Tsinghua Science and Technology

Volume 18 | Issue 4 Article 4

2013

MICkNN: Multi-Instance Covering kNN Algorithm

Shu Zhao
Department of Computer Science and Technology and Key Lab of Intelligent Computing and Signal Processing, Anhui University, Hefei 230601, China

Chen Rui
Department of Computer Science and Technology and Key Lab of Intelligent Computing and Signal Processing, Anhui University, Hefei 230601, China

Yanping Zhang
Department of Computer Science and Technology and Key Lab of Intelligent Computing and Signal Processing, Anhui University, Hefei 230601, China

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 04/10 pp360-368
Volume 18, Number 4, August 2013

MIC*k*NN: Multi-Instance Covering *k*NN Algorithm

Shu Zhao, Chen Rui, and Yanping Zhang*

Abstract: Mining from ambiguous data is very important in data mining. This paper discusses one of the tasks for mining from ambiguous data known as multi-instance problem. In multi-instance problem, each pattern is a labeled bag that consists of a number of unlabeled instances. A bag is negative if all instances in it are negative. A bag is positive if it has at least one positive instance. Because the instances in the positive bag are not labeled, each positive bag is an ambiguous. The mining aim is to classify unseen bags. The main idea of existing multi-instance algorithms is to find true positive instances in positive bags and convert the multi-instance problem to the supervised problem, and get the labels of test bags according to predict the labels of unknown instances. In this paper, we aim at mining the multi-instance data from another point of view, i.e., excluding the false positive instances in positive bags and predicting the label of an entire unknown bag. We propose an algorithm called Multi-Instance Covering kNN (MICkNN) for mining from multi-instance data. Briefly, constructive covering algorithm is utilized to restructure the structure of the original multi-instance data at first. Then, the kNN algorithm is applied to discriminate the false positive instances. In the test stage, we label the tested bag directly according to the similarity between the unseen bag and sphere neighbors obtained from last two steps. Experimental results demonstrate the proposed algorithm is competitive with most of the state-of-the-art multi-instance methods both in classification accuracy and running time.

Key words: mining ambiguous data; multi-instance; classification; constructive covering algorithm; kNN algorithm

1 Introduction

Multi-Instance (MI) problem was introduced to solve the problem of musk activity prediction by Dietterich et al.[1] Note that a specific drug molecule can adopt a wide range of shapes by rotating some of its internal bonds, as shown in Fig. 1. A molecule is *active* if one or some of its shapes bind well to the target, though the others do not. A molecule is *inactive* if none of its configurations binds well to the target. In *supervised problem*, each training sample is perfectly labeled; in *unsupervised problem*, all the training samples are not labeled. Thus, supervised learning algorithms can not achieve satisfactory results when dealing with MI problem. In *MI problem*, all the training patterns are only labeled at the bag-level. Each positive bag could contain extensive *false positive instances*, therefore, the proportion of the noises in the positive bag would be extremely high.

Due to its particularity, MI problem has attracted

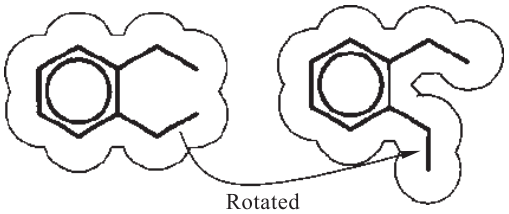


Fig. 1 The shape of a molecule changes as it rotates internal bonds[1].

• Shu Zhao, Chen Rui, and Yanping Zhang are with Department of Computer Science and Technology and Key Lab of Intelligent Computing and Signal Processing, Anhui University, Hefei 230601, China. E-mail: zhangyp2@gmail.com.

∗ To whom correspondence should be addressed.
Manuscript received: 2013-06-15; accepted: 2013-06-25

much attention and traditional supervised algorithms cannot be used directly. More and more real-world data mining problems can be appropriately modeled under MI category, such as web mining[2-4], text categorization[5,6], spam filtering[7], content-based image categorization[8,9], nature scene classification[10,11], visual tracking[12-16], object detection[17], computer security[18], etc.

A large number of MI algorithms are proposed to solve these MI problems. Some of them first work on the instance-level and then combine the classifications of instances to obtain the label of an unknown bag, while others focus only on the bag-level without necessarily knowing the label of each instance. It looks like majority of these approaches belongs to instance-level group. For example, Dietterich et al.[1] proposed three specialized tailored Axis-Parallel Rectangle (APR) algorithms. Maron and Lozano-Pérez[19] proposed the first MI framework, Diverse Density (DD). Zhang and Goldman[20] proposed the EM-DD algorithm, as an enhanced version of DD, which combines EM with DD together. Some introduce a latent variable to calculate whether an instance is from the concept, such as the mi-SVM method[5,21] and the methods in Refs. [22, 23]. A neural network based MI algorithm named RBF-MIP, which is derived from the popular Radial Basis Function (RBF) method is proposed in Ref. [24] by Zhang and Zhou. Kwok and Cheung[25] addressed the ambiguity of instance labels in the design of MI kernels by using the method of marginalized kernels. The maximum margin MI clustering (M^3IC) algorithm which enables an efficient optimization solution with a combination of Constrained *Concave-Convex Procedure* (*CCCP*) and the cutting plane method was coined by Zhang et al.[26] Recently, the parallel MI algorithm MIBP which is based on the parallel training of MI neural network was designed by Li et al.[27]

For MI algorithms which belong to the bag-level, Wang et al.[28] defined kernels between bags. Wang and Zucker[29] proposed two MI learning *k*NN algorithms: Bayesian-*k*NN and Citation-*k*NN, as variants of the standard *k*NN algorithm. Zhou et al.[6] defined a graph with instances from a bag. Deselaers and Ferrari[30] treated bags as nodes and instances as the nodes states in a conditional random field model. Babenko et al.[31] modeled bags as manifolds in the instance space. Zhang et al.[32] incorporated structure information between bags or instances. Recently, Jiang et al.[33] proposed a bag-level algorithm called Bayesian Citation-*k*NN (BC*k*NN) with distance weighting as an improved version of MI lazy learning *k*NN algorithm.

Although many MI algorithms effectively worked, it is noteworthy that almost all these algorithms mentioned above try to select the positive bags or the *true positive instances* in the positive bags rather than exclude the *false positive instances*. When labeling the unknown bags, if these algorithms can find positive instances in a bag, the bag is positive; otherwise, the bag is negative. These methods are very complex and often take a long time to run.

As Maron[34] pointed out, the irregular structural features of the positive bags, i.e., the ambiguity of the positive bags leads to the inherent difficulty to MI problem. In order to improve the classification accuracy and reduce the complexity of the algorithm, it is necessary to break through the restrictions of the structure of the MI data set and reorganize the structure of the bags, so that the real characteristics of the MI data set can be revealed. In this paper, constructive covering algorithm is utilized to generate a set of *covers* to help exclude the false positive instances by using *k*NN algorithm. In short, we reorganize the structure of MI data using constructive covering algorithm, exclude the *false positive instances* as many as possible, discriminate the *false positive instances*, and label the whole bags and sphere neighbors instead of labeling single instances.

2 Related Work

2.1 Constructive covering algorithm

Constructive Covering Algorithm (CCA) was proposed by Zhang and Zhang[35] in 1999, as a constructive supervised learning algorithm for McCulloch-Pitts Neural Model[36]. The main idea of CCA is mapping all patterns in the data set to a d-dimensional sphere S^d at first. Then the sphere neighbors (*Covers*) are utilized to divide the patterns.

Given a classification data set $D = \{X_i | X_i = (x_i, y_i), i = 1, 2, \cdots, N, y_i = 1, 2, \cdots, m\}$, where $x_i \in \mathbf{R}^d$, y represents m classes. Let N and d be the size and dimension of D, $X_i = (x_i, y_i)$ be a pattern in D. Let D_i be a subset of D, which contains all the patterns of the i-th class, $i = 1, 2, \cdots, m$. The classification model created by CCA is a cover set: $C = \{C_i | C_i = (c, r), i = 1, 2, \cdots\}$, where c is the center of a cover, r is its radius.

Since the dimension of instances in MI data set is

identical, all input vectors can be restricted to a d-dimensional hyper-sphere S^d. The input vector is a bounded set D of the d-dimensional space.

Define a transformation T:

$$T(x) = (x, \sqrt{R^2 - \|x\|^2}), R \geqslant \max\{\|x\| | x \in D\} \quad (1)$$

Use T to do a projection: T: $D \rightarrow S^d$, S^d is a d-dimensional sphere of the $(d+1)$-dimensional space. R is the radius of the $(d+1)$-dimensional space, the value of R must be greater or equal to the maximum value of the mold length of all the instances. $\sqrt{R^2 - \|x\|^2}$ is the additional value of x, i.e., the value of the $(d+1)$-dimension after projection as shown in Fig. 2.

After all the instances in D are projected upward on S^d by transformation T, construct a series of positive covers ("positive sphere neighbors") that only consist of instances in the positive bags and a series of negative covers ("negative sphere neighbors") that only consist of instances in the negative bags. This procedure is shown as follows.

To generate the sphere neighbors that only contain instances belonging to the same class, first of all, select an instance $x_i \in D$, randomly. Let X be the set of instances has the same label as x_i. Therefore, the set of instances has the opposite label from x_i can be denoted as $\widetilde{X}$.

Then, we compute the distance d_1 between x_i and $\widetilde{x_j}$, $\widetilde{x_j}$ is the nearest instance from x_i which belongs to the set of $\widetilde{X}$ and the distance d_2 between x_i and x_k, where x_k is the furthest instance from x_i which belongs to the set of X. Note that in CCA, d_2 must be smaller than d_1.

The above procedure can be expressed as Eqs. (2) and (3):

$$d_1 = \max\{ < x_i, \widetilde{x_j} > | x_i \in X, x_j \in \widetilde{X}\} \quad (2)$$

$$d_2 = \min\{ < x_i, x_k > | x_i, x_k \in X, < x_i, x_k >> d_1\} \quad (3)$$

where $\langle a, b \rangle$ denotes the "inner product" between instances a and b. Note that the bigger "inner product" indicates the smaller distance.

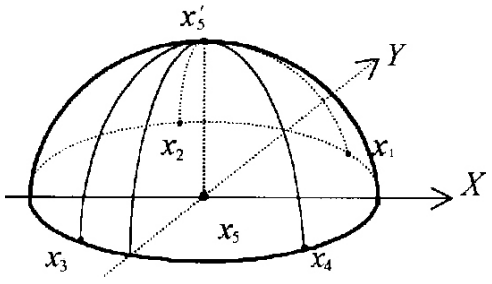


Fig. 2 Input vectors of instance x and their projections.

Next, we calculate the radius of a sphere neighbor so that in this sphere neighbor there are only instances belonging to the same class (See Fig. 3). Let r be the radius of a sphere neighbor. The following formula is utilized to calculate the value of r:

$$r = (d_1 + d_2)/2 \quad (4)$$

The basic constructive covering algorithm is shown in Algorithm 1.

We are inspired by the inherent feature of CCA. That is, the learning results of CCA is a series of $Covers$, each of which only contains samples belonging to the same class. It is this feature that makes CCA very suitable for breaking through and reorganizing the structure of MI data.

2.2 k-nearest neighbor algorithm

k-Nearest Neighbor (kNN) algorithm[37] is one of the most popular learning algorithms in data mining. kNN

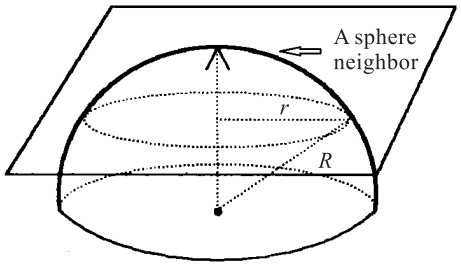


Fig. 3 A sphere neighbor for classification.

Algorithm 1 Basic Constructive Covering Algorithm

Input:

D: Training set

m: Number of classes of training set

Output:

C: Cover sets

Define a transform:
$T(x) = (x, \sqrt{R^2 - \|x\|^2}), R \geqslant \max\{\|x\| | x \in D\}$;
Use $T(x)$to do a projection: $D \longrightarrow S^d$, where
S^d is an d-dimensional sphere in $(d+1)$-dimensional space;
$C = \phi$;
for $i = 1$ **to** m **do**
  while $D_i \neq \phi$ **do**
    Randomly select a pattern x from D;
    $d_1 = \max\{< x, x' > | x \in D_i, x' \notin D_i\}$;
    $d_2 = \min\{< x, x' > | x, x' \in D_i, < x, x' >> d_1\}$;
    $r = (d_1 + d_2)/2$;
    Create a cover c_i with x as its center and r as its radius;
    $C = C \bigcup c_i$;
  end while
end for
return C;

is well known for its relatively simple implementation and decent results. The main idea of *k*NN algorithm is to find a set of k objects in the training data that are close to the test pattern, and base the assignment of a label on the predominance of a particular class in this neighbor. *k*NN is a lazy learning technique based on voting and distances of the k nearest neighbors. Given a training set D and a test pattern x, *k*NN computes the similarity (distance) between x and the nearest k neighbors. The label of x is assigned by voting from the majority of neighbors.

The basic *k*NN algorithm is shown in Algorithm 2.

3 MI Covering *k*NN (MIC*k*NN) Algorithm

3.1 Definition of multi-instance problem

In this paper we follow the trend established by the majority of the work in this field and assume the set of class labels $\Omega = \{+, -\}$. Let χ be the instance space. Then the MI problem is a function $f_{\mathrm{MI}} : 2^{\chi} \longrightarrow \Omega$. Given an MI data set D. The instances in D are defined as $\{x_k | k = 1, 2, \cdots, m\}$, thus, $D = \{x_k | k = 1, 2, \cdots, m\}$. All the instances in the positive bags and negative bags is defined as D^+ and D^-, respectively, where $D^+ = \{x_i^+ | i = 1, 2, \cdots, p\}$, $D^- = \{x_j^- | j = 1, 2, \cdots, n\}$, $D = D^+ \cup D^-$. Let d be the dimension of x. Note that, in MI problem, each instance only belongs to one specific bag. In other words, two different bags cannot share the same instance. Therefore, given two MI bags B_i and B_j, $B_i \bigcap B_j = \varnothing$.

3.2 Exclude the noises in positive bags

An example of the original distribution of MI data in the two-dimensional space is given in Fig. 4.

Algorithm 2 Basic *k*NN Algorithm

Input:

D: Training set
n: Number of patterns in D
x: Test pattern

Output:

ℓ: label of x

1: **for** $i = 1$ **to** n **do**
2: Compute $d(x, x_i)$, the distance between x and x_i;
3: **end for**
4: Select the set of k nearest training patterns for x;
5: $\ell = \arg\max_{\upsilon \in L} f(\upsilon = \mathrm{class}(c_{x_i}))$;
$f(\cdot)$ is an indicator function that returns the value 1 if its argument is true and 0 otherwise.
6: **return** ℓ;

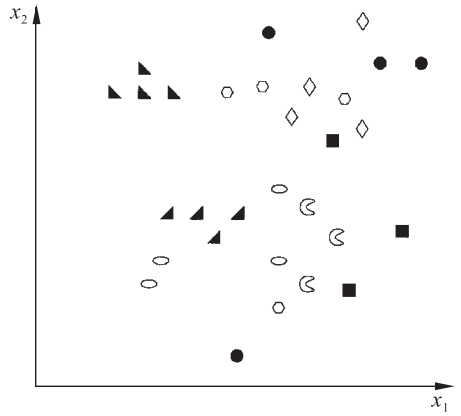


Fig. 4 The original distribution of MI bags before preprocess.

Unfilled shapes represent feature vectors of positive bags and filled shapes represent feature vectors of negative bags. All points of the same shape denote feature vectors of the same bag. For example, the bag "*unfilled regular hexagon*" is a positive bag and it has four instances. As Fig. 4 depicts, from the distribution of instances in each positive bag, we cannot find the distribution pattern that the instances in each positive bag have got to abide due to the existence of the *false positive instances* in each positive bag. Specifically, it is difficult to find a linear or non-linear dividing line to separate the positive bags and the negative bags precisely. On the other hand, the distribution of instances in the same positive bag is random and discrete. From our perspective, it is precisely because of the irregular structural features of the positive bags, i.e., the ambiguity of the positive bags leads to the inherent difficulty of MI problem.

The learning result of CCA is a *Cover* set and this feature that makes CCA very suitable for breaking through and restructuring the structure of MI data. By regarding the *Cover* set as the new structure of MI bags, the original structure of bags is reorganized from irregular to regular, so that the *false positive instances* can be excluded.

In order to discriminate the noises in the positive bags, construct a set $\mathrm{PCover} = \{\mathrm{PCover}_i | i = 1, 2, \cdots, p\}$ that only contains instances from the positive bags and a set $\mathrm{NCover} = \{\mathrm{NCover}_j | j = 1, 2, \cdots, n\}$ that only contains instances from the negative bags, respectively. Thus, the irregular discrete structure of bag can be broken through by converting the original data set into a set $\mathrm{Cover} = \mathrm{PCover} \bigcup \mathrm{NCover}$ as depicted in Fig. 5. The new structure $Cover$ is very convenient

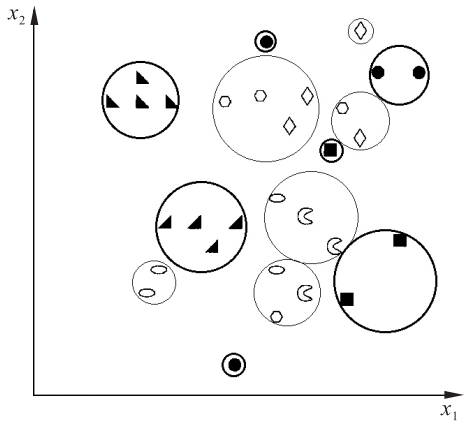


Fig. 5 The bag structure of MI data set after preprocess. Instances from negative bags are covered by "thick lines"; instance from positive bags are covered by "thin lines".

for excluding the *false positive instances* in the positive sphere neighbors by using *k*NN algorithm. The details are as follows.

First, construct the set Cover by using CCA. After this procedure, *k*NN is utilized to exclude the *false positive sphere neighbors* that mainly consist of *false positive instances*. According to the *k*NN algorithm, a Cover_i obtained by CCA is considered as a whole. For each PCover_i, if the majority of its nearest neighbors are belong to the set NCover, we delete it from the set PCover and add it to the set NCover. The distance between Cover_i and Cover_j can be calculated by using *Hausdorff distance*. Edgar[38] and Wang and Zucker[29] pointed out two kinds of such *Hausdorff distance*, that is *maximal Hausdorff distance* (maxHD) and *minimal Hausdorff distance* (minHD).

According to the definition, given two sets of instances $A = \{a_1, \cdots, a_m\}$ and $B = \{b_1, \cdots, b_n\}$.

The maxHD is defined as

$$\text{maxHD}(A, B) = \max\{h(A, B), h(B, A)\} \quad (5)$$

where

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \quad (6)$$

The minHD is defined as

$$\text{minHD}(A, B) = \min \|a - b\| \quad (7)$$

where $\|a - b\|$ is the *Euclidean distance* between instance a and instance b.

The complete description of excluding the noises in positive bags is illustrated in Algorithm 3.

After preprocess the MI data in the first stage, the original structure of MI data set is converted into positive cover set PCover and negative cover set NCover. In addition, a fair amount of *false positive instances* in the set PCover are excluded.

Algorithm 3 Exclude the Noises in Positive Bags

Input:

$D = D^+ \cup D^-$: MI data set

p: The cardinality of PCover

n: The cardinality of NCover

Output:

Cover = PCover $\bigcup$ NCover: Cover sets consist all instances in D

```
PCover =∅, NCover = ∅;
p = 0, n = 0;
repeat
    Randomly select an instance x from D;
    if x ∈ D+ then
        Obtain PCover_i using CCA;
        D+ = D+ − PCover_i;
        p = p + 1;
    else
        Obtain NCover_j using CCA;
        D− = D− − NCover_j;
        n = n + 1;
    end if
until D = ∅;
PCover = {PCover_i | i = 1, 2, ..., p};
NCover = {NCover_j | j = 1, 2, ..., n};
for i = 1 to p do
    Compute HD(PCover_i, Cover);
    ℓ_i = Learner(kNN)
    if ℓ_i = −1 then
        ℓ(PCover_i) =−1;
        PCover = PCover − PCover_i
        NCover = NCover ⋃ PCover_i;
    end if
end for
return Cover = PCover ⋃ NCover;
```

3.3 Predict the labels of test bags

In this stage, a PCover_i or an NCover_j is treated as the new structure of a bag, then the *k*NN algorithm at the bag-level is utilized to predict the real labels of the unknown bags. Because of a large number of noises in the positive bags are excluded during the procedures above. It is quite convenient to predict the labels of test bags by using *k*NN algorithm at the bag-level. We compute the similarity between each test bag and its nearest neighbors, if there are more negative $Covers$ around a test bag, the label of the test bag is negative, otherwise positive. The complete description of the predict algorithm is illustrated in Algorithm 4.

Algorithm 4 Predict the Labels of Test Bags

Input:

Cover = PCover $\bigcup$ NCover: Cover sets
testBags: The set of test bags
t: The number of test bags

Output:

L: Label set of test bags

```
for i = 1 to t do
    Compute HD(testBag_i, Cover);
    ℓ_i = Learner(kNN)
    if ℓ_i = −1 then
        L(testBag_i) = −1;
    else
        L(testBag_i) = +1;
    end if
end for
return L;
```

4 Experiments

All the experimental results and analysis are summarized in this section. MIC*k*NN excludes the false positive instances at the cover-level and labels the unknown bags at the bag-level. The experiments are performed on two real world benchmark date sets. All the experiments are performed on a 2.8 GHz Inter Pentium Dual-Core PC running Windows 7 with 2.0 GB main memory.

4.1 Data sets

Musk data sets (ftp://ftp.ics.uci.edu/pub/machine-learning-databases/musk/), i.e., Musk1 and Musk2 are the only real-world benchmark data sets for MI problem so far. There are 92 bags in Musk1, 47 of them are active and the remaining 45 are inactive. There are 102 bags in Musk2, 39 of them are active and the remaining 63 are inactive.

Note that the main difference between the two data sets is that Musk2 contains molecules that have more possible conformations than Musk1. The total instances of Musk1 and Musk2 are 476 among 92 bags and 6598 in 102 bags. The average instances per bag is 5.2 in Musk1, but the average instances per bag is 64.7 in Musk2. Thus, the proportion of the noise of positive bags in Musk2 is much greater than that in Musk1. This makes the prediction of Musk2 even more difficult than Musk1. Some detailed information of these two Musk data sets are shown in Table 1.

Table 1 Summary description of the two Musk data sets.

Data set	Musk	Non-Musk	Total bags	Instances per bag			
				Total	Min	Max	Average
Musk 1	47	45	92	476	2	40	5.2
Musk 2	63	39	102	6598	1	1044	64.7

4.2 Experimental methods and results

We perform several iterations of *10-fold cross validation* for different values of k_1 and k_2. k_1 is the value of *k*NN algorithm applied to exclude the noises in Algorithm 3. k_2 is the value of *k*NN algorithm used to label the test bags in Algorithm 4. In addition, two kinds of *Hausdorff distance*: minHD and maxHD are used for experiments.

In each *10-fold cross validation* run, the data set is randomly divided into 10 partitions. The learner is then trained 10 times, with each iteration involving a different combination of 9 partitions as the training set and the remaining one partition as the testing set.

The details of experimental results are shown in Table 2. The value of k_1 used in the first stage for discriminating the *false positive covers* is set to 1, 3, 5, and 7, respectively. The value of k_2 used in the second stage for predicting the labels of test bags is set the same value as k_1. Since the value of k_1 and k_2 are both odd numbers, the situation that the number of positive covers or bags equals the negative covers or bags in the *k*NN algorithm will not happen.

For Musk1, the minHD method acquired the best prediction accuracy 92.8%, wherein k_1=3, k_2=3. For Musk2, the minHD method acquired the best prediction accuracy 91.2%, wherein k_1=1, k_2=7. Obviously, the

Table 2 The performance of MIC*k*NN (% correct) on two Musk data sets.

k_1	k_2	Musk1		Musk2	
		minHD (%)	maxHD (%)	minHD (%)	maxHD (%)
1	1	87.8	76.8	87.3	78.0
	3	85.3	81.3	86.4	84.2
	5	86.4	73.4	84.6	80.3
	7	92.0	74.0	**91.2**	77.3
3	1	87.8	76.8	89.7	78.0
	3	**92.8**	77.6	86.4	74.2
	5	91.3	74.3	82.4	77.3
	7	92.0	75.0	84.0	70.1
5	1	91.1	79.1	82.0	78.2
	3	91.3	75.3	78.6	74.3
	5	86.4	80.4	83.1	70.3
	7	92.2	79.6	75.6	69.3
7	1	89.3	82.5	80.3	78.5
	3	87.1	80.4	81.2	74.8
	5	91.0	76.3	82.8	68.7
	7	83.1	74.0	78.6	70.1

performance of minHD is much more effective than maxHD. Additionally, it seems that when the value of k is smaller, the performance is better.

4.3 Experimental comparisons and analysis

Various learning algorithms have been adapted to the MI problem. We compare MIC*k*NN both with previous instance-level classifiers and bag-level classifiers. Tables 3-6 show that the performance of MIC*k*NN is competitive with most well-established MI learning algorithms both in accuracy and computation time. The first seven algorithms are instance-level MI algorithms and the last four algorithms are bag-level MI algorithms.

Table 3 Accuracy on Musk1 benchmark MI data set.

Algorithm		Accuracy (%)
Instance-level methods	Iterated-APR	92.4
	DD	88.9
	EM-DD	**96.8**
	mi-SVM	87.4
	MILIS	88.6
	BP-MIP	83.7
	ISMIBP	83.8
Bag-level methods	Bayesian-*k*NN	90.2
	Citation-*k*NN	92.4
	BC*k*NN	92.2
	MIC*k*NN	**92.8**

Table 4 Running time on Musk1 benchmark MI data set.

Algorithm		Computation time (s)
Instance-level methods	Iterated-APR	465
	DD	61 240
	EM-DD	36 310
	mi-SVM	N/A
	MILIS	N/A
	BP-MIP	123
	ISMIBP	130
Bag-level methods	Bayesian-*k*NN	N/A
	Citation-*k*NN	**30**
	BC*k*NN	48
	MIC*k*NN	**35**

Table 5 Accuracy on Musk2 benchmark MI data set.

Algorithm		Accuracy (%)
Instance-level methods	Iterated-APR	89.2
	DD	82.5
	EM-DD	**96.0**
	mi-SVM	83.6
	MILIS	91.1
	BP-MIP	80.4
	ISMIBP	78.6
Bag-level methods	Bayesian-*k*NN	82.4
	Citation-*k*NN	86.3
	BC*k*NN	83.2
	MIC*k*NN	**91.2**

Table 6 Running time on Musk2 benchmark MI data set.

Algorithm		Computation time (s)
Instance-level methods	Iterated-APR	**1560**
	DD	184 320
	EM-DD	42 370
	mi-SVM	N/A
	MILIS	N/A
	BP-MIP	3510
	ISMIBP	3525
Bag-level methods	Bayesian-*k*NN	N/A
	Citation-*k*NN	5920
	BC*k*NN	6617
	MIC*k*NN	**6739**

For the classification accuracy, Table 3 and Table 5 report the highest prediction accuracy of the compared MI algorithms and MIC*k*NN achieved in Musk1 and Musk2. For the bag-level multi-instance *k*NN algorithms, including MIC*k*NN, the average precisions of the compared bag-level MI *k*NN algorithms are not available due to the value of k can be an arbitrary integer. For details, MIC*k*NN only lags behind iterated-APR and EM-DD in Musk1 and EM-DD in Musk2. Actually, the average prediction accuracy of EM-DD is 84.8% for Musk1 and 84.9% for Musk2[6, 9, 30, 39, 40]. Currently, MIC*k*NN can catch up with EM-DD sometimes when the value of k is appropriate. In addition, MIC*k*NN outperforms other bag-level MI learning *k*NN algorithms, i.e., Bayesian-*k*NN, Citation-*k*NN, and BC*k*NN. Besides, for the task of predicting the molecule activity of Musk2 data set, only MIC*k*NN achieves the prediction accuracy above 90%. This illustrates that MIC*k*NN algorithm can effectively exclude a lot of noises in the benchmark MI data set when compare with other bag-level MI *k*NN algorithms. In addition, please note that only MIC*k*NN has the ability to restructure the structure of the original bag of MI data.

For the computation time, the running time of the instance-level MI algorithms often take more time than the bag-level algorithms. MIC*k*NN is faster than all the instance-level MI algorithms and can compete with the other bag-level *k*NN algorithms when deal with Musk1. For Musk2, the CPU time is comparable with the other bag-level *k*NN algorithms.

5 Conclusions

In this paper, a novel algorithm MIC*k*NN for mining MI data is proposed. First, in order to discriminate the *false positive instances*, CCA is utilized to break through and restructure the structure of the original bags so that the noises in the positive bags can be excluded by using *k*NN algorithm. Then, the set of $Covers$ constructed are regarded as a whole for labeling the unknown bags. MIC*k*NN is a cover-level MI algorithm, it is different from the previous bag-level and instance-level MI methods. MIC*k*NN is very convenient for excluding the false positive instances in MI data and predicting the labels of unknown bags. The experimental results show that MIC*k*NN is competitive with the state-of-the-art MI algorithms.

Whether other supervised learning algorithms can be applied to discriminate the noises in positive bags and label the unknown bags is one of our future work. Besides, we will also explore the specific applications of MIC*k*NN algorithm. In addition, the same as Bayesian-*k*NN, Citation-*k*NN, and BC*k*NN, the basic Euclidean distance is also utilized to measure the similarity between instances in MIC*k*NN. Whether some novel proposed similarity measures can be applied to improve the accuracy of MIC*k*NN is also one of the main directions for our future work.

Acknowledgements

The authors would like to thank all the reviewers for their insightful and constructive comments. This work was partly supported by the National Natural Science Foundation of China (Nos. 61073117 and 61175046), the Provincial Natural Science Research Program of Higher Education Institutions of Anhui Province (No. KJ2013A016), the Academic Innovative Research Projects of Anhui University Graduate Students (No. 10117700183), and the 211 Project of Anhui University.

References

[1] 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, pp. 31-71, 1997.

[2] B. B. Ni, Z. Song, and S. C. Yan, Web image mining towards universal age estimator, in *Proceedings of the 17th ACM International Conference on MultimediaInt*, ACM, 2009, pp. 85-94.

[3] A. Zafra, C. Romero, S. Ventura, and E. Herrera-Viedma, Multi-instance genetic programming for web index recommendation, *Expert Systems with Applications*, vol. 36, no. 9, pp. 11470-11479, 2009.

[4] Z. H. Zhou, K. Jiang, and M. Li, Multi-instance learning based web mining, *Applied Intelligence*, vol. 22, no. 2, pp. 135-147, 2005.

[5] S. Andrews, I. Tsochantaridis, and T. Hofmann, Support vector machines for multiple-instance learning, *Advances in Neural Information Processing Systems*, vol. 15, pp. 561-568, 2002.

[6] Z. H. Zhou, Y. Y. Sun, and Y. F. Li, Multi-instance learning by treating instances as non-I.I.D samples, in *Proceedings of the 26th Annual International Conference on Machine Learning*, ACM, 2009, pp. 1249-1256.

[7] Z. Jorgensen, Y. Zhou, and M. Inge, A multiple instance learning strategy for combating good word attacks on spam filters, *The Journal of Machine Learning Research*, vol. 9, no. 6, pp. 1115-1146, 2008.

[8] J. B. Bi, Y. X. Chen, and J. Z. Wang, A sparse support vector machine approach to region-based image categorization, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 1121-1128, 2005.

[9] Z. Y. Fu, A. Robles-Kelly, and J. Zhou, MILIS: Multiple instance learning with instance selection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 958-977, 2011.

[10] S. H. Feng and D. Xu, Transductive multi-instance multi-label learning algorithm with application to automatic image annotation, *Expert Systems with Applications*, vol. 37, no. 1, pp. 661-670, 2010.

[11] D. T. Nguyen, C. D. Nguyen, R. Hargraves, L. A. Kurgan, and K. J. Cios, mi-DS: Multiple-instance learning algorithm, *IEEE Transactions on Systems, Man, and Cybernetics Society. Part B, Cybernetics*, vol. 43, no. 1, pp. 143-154, Feb. 2013.

[12] B. Babenko, M. H. Yang, and S. Belongie, Visual tracking with online multiple instance learning, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 983-990.

[13] B. Babenko, M. H. Yang, and S. Belongie, Robust object tracking with online multiple instance learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619-1632, 2011.

[14] C. Leistner, A. Saffari, and H. Bischof, MIForests: Multiple-instance learning with randomized trees, in *Proceedings of the 11th European Conference on Computer Vision: Part VI*, Springer, 2010, pp. 29-42.

[15] Y. Xie, Y. Y. Qu, C. H. Li, and W. S Zhang, Online multiple instance gradient feature selection for robust visual tracking, *Pattern Recognition Letters*, vol. 33, no. 9, pp. 1075-1082, 2012.

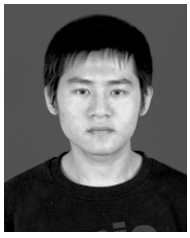
[16] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof, On-line semi-supervised multiple-instance boosting, in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 1879-1886.

[17] Z. Q. Qi, Y. T. Xu, L. S. Wang, and Y. Song, Online multiple instance boosting for object detection, *Neurocomputing*, vol. 74, no. 10, pp. 1769-1775, 2011.

[18] M. Bellare, T. Ristenpart, and S. Tessaro, Multi-instance security and its application to password-based cryptography, *Advances in Cryptology-CRYPTO 2012*, Springer, pp. 312-329, 2012.

[19] O. Maron and T. Lozano-Pérez, A framework for multiple-instance learning, *Advances in Neural Information Processing Systems*, vol. 10, pp. 570-576, 1998.
[20] Q. Zhang and S. A. Goldman, EM-DD: An improved multiple-instance learning technique, *Advances in Neural Information Processing Systems*, vol. 14, no. 2022, pp. 1073-1080, 2001.
[21] Y. J. Han, Q. Tao, and J. Wang, Avoiding false positive in multi-instance learning, *Advances in Neural Information Processing Systems*, vol. 23, pp. 1-8, 2010.
[22] M. Y. Kim and F. Torre, Gaussian processes multiple instance learning, in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 535-542.
[23] F. X. Li and C. Sminchisescu, Convex multiple-instance learning by estimating likelihood ratio, *Advances in Neural Information Processing Systems*, vol. 23, pp. 1360-1368, 2010.
[24] M. L. Zhang and Z. H. Zhou, Adapting RBF neural networks to multi-instance learning, *Neural Processing Letters*, vol. 23, no. 1, pp. 1-26, 2006.
[25] J. T. Kwok and P. M. Cheung, Marginalized multi-instance kernels, in *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, Morgan Kaufmann Publishers Inc, 2007, pp. 901-906.
[26] D. Zhang, F. Wang, L. Si, and T. Li, M^3IC: Maximum margin multiple instance clustering, in *Proceedings of the 21th International Joint Conference on Artifical intelligence*, Morgan Kaufmann Publishers Inc, 2009, pp. 1339-1344.
[27] C. H. Li, I. Gondra, and L. Liu, An efficient parallel neural network-based multi-instance learning algorithm, *The Journal of Supercomputing*, vol. 62, no. 2, pp. 724-740, 2012.
[28] H. Y. Wang, Q. Yang, and H. B. Zha, Adaptive p-posterior mixture-model kernels for multiple instance learning, in *Proceedings of the 25th International Conference on Machine Learning*, ACM, 2008, pp. 1136-1143.
[29] 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*, Morgan Kaufmann Publishers Inc, 2000, pp. 1119-1125.
[30] T. Deselaers and V. Ferrari, A conditional random field for multiple-instance learning, in *Proceedings of the 27th International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc, 2010, pp. 1119-1125.
[31] B. Babenko, N. Verma, P. Dollár, and S. J. Belongie, Multiple instance learning with manifold bags, in *Proceedings of the 28th International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc, 2011, pp. 81-88.
[32] D. Zhang, Y. Liu, L. Si, J. Zhang, and R. D. Lawrence, Multiple instance learning on structured data, *Advances in Neural Information Processing Systems*, vol. 24, pp. 145-153, 2011.
[33] L. X. Jiang, Z. H. Cai, D. H. Wang, and H. Zhang, Bayesian citation-KNN with distance weighting, *International Journal of Machine Learning and Cybernetics*, pp. 1-7, 2013.
[34] O. Maron, Learning from ambiguity, Ph.D. dissertation, Massachusetts Institute of Technology, USA, 1998.
[35] L. Zhang and B. Zhang, A geometrical representation of mcculloch-pitts neural model and its applications, *IEEE Transactions on Neural Networks*, vol. 10, no. 4, pp. 925-929, 1999.
[36] W. S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biology*, vol. 5, no. 4, pp. 115-133, 1943.
[37] P. N. Tan, *Introduction to Data Mining*, Pearson Education India, 2007.
[38] G. A. Edgar, *Measure, Topology, and Fractal Geometry*, Springer-Verlag, New York, 2008.
[39] Y. Lia, D. M. J. Taxa, R. P. W. Duina, and M. Looga, Multiple-instance learning as a classifier combining problem, *Pattern Recognition*, vol. 46, no. 3, pp. 865-874, 2013.
[40] Z. H. Zhou, M. L. Zhang, S. J. Huang, and Y. F. Li, Multi-instance multi-label learning, *Artificial Intelligence*, vol. 176, no. 1, pp. 2291-2320, 2012.

Shu Zhao received her PhD degree in computer science from Anhui University in 2007. She is currently an associate professor in the Department of Computer Science and Technology, Anhui University. From August 2010 to August 2011, she was a visiting scholar at Berkeley, University of California. Her current research interests include quotient space theory, granular computing, and machine learning.



Chen Rui is currently a master student of Department of Computer Science and Technology in Anhui University. He received his BEng degree in computer science from Chaohu University in 2011. His primary research interest is machine learning.

Yanping Zhang received her PhD degree in computer science from Anhui University in 2003. She is currently a professor in Department of Computer Science and Technology, Anhui University. Her primary research interests include computational intelligence, quotient space theory, artificial neural networks, and machine learning.