cdots, B_n^-\}|c_k\right)$$

$$= \arg\max_{1\le k\le t} \prod_{1\le i\le m} \Pr\left(B_i^+ \,|c_k\right) \prod_{1\le j\le n} \Pr\left(B_j^- \,|c_k\right) \tag{4}$$

Now apply the Bayes rule to Eq. 4, we get Eq. 5.

$$\text{Obj} = \arg\max_{1\le k\le t} \prod_{1\le i\le m} \frac{\Pr\left(c_k \left| B_i^+\right.\right) \Pr\left(B_i^+\right)}{\Pr\left(c_k\right)} \prod_{1\le j\le n} \frac{\Pr\left(c_k \left| B_j^-\right.\right) \Pr\left(B_j^-\right)}{\Pr\left(c_k\right)} \tag{5}$$

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

$$\text{Obj} = \arg\max_{1\le k\le t} \prod_{1\le i\le m} \Pr\left(c_k \left| B_i^+\right.\right) \prod_{1\le j\le n} \Pr\left(c_k \left| B_j^-\right.\right) \tag{6}$$

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

$$\text{Obj}^x = \arg\max_{1\le k\le t} \prod_{1\le i\le m} \Pr\left(x = c_k \left| B_i^+\right.\right) \prod_{1\le j\le n} \Pr\left(x = c_k \left| B_j^-\right.\right) \tag{7}$$

If we want to find out 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 Eq. 8.

$$\begin{aligned} \hat{x} &= \arg\max_x \Pr\left(\text{Obj}^x = c_h\right) \\ &= \arg\max_x \prod_{1\le i\le m} \Pr\left(x = c_h \left| B_i^+\right.\right) \prod_{1\le j\le n} \Pr\left(x = c_h \left| B_j^-\right.\right) \end{aligned} \tag{8}$$

Eq. 8 is exactly the formal definition of the maximum diverse density used in Maron and Lozano-Pérez's paper [17]. The process for finding the maximum diverse density is time-consuming, but is relatively straightforward, that is, using gradient ascent with multiple starting points that are instances from positive bags.

The performance of the Diverse Density algorithm on the *Musk* data is not so good as that of Iterated-discrim APR, but it is the first general multi-instance learning method which has not been geared to the *Musk* data, and it have been

applied to several applications that will be introduced in the later sections of this paper. It is worth mentioning that through combining Diverse Density with EM, Zhang and Goldman [25] presented the EM-DD algorithm which achieved the best performance on the *Musk* data at that time.

4.2 Citation-kNN

Citation-*k*NN [21] is a nearest neighbor algorithm, which borrows the notion of *citation* and *reference* of scientific literatures in the way that a bag is labeled through 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 how the distance metric used by Citation-*k*NN, i.e. the *minimal Hausdorff distance*, can be achieved through 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. a and b, the distance between them can be written as Eq. 9. Usually $\|a-b\|$ is realized as the Euclidean distance.

$$\text{Dist}\,(a,b) = \|a-b\| \tag{9}$$

When the goal is to discriminate the instances, Eq. 9 is enough to be substantiated. But if the goal is to discriminate the bags, then Eq. 9 must be extended because now we should measure the distance between different bags.

Suppose there are two different bags, i.e. $A = \{a_1, a_2, \cdots, a_m\}$ and $B = \{b_1, b_2, \cdots, b_n\}$ where a_i $(1 \le i \le m)$ and b_j $(1 \le j \le n)$ are the instances. It is obvious that they can be regarded as two feature vector sets, where each a_i $(1 \le i \le m)$ or

b_j $(1 \leq j \leq n)$ 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. 5.

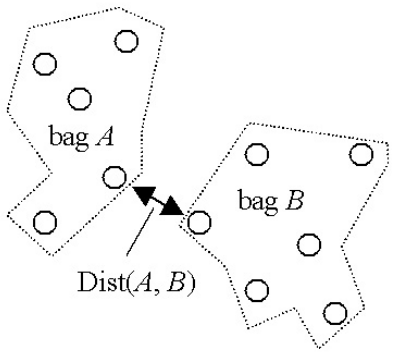


Fig. 5. An intuitive way to define the distance between bags

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

$$\begin{aligned} \text{Dist}\,(A, B) &= \underset{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}}{\text{MIN}}\,(\text{Dist}\,(a_i, b_j)) \\ &= \underset{a \in A}{\text{MIN}}\,\underset{b \in B}{\text{MIN}}\,\|a - b\| \end{aligned} \tag{10}$$

Eq. 10 is exactly the formal definition of the minimum Hausdorff distance used in Wang and Zucker's paper [21]. The Citation-kNN algorithm achieved the best performance on the *Musk* data at the time it was proposed. Recently, Zhou et al. [28] proposed a variant of Citation-kNN for a web mining task through modifying the minimum Hausdorff distance for text features, which will be introduced later.

It is noteworthy that although Wang and Zucker admitted that using the minimal Hausdorff distance did allow k-nearest neighbor algorithm to be adapted to multi-instance learning, they also indicated that it was not sufficient [21]. This is because the common prediction-generating scheme employed by k-nearest neigh-

bor 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 was 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, through analyzing the experimental results presented in the Appendix of Wang and Zucker's paper [21], we 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 [20] and MULTINST [4] on *Musk1*, and RIPPER-MI [8] and GFS elim-count APR [10] on *Musk2*. Moreover, if the fact that the occurrence of positive bags is much smaller than that of negative bags had been considered so that a new bag is negatively labeled when tie occurs in determining its label, the performance of the algorithm employing the minimal Hausdorff distance without utilizing citation and reference would be 90.2% on *Musk1* and 82.4% on *Musk2*, 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, proposed by Wang and Zucker [21].

4.3 ID3-MI

ID3-MI [8] 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 belong 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 almost in the same way as standard decision tree does, i.e. the label of an unseen bag is determined by that of the leaf nodes the bag falling into, it is evident that its key 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 how such a definition can be achieved through 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 Eq. 11.

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

Suppose attribute A is chosen as the test, which partitions D into $\{D_1, D_2, \cdots, D_l\}$, then the entropy of D after partitioning with A is shown as Eq. 12, where $|D|$ and $|D_i|$ denote the number of instances contained in D and D_i, respectively.

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

Then, the *information gain* of A on D is computed according to Eq. 13.

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

Eq. 13 is enough 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 $\pi(X)$ and $\nu(X)$ denote the number of positive and negative bags contained in data set X, respectively. Then the entropy of D before and after partitioning with A defined at the bag level are shown in Eq. 14 and Eq. 15, respectively.

$$\begin{aligned}\mathrm{Info}_{\mathrm{multi}}(D) = &-\frac{\pi(D)}{\pi(D)+\nu(D)}\log_2\left(\frac{\pi(D)}{\pi(D)+\nu(D)}\right)\\ &-\frac{\nu(D)}{\pi(D)+\nu(D)}\log_2\left(\frac{\nu(D)}{\pi(D)+\nu(D)}\right)\end{aligned} \tag{14}$$

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

Then the information gain of A on D defined at the bag level is computed according to Eq. 16.

$$\begin{aligned}\mathrm{Gain}_{\mathrm{multi}}(D, A) &= \mathrm{Info}_{\mathrm{multi}}(D) - \mathrm{Info}_{\mathrm{multi}}(D, A)\\ &= \mathrm{Info}_{\mathrm{multi}}(D) - \sum_{i=1}^{l}\frac{\pi(D_i)+\nu(D_i)}{\pi(D)+\nu(D)}\mathrm{Info}_{\mathrm{multi}}(D_i)\end{aligned} \tag{16}$$

Eq. 16 is exactly the formal definition of the multi-instance entropy used in Chevaleyre and Zucker's paper [8]. Note that although the multi-instance entropy enables the building of multi-instance decision trees, directly implementing it in a decision tree learner may result in complex trees. Therefore, ID3-MI employed a slightly modified divide-and-conquer style, that is, when an instance from a positive bag is positively classified by the tree being induced, removing all the other instances in the bag from the training set. Chevaleyre and Zucker [8] indicated that such a scheme could help generate relatively simple trees.

4.4 RIPPER-MI

RIPPER-MI [8] 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 established.

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 number of instances covered by the rule is very important no matter the rule is being grown or pruned. In detail, when a rule is being grown, rule conditions can be repeatedly added to the rule until the rule covers no negative instances in the growing data set; while when a rule is being pruned, rule conditions can be repeatedly deleted from the rule to maximize some evaluation function, such as the one shown in Eq. 17, where p and n respectively denote the number of positive and negative instances in the pruning data set covered by the rule.

$$v = \frac{p - n}{p + n} \tag{17}$$

Since the only difference between RIPPER-MI and its corresponding supervised learner, i.e. RIPPER, is on the definition of coverage, 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 how such a definition can be achieved through 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 as Eq. 18, where $Cover(R,\ instance_i)$ means the i-th instance in D is covered by R, that is, R is more general than $instance_i$ if the latter is also regarded as a rule.

$$Coverage\,(R) = |\{instance_i | Cover\,(R, instance_i)\}| \tag{18}$$

Eq. 18 is enough when the goal is to discriminate the instances. But for discriminating the bags, the coverage function must be extended. For this we should define

in which situation a bag can be regarded as being covered by rule R. If we adopt the definition shown as Eq. 19, then the coverage function at the bag level is shown as Eq. 20, where bag_i denotes the i-th bag in D.

$$Cover_{multi}\left(R, bag\right) = \exists\left(instance \in bag\right) Cover\left(R, instance\right) \tag{19}$$

$$Coverage_{multi}\left(R\right) = |\{bag_i | Cover_{multi}\left(R, bag_i\right)\}| \tag{20}$$

Eq. 20 is exactly the formal definition of the multi-instance coverage used in Chevaleyre and Zucker's paper [8]. The RIPPER-MI algorithm has been applied to the prediction of mutagenecity, which is a typical benchmark for first-order induction tools. Chevaleyre and Zucker [8] reported that comparing to popular relational learners such as PROGOL and FOIL, RIPPER-MI could generate concise rule sets in less time and with comparable accuracy.

4.5 BP-MIP

BP-MIP [29] 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.

Since the training process of the BP-MIP algorithm is almost the same as that of the classical BP algorithm, it is evident that the key of BP-MIP 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. Below we show how such a definition can be achieved through modifying standard BP algorithm according to 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 Eq. 21, where o_i and d_i is the actual output and desired

[1] This error function was not named in Zhou and Zhang's paper [29]. Here we call it *multi-instance error function* for convenience.

output on the i-th instance, respectively.

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

Eq. 21 is enough when the goal is to discriminate the instances. But for discriminating the bags, the error function must be extended. For this we must define what is the actual output of a bag. If we adopt the definition shown as Eq. 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 Eq. 23.

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

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

Eq. 23 is exactly the formal definition of the multi-instance error function used in Zhou and Zhang's paper [29]. When a trained BP-MIP network is used in prediction, a bag is positively labeled if and only if the output of the network on at least one of its instances is positively predicted. It is worth mentioning that Zhang and Zhou [27] reported later that the performance of BP-MIP could be improved through adopting feature selection techniques such as feature scaling with Diverse Density and feature reduction with principal component analysis.

5 Application

There are two important issues that must be taken into account when applying multi-instance learning technique to real-world tasks. The first is the selection of an appropriate multi-instance learning algorithm. The second is the design of an appropriate method for abstracting the real-world problem to multi-instance representation, that is, determining what is the bag and what are the instances in the bag. Here we call such methods as *bag generators*. In some sense the design of bag generators are more important than the selection of multi-instance learning

algorithms, because the learning task might be easy if an appropriate bag generator is used while very difficult if a poor bag generator is used.

In the paper proposing the Diverse Density algorithm [17], Maron and Lozano-Pérez described two applications. The first one is to learn a simple description of a person from a series of images that are positively labeled if they contain the person and negatively otherwise. Here the bag generator works as follows. Fifty-four subimages of varying centers and sizes are sampled from each image, and each of them is regarded as an instance in the bag corresponding to the original image. Each subimage is divided into three parts roughly corresponding to where the head, torso and legs of the person are, and the three dominant colors each for one subsection are used to represent the subimage. The second application is stock selection where the goal is to select stocks that perform well for fundamental reasons. Here the bag generator works as follows. For every month, 100 stocks with the highest return are put in a positive bag, while the bottom 5 stocks are put in a negative bag. An instance is described with 17 features such as momentum, price to fair-value, etc. Maron and Lozano-Pérez reported that the results achieved by Diverse Density were better than that of a GMO predictor. These two applications seem very interesting, but unfortunately too few details were presented in [17].

Maron and Ratan [18] applied Diverse Density to natural scene classification, which is an image understanding task. Here they tried several kinds of bag generators. All these bag generators regard each image as a bag, which is initially filtered and subsampled to a matrix of *color blobs*. The differences between these bag generators are in the style how they transform various configurations of blobs of each image into instances of the corresponding image bag. A figure from Maron and Ratan's paper [18] illustrated five bag generators, where the first bag generator regards different rows as different instances, the second one regards different single blobs as different instances, the third one regards a blob and its four neighbors together as an instance, the fourth one regards a pair of single blobs as an instance, the fifth one regards a pair of single blobs and their neighbors together as an instance. The instances are represented in similar ways. For example, in the third bag generator, i.e. *single blob with neighbors* (abbreivated as SBN), an instance comprises a 2×2

set of pixels (a blob) and its four neighboring blobs, which is described as a 15-dimensional vector $[x_1, x_2, \cdots, x_{15}]$. Here x_1, x_2, x_3 are the mean RGB values of the central blob, x_4, x_5, x_6 are the differences in mean RGB values between the central blob and the blob above it, etc. The results reported by Maron and Ratan [18] are very well, which demonstrates the great potential of applying multi-instance learning techniques to image involved tasks.

Fig. 6. Five bag generators used by Maron and Ratan

Yang and Lozano-Pérez [24] extended the Diverse Density algorithm and applied it to content-based image retrieval. They developed a complex bag generator. Here each image is regarded as a bag, which is divided into 40 overlapping regions. Regions with low variances are thrown out and the remained regions are regarded as instances in the bag. Each remained region is smoothed and sampled to a low-resolution $h \times h$ matrix. For example, a region containing $m \times n$ pixels is smoothed with a $\frac{2m}{h+1} \times \frac{2n}{h+1}$ averaging kernel and then subsampled to an $h \times h$ matrix. Each element in the resulting matrix is the average gray-scale value of a corresponding subregion. Through concatenating these elements, an h^2-dimensional feature vector is generated. After subtracting the mean of the feature vector from it and then dividing it by its standard deviation, a new h^2-dimensional feature vector is obtained, which is used to describe the corresponding instance. It is worth noting that this bag generator requires converting color images into gray-scale images, therefore it may not suit to the process of color images.

Zhou et al. [31] have also applied Diverse Density to content-based image retrieval. They developed a bag generator named ImaBag, which was derived from a SOM-based image segmentation technique. Here each image is regarded as a bag. The pixels in each image are clustered based on their color and spatial features with a SOM neural network, and then the clustered blocks are merged into a specific number of regions. Each region is represented with a 3-dimensional feature vector

formed by its mean R, G, B values, which is regarded as an instance in the corresponding bag. Zhou et al. [31] reported that the performance of ImaBag is better than that of Yang and Lozano-Pérez's bag generator [24], but worse than that of Maron and Ratan's SBN [18], when they are coupling with Diverse Density.

Recently, Zhou et al. [28] applied multi-instance learning to a specific web mining task, i.e. *web index page recommendation*. *Web index page* is a kind of web page which contains plentiful information but itself only provides titles or brief summaries while leaving the detailed presentation to its linked pages. Here the goal is to identify whether a new web index page will interest a user or not through analyzing the web index pages that the user has browsed. The bag generator regards each web index page as a bag, and the linked pages as the instances in the bag. A figure from Zhou et al.'s paper [28] well illustrated this idea, as shown in Fig. 7. Here each instance is described by a text vector $\mathbf{T} = [t_1, t_2, \cdots, t_n]$, where t_i $(i = 1, 2, \cdots, n)$ is one of the n most frequent terms appearing in the corresponding linked page. $\mathbf{T}$ is obtained through pre-accessing the linked page and then counting the occurrence of different terms. Zhou et al. [28] proposed the Fretcit-kNN algorithm, which is a variant of Citation-kNN, for this task through modifying the minimal Hausdorff distance for measuring the distance between text vectors, and they reported that on this task the performance of Fretcit-kNN is better than that of a classic information retrieval algorithm, i.e. TFIDF. The data sets used by Zhou et al. [28] are freely available at http://cs.nju.edu.cn/people/zhouzh/zhouzh.files/publication/annex/milweb-data.rar.

Multi-instance learning has also been applied to robot control. In order to enable a robot navigate through a large-scaled environment, it is usually needed to accomplish *landmark matching*. That is, the robot should be able to recognize whether or not it is in the vicinity of a given landmark from data taken at the robot's current location. A common approach is to match the visual image to the data taken at landmark position. But since the image may significantly change as small movements around the landmark position are made, this approach encounters difficulties. A better approach is to transform this problem to the learning of geometric patterns. Goldman et al. [12] proposed an online agnostic learning algorithm for

Fig. 7. The web index page is regarded as a bag, while its linked pages are regarded as the instances in the bag

learning the class of discretized constant-dimensional geometric patterns through reducing the problem to that of learning a disjunction of a large set of variables, which is in fact a multi-instance learning algorithm that could learn axis-parallel rectangles. Later, Goldman and Scott [13] extended this algorithm so that it could deal with real-valued geometical patterns.

There are several other applications of multi-instance learning. For example, Weiss and Hirsh [23] proposed to transform event prediction to multi-instance problem so that a kind of time series analysis problem could be solved under the multi-instance learning framework; Ruffo [20] applied a multi-instance decision tree, i.e. Relic, to computer security problems such as password checking, intrusion detection, network management, etc.

6 Extension

In the early years of the research of multi-instance learning, most works were on multi-instance classification with discrete-valued output. Recently, multi-instance regression with real-valued output begins to attract the attention of some researchers. Ray and Page [19] showed that the general formulation of the multi-instance regression task is NP-hard, and proposed an EM-based multi-instance regression algorithm. Amar et al. [2] extended the Citation-kNN and Diverse Density

algorithms for multi-instance regression, and designed some method for artificially generating multi-regression data. The data sets used by them [2] are freely available at http://www.cs.wustl.edu/∼sg/multi-inst-data/.

Chevaleyre and Zucker [8] indicated that strictly speaking, the image involved tasks studied by Maron and his colleagues [17][18] are different from the problem of drug activity prediction. This is because in drug activity prediction, an instance is in fact a description of the bag and for a specific bag, at one time there is only one instance can appear; while in image involved tasks, an instance is a description of a part of the bag and for a specific bag, at one time all the instances must appear together. Chevaleyre and Zucker [8] called the latter case as *multi-part learning*. However, if strictly distinguishing multi-instance and multi-part learning, then besides the original work on drug activity prediction, almost all the application research are actually on multi-part problems. Note that Chevaleyre and Zucker [8] admitted that multi-part problems can be solved with multi-instance learning algorithms without any modification. Therefore, at least at present it is not necessary to distinguish these two notions.

Recently, Weidmann et al. [22] indicated that through employing different assumptions of how the instances' classifications determine their bag's label, different kinds of multi-instance problems could 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. 24, 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.

$$\nu_{MI}(X) \Leftrightarrow \exists x \in X : c_i(x) \tag{24}$$

Based on this recognition, Weidmann et al. [22] defined three kinds of generalized multi-instance problems, i.e. *presence-based MI* [2], *threshold-based MI*, and *count-based MI*. Presence-based MI is defined in terms of the presence of instances of each concept in a bag, as shown in Eq. 25; threshold-based MI requires a certain

[2] Here 'multi-instance' is abbreviated as MI.

number of instances of each concept to be present simultaneously, as defined in Eq. 26; count-based MI requires a maximum as well as a minimum number of instances of a certain concept in a bag, as defined in Eq. 27.

$$\nu_{PB}(X) \Leftrightarrow \forall c \in C : \Delta(X, c) \geq 1 \tag{25}$$

$$\nu_{TB}(X) \Leftrightarrow \forall c_i \in C : \Delta(X, c) \geq t_i \tag{26}$$

$$\nu_{CB}(X) \Leftrightarrow \forall c_i \in C : t_i \leq \Delta(X, c) \leq z_i \tag{27}$$

In Eqs. 25 to 27, ν_{PB}, ν_{TB} and ν_{CB} are functions defined on $2^{\chi} \rightarrow \Omega$, $C \subset \mathcal{C}$ is a given set of concepts, Δ is a counting function $\Delta : 2^{\chi} \times \mathcal{C} \rightarrow \mathbf{N}$ which counts the number of a given concept in a bag, $t_i \in \mathbf{N}$ and $z_i \in \mathbf{N}$ are respectively the lower and upper threshold for concept c_i. Weidmann et al. [22] proposed a *two-level-classification* method for solving these generalized multi-instance problems. They also designed some schemes for artificially generating generalized multi-instance data sets.

It is worth noting that multi-instance learning has also attracted the attention of the ILP community. It has been suggested [9] 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. Recently, Alphonse and Matwin [1] successfully employed multi-instance learning to help relational learning. At first, the original relational learning problem is approximated by a multi-instance problem. The resulting data is fed to feature selection techniques adapted from propositional representations. Then the filtered data is transformed back to relational representation for a relational learner. 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 could act as a bridge between propositional and relational learning.

7 Discussion

The most serious problem encumbering the advance of multi-instance learning is that there is only one popularly used real-world benchmark data, i.e. the *Musk* data sets. Although some application data have been used in some works, they can hardly act as benchmarks for some reasons. For example, the COREL image database has been used by Maron and Ratan [18], Yang and Lozano-Pérez [24], Zhang et al. [26], and Zhou et al. [31], but since the COREL database contains a huge number of images, usually only a portion of the database is used while the portion used by different researchers are usually different; the *Mutagenesis* data has been used by Chevaleyre and Zucker [8] and Alphonse and Matwin [1], but this data is typically used to test ILP learners instead of multi-instance learners. Although there are some artificial data sets [2][19][22], they can hardly be widely used because they are designed for extensions of multi-instance learning such as multi-instance regression and generalized multi-instance learning. In addition, the meaningfulness of artificial data sets may be smaller than that of real-world ones.

Dietterich et al. [10] have estimated that the performance of the Iterated-discrim APR algorithm might be the upper bound on the *Musk* data, but such a performance level has already been exceeded by several algorithms. Table 2 summarizes the best results reported in the literatures.

In fact, the current accuracy on the *Musk* data is so high that it is difficult to anticipate new algorithms do better. Even if some new algorithms could do so, it might not be a good news because these algorithms have great chances to overly favor this specific data. In order to provide a fair basis for testing new algorithms and comparing different algorithms, more data sets are seriously needed. Recently, a real-world data set has been shared out [28]. It is very desirable that more data sets are publicized in the near future.

When proposing the notion of multi-instance learning, Dietterich et al. [10] raised an open problem, i.e. *how to design multi-instance modifications for popular machine learning algorithms*. This open problem has greatly pushed the advance of

Table 2
Predictive error rates (%) on *Musk* data sets reported in literatures

Musk1		*Musk2*	
Algorithm	Error	Algorithm	Error
Ensemble EM-DD	3.1 [30]	Ensemble EM-DD	3.0 [30]
EM-DD	3.2 [25]	EM-DD	4.0 [25]
Ensemble Citation-kNN	5.2 [30]	Ensemble Iterated-discrim APR	6.9 [30]
Ensemble Iterated-discrim APR	7.2 [30]	MI Gaussian RBF Kernel	7.8 [11]
Iterated-discrim APR	7.6 [10]	Iterated-discrim APR	10.8 [10]
Citation-kNN	7.6 [21]	Ensemble Diverse Density	11.0 [30]
MI Polynomial Kernel	7.6 [11]	MI Polynomial Kernel	11.8 [11]
Ensemble Diverse Density	8.2 [30]	MI SVM	12.0 [3]
GFS elim-kde APR	8.7 [10]	Relic	12.7 [20]
GFS elim-count APR	9.8 [10]	Ensemble Citation-kNN	12.9 [30]
Bayesian-kNN	9.8 [21]	Citation-kNN	13.7 [21]
MI Gaussian RBF Kernel	10.9 [11]	MULTINST	16.0 [4]
Diverse Density	11.1 [17]	BP-MIP-PCA	16.7 [27]
TLC without AS	11.3 [22]	TLC without AS	16.9 [22]
RIPPER-MI	12.0 [8]	Diverse Density	17.5 [17]
BP-MIP-PCA	12.0 [27]	Bayesian-kNN	17.6 [21]
MI SVM	13.6 [3]	GFS elim-kde APR	19.6 [10]
BP-MIP-DD	14.1 [27]	BP-MIP	19.6 [29]
Relic	16.3 [20]	BP-MIP-DD	19.6 [27]
BP-MIP	16.3 [29]	RIPPER-MI	23.0 [8]
MULTINST	23.3 [4]	GFS elim-count APR	24.5 [10]
BP	25.0 [10]	BP	32.3 [10]
C4.5	31.5 [10]	C4.5	41.2 [10]

this area. In fact, multi-instance versions of almost all popular machine learning algorithms have been developed in the past years. Now it seems it is the time to propose new challenging problems to stimulate the design of new algorithms. The following one might be a good candidate: *can we effectively label the instances in*

unseen bags? Actually, in the original definition of multi-instance learning, the task is only to learn some concept from the training set for correctly labeling unseen bags. But in most applications, it will be more helpful if the instances in the bags can be correctly labeled. At present several algorithms including APR algorithms, Diverse Density and EM-DD can identify a small area containing real positive instances, which might provide a basis to the solution to the above problem. It is obvious that there are many works to be done in this direction.

Many researchers have tried to introduce new issues to the research scope of multi-instance learning, which has been surveyed in Section 6. Multi-instance regression is evidently worth studying because its potential is clear, at least as Dietterich et al. [10] indicated, if real-valued output could be generated then the binding strength of different molecules could be predicted, which is valuable to drug design. Exploiting multi-instance paradigm to bridge propositional and relational learning is also well worth exploring, because this might bring out new generation powerful learning algorithms. As for multi-part learning, as mentioned in Section 6, at least at present it is not needed to be investigated because all current multi-part problems can be solved by multi-instance learning algorithms. The value of generalized multi-instance learning should be examined from applications. If some real-world generalized multi-instance problems could be identified, not tales such as the door-key problem [22], then generalized multi-instance learning is worth studying. Indeed, keeping an eye on applications may not only help us determine the value of extensions of multi-instance learning, but also help us obtain data sets and stimulating problems for multi-instance learning.

Acknowledgement

to be added ...

References

[1] É. Alphonse and S. Matwin. Filtering multi-instance problems to reduce dimensionality in relational learning. *Journal of Intelligent Information Systems*, vol.22, no.1, pp.23–40, 2004.

[2] R.A. Amar, D.R. Dooly, S.A. Goldman, and Q. Zhang. Multiple-instance learning of real-valued data. In *Proceedings of the 18th International Conference on Machine Learning*, Williamstown, MA, pp.3–10, 2001.

[3] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In S. Becker, S. Thrun, and K. Obermayer, Eds. *Advances of Neural Information Processing Systems 15*, Cambridge, MA: MIT Press, pp.561–568, 2003.

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

[5] P. Auer, P.M. Long, and A. Srinivasan. Approximating hyper-rectangles: learning and pseudo-random sets. *Journal of Computer and System Sciences*, vol.57, no.3, pp.376–388, 1998.

[6] C. Blake, E. Keogh, and C.J. Merz. 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.

[7] A. Blum and A. Kalai. A note on learning from multiple-instance examples. *Machine Learning*, vol.30, no.1, pp.23–29, 1998.

[8] Y. Chevaleyre and J.-D. Zucker. Solving multiple-instance and multiple-part learning problems with decision trees and decision rules. Application to the mutagenesis problem. In *Proceedings of the 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, Ottawa, Canada, *LNAI 2056*, pp.204–214, 2001.

[9] L. De Raedt. Attribute-value learning versus inductive logic programming: the missing links. In *Proceedings of the 8th International Conference on Inductive Logic Programming*, Madison, WI, *LNAI 1446*, pp.1–8, 1998.

[10] T.G. Dietterich, R.H. Lathrop, and T. Lozano-Pérez. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, vol.89, no.1–2, pp.31–71, 1997.

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

[12] S.A. Goldman, S.S. Kwek, and S.D. Scott. Agnostic learning of geometric patterns. *Journal of Computer and System Sciences*, vol.62, no.1, pp.123–151, 2001.

[13] S.A. Goldman and S.D. Scott. Multiple-instance learning of real-valued geometric patterns. *Annals of Mathematics and Artificial Intelligence*, vol.39, no.3, pp.259–290, 2003.

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

[15] P.M. Long and L. Tan. PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. *Machine Learning*, vol.30, no.1, pp.7–21, 1998.

[16] O. Maron. Learning from ambiguity. PhD dissertation, Department of Electrical Engineering and Computer Science, MIT, Jun. 1998.

[17] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In M.I. Jordan, M.J. Kearns, and S.A. Solla, Eds. *Advances in Neural Information Processing Systems 10*, Cambridge, MA: MIT Press, pp.570–576, 1998.

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

[19] S. Ray and D. Page. Multiple instance regression. In *Proceedings of the 18th International Conference on Machine Learning*, Williamstown, MA, pp.425–432, 2001.

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

[21] J. Wang and J.-D. Zucker. Solving the multiple-instance problem: a lazy learning approach. In *Proceedings of the 17th International Conference on Machine Learning*, San Francisco, CA, pp.1119–1125, 2000.

[22] N. Weidmann, E. Frank, and B. Pfahringer. A two-level learning method for generalized multi-instance problem. In *Proceedings of the 14th European Conference on Machine Learning*, Cavtat-Dubrovnik, Croatia, *LNAI 2837*, pp.468–479, 2003.

[23] G.M. Weiss and H. Hirsh. Event prediction: learning from ambiguous examples. In *Working Notes of the NIPS'98 Workshop on Learning from Ambiguous and Complex Examples*, Breckenridge, CO, 1998.

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

[25] Q. Zhang and S.A. Goldman. EM-DD: an improved multi-instance learning technique. In T.G. Dietterich, S. Becker, and Z. Ghahramani, Eds. *Advances in Neural Information Processing Systems 14*, Cambridge, MA: MIT Press, pp.1073–1080, 2002.

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

[27] M.-L. Zhang and Z.-H. Zhou. Improve multi-instance neural networks through feature selection. *Neural Processing Letters*, vol.19, no.1, pp.1–10, 2004.

[28] Z.-H. Zhou, K. Jiang, and M. Li. Multi-instance learning based web mining. *Applied Intelligence*, in press.

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

[30] Z.-H. Zhou and M.-L. Zhang. Ensembles of multi-instance learners. In *Proceedings of the 14th European Conference on Machine Learning*, Cavtat-Dubrovnik, Croatia, *LNAI 2837*, pp.492–502, 2003.

[31] Z.-H. Zhou, M.-L. Zhang, and K.-J. Chen. 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, pp.565–569, 2003.

[32] J.-D. Zucker and J.-G. Ganascia. Changes of representation for efficient learning in structural domains. In *Proceedings of the 13th International Conference on Machine Learning*, Bary, Italy, pp.543–551, 1996.