Tsinghua Science and Technology

Volume 19 | Issue 3 Article 7

2014

Multiple-Instance Learning with Instance Selection via Constructive Covering Algorithm

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

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

Huazhen Wei
Department of Computer Science and Technology and Key Lab of Intelligent Computing and Signal Processing, Anhui University, Hefei 230601, China.

Jie Tang
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China.

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

Recommended Citation

Yanping Zhang, Heng Zhang, Huazhen Wei et al. Multiple-Instance Learning with Instance Selection via Constructive Covering Algorithm. Tsinghua Science and Technology 2014, 19(03): 285-292.

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 07/11 pp285-292
Volume 19, Number 3, June 2014

Multiple-Instance Learning with Instance Selection via Constructive Covering Algorithm

Yanping Zhang, Heng Zhang, Huazhen Wei, Jie Tang, and Shu Zhao*

Abstract: Multiple-Instance Learning (MIL) is used to predict the unlabeled bags' label by learning the labeled positive training bags and negative training bags. Each bag is made up of several unlabeled instances. A bag is labeled positive if at least one of its instances is positive, otherwise negative. Existing multiple-instance learning methods with instance selection ignore the representative degree of the selected instances. For example, if an instance has many similar instances with the same label around it, the instance should be more representative than others. Based on this idea, in this paper, a multiple-instance learning with instance selection via constructive covering algorithm (MilCa) is proposed. In MilCa, we firstly use maximal *Hausdorff* to select some initial positive instances from positive bags, then use a Constructive Covering Algorithm (CCA) to restructure the structure of the original instances of negative bags. Then an inverse testing process is employed to exclude the false positive instances from positive bags and to select the high representative degree instances ordered by the number of covered instances from training bags. Finally, a similarity measure function is used to convert the training bag into a single sample and CCA is again used to classification for the converted samples. Experimental results on synthetic data and standard benchmark datasets demonstrate that MilCa can decrease the number of the selected instances and it is competitive with the state-of-the-art MIL algorithms.

Key words: multiple-instance learning; instance selection; constructive covering algorithm; maximal *Hausdorff*

1 Introduction

Multiple-Instance Learning (MIL) is a machine learning framework proposed by Dietterich et al.[1,2] for the prediction of drug molecule activity. MIL has become widely used in many applications, including drug-activity prediction, image classification[3,4], image retrieval[5], text categorization[6], face detection[7], etc.

At the high-level, existing MIL methods can be grouped into two main classes. The first class is to locate a region of interest in the instance space such that all true positive instances lie in their vicinity and all negative instances are far from them. The target concept can be found by DD[8], EM-DD[9], and GEM-DD[10] algorithms. However, a single target concept may be insufficient to represent all positive instances because the distribution of positive instances could be arbitrary.

The second class employs discriminative method to convert MIL into the Standard Supervised Learning (SSL). The conversion process is mainly divided into two kinds. The first is to label the instances with the corresponding bags' label directly. Then an SSL algorithm is employed to learn and the result of SSL is adapted at the same time. For example, MI-SVM[1] and

• Yanping Zhang, Heng Zhang, Huazhen Wei, and Shu Zhao are with the Department of Computer Science and Technology and Key Lab of Intelligent Computing and Signal Processing, Anhui University, Hefei 230601, China. E-mail: zhaoshuzs2002@hotmail.com.
• Jie Tang is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: jietang@tsinghua.edu.cn.
∗ To whom correspondence should be addressed.
Manuscript received: 2014-05-02; accepted: 2014-05-04

mi-SVM[6] were proposed by employing the Support Vector Machine (SVM) classifier. Citation K-Nearest Neighbour (KNN) and Bayesian-KNN[11] algorithms were proposed by extending the standard KNN[12]. The other kind is to embed each bag into an instance space based on a Representative Set of Instances (RSI) that are selected from the training bags, and then to learn a classifier in the instance space. This kind of methods mainly uses the RSI and are similarity function to map bags into an instance space, which includes KID[13], DD-SVM[5], MILIS[14], MILES[3], MILD_B[15], and MILDS[16].

However, most existing multiple-instance learning methods with instance selection ignore the representative degree of the selected instances. To address this problem, in this paper, we focus on the method of instance selection and propose a multi-instance learning with instance selection via constructive covering algorithm (MilCa), which is able to select the high representative degree instances and gain the representative degree of the selected instances. Synthetic data and five standard benchmark datasets are used in the experiment and the results demonstrate MilCa is competitive with state-of-the-art MIL algorithms.

2 Preliminaries

2.1 Notation

Let χ denotes the instance space. Given a data set $\text{DS} = \{(X_1, y_1), (X_2, y_2), \cdots, (X_m, y_m)\}$, where $X_i = \{\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \cdots, \boldsymbol{x}_{in_i}\} \subseteq \chi \ (i = 1, \cdots, m)$ is a set of instances called a bag and $y_i \in \{-1, +1\}$ is a class label. Here $\boldsymbol{x}_{ij} = [x_{ij1}, x_{ij2}, \cdots, x_{ijd}] \in \chi (j = 1, \cdots, n_i)$, where n_i denotes the number of instances in X_i and $x_{ijk} \, (k = 1, \cdots, d)$ is the value of $\boldsymbol{x}_{ij}$ at the k-th attribute. The training set is represented as: $X = \{X_1^+, \cdots, X_{m^+}^+, X_1^-, \cdots, X_{m^-}^-\}$, where X_i^+ and X_i^- denote that X_i is a positive or negative bag and $m^+ + m^- = m$. $X_{\text{ins}}^+ = \{x_i^+ | \, i = 1, 2, \cdots\}$ and $X_{\text{ins}}^- = \{X_i^- | \, i = 1, 2, \cdots\}$ denote the sets of all the instances in the positive bag and negative bag, respectively. The model created by CCA is a cover set: $C = \{c_i | c_i = (\text{center}_i, r_i, \text{no}_i), i = 1, 2, \cdots\}$, center_i denotes the center of c_i, r_i denotes its radius, and no_i denotes the number of samples that are covered by c_i. $C^- \subseteq C$ denotes the cover set of X_{ins}^- and C_{ins}^- denotes the set of centers of C^-. $\text{RSI}^+ (\text{RSI}^-)$ denotes the set of selected representative instances from positive (negative) bags.

2.2 Multiple-instance learning

Definition 1 Given a bag X_i, X_i is a positive bag if at least one of its instances is positive; otherwise, X_i is a negative bag.

Theorem 1 Given two bags X_i and X_k $(i \neq k)$, $X_i \bigcap X_k = \varnothing$.

The goal of MIL is to learn a classifier based on instances $f(x) : x \to y, x \in \chi$ or a classifier based on bags $F(X) : X \to y$ that correctly predicts the unlabeled bag[1,2,17-24]. The framework of MIL is given in Fig. 1.

2.3 Constructive covering algorithm

Constructive Covering Algorithm (CCA) is a standard supervised learning algorithm proposed by Zhang and Zhang[25]. The main idea of CCA is to construct a set of sphere neighbors (a set of covers), and each sphere neighbor covers the samples with the same class label. This set of covers can be regarded as a classifier.

Given a set of samples: $K = \{(s_i, l_i) | i = 1, \cdots, n, l_i = 1, \cdots, t, s_i \in R^d\}$, where n and t represent the number of samples and the number of the classes. S_{ij} is the value of S_i at the j-th attribute. Let $S = \{S_1, \ S_2, \cdots, S_t\}$ and S_i contains all samples of i-th class, $i = 1, 2, \cdots, t$. We use a flag to determine whether a sample is covered or not. For a sample s, $\text{flag}(s) = 1$ means that s is covered, $\text{flag}(s) = 0$ means that s is not covered. Note that the bigger inner product makes the smaller distance between s and $s^{'}$ and the more similarity of s and $s^{'}$. The outline of CCA's training process is summarized in Algorithm 1. $< s, s^{'} >$ denotes the inner product of sample s and $s^{'}$.

After the training of CCA, we can obtain a cover set C and construct a three-layer feed forward neural network for classification. Each cover $c_i \ (c_i \in C)$ is used to make up a hidden node. For a hidden node c_i, the output function is defined as

$$\text{result}_i = \text{sign}(< t_s, \text{center}_i > - r_i) \cdot l_i \tag{1}$$

where "sign" is a sign function, $t_s (t_s \in R^d)$ denotes the

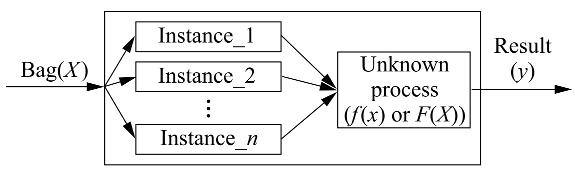

Fig. 1 The framework of MIL.

Algorithm 1 The outline of CCA's training process

Input: Training set K.
Output: Cover set C.
Begin
1: Initialization
Set $C = \varnothing, T(s) = (s, \sqrt[2]{R^2 - \|s\|^2}), R \geqslant \max\{\|s\| s \in K\}$.
2: Use the transform $T(s)$ to do a projection $s \rightarrow H^d$, where H^d is an d-dimensional sphere of an $(d+1)$-dimensional space.
3: For each S_i in S.
While $S_i \neq \varnothing$ do
randomly select a s in S_i and flag$(s) = 0$
$d_1 = \max\{< s, s' > | s \in S_i, s' \notin S_k, k = 1, \cdots, m, k \neq i\}$
$d_2 = \min\{< s, s' > | s, s' \in S_i, < s, s' >> d_1\}$
$r = (d_1 + d_2)/2$
create a cover c with s as its center (center $= s$) and r as its radius,
computer the number of samples(no) that covered by c in S_i, set
flag$(s) = 1$ and flag$(s'' = 1)$ where s'' is covered by c in S_i
$C = C \cup c$
End while
End For
End.

test input sample, the center$_i$ and l_i denote the center of c_i and its class label. A sample falls in c_i if result$_i$ > 0.

The output layer uses a "OR" operation to output the test result. The operation of "OR" denotes if a new sample is covered by several overlapped covers or is not covered by any covers, the output class is corresponding to the class of the nearest cover which has the largest inner product between t_s and center$_i$. Figure 2 gives the three-layer neural network[17, 25-27] constructed by CCA.

From the training and testing processes of CCA, we know that the CCA has the character of clustering because the samples with the same class label are classified into several covers. In this sense, the cover

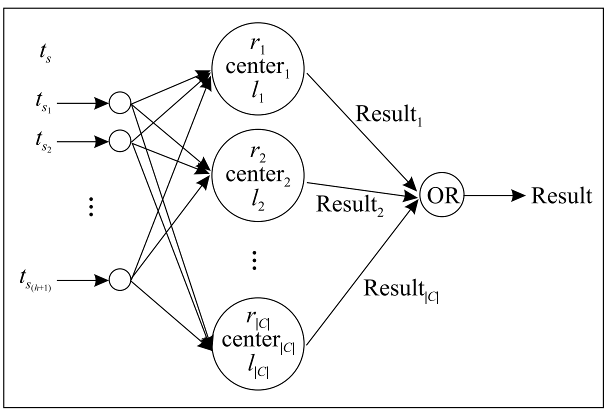

Fig. 2 A three-layer neutral constructed by CCA.

center can be regarded as the representative samples of the sphere neighbors. In the experiment, we update no$_i$ according to the number of samples that are covered by c_i. Because the more samples covered by c_i, the more representative the center of c_i.

Note: Clustering analysis is an important method in machine learning. CCA also has the character of clustering. The cover of CCA can be regarded as a cluster. Because the samples in the same cover have the same class, the cover center can be used as the clustering center. If there are many samples in a cluster, the representative degree of the center is higher than the others.

3 MilCa Algorithm

In this section, we present the novel MIL method called MilCa. MilCa aims to exclude the false positive instances and select the high representative degree instances from both positive and negative bags. It makes use of the clustering character of CCA for instance selection to build a more effective instance space.

3.1 Instance selection via constructive covering algorithm

This section mainly introduces the process of instance selection via CCA. Firstly, the definition of the inverse testing process is introduced.

Definition 2 Inverse testing process: The process is testing the instances in the *i*-th class bags by using the cover set of the *j*-th class ($j \neq i$) so that some representative instances can be selected.

As the false positive instances have great effects on the prediction of positive bags, it is necessary to exclude these instances. Firstly, maximal *Hausdorff*[2] and CCA are used to select some initial positive instances and restructure the structure of the original instances of negative bags. An inverse testing process is employed to exclude the false positive instances and to select the instances who have the high representative degree. Obviously, the number of instances in a cover is used to measure the representative degree of the representative instances.

3.1.1 Finding the initial positive instances

A distance metric between a positive bag and the set of negative bags is defined to select the initial positive instances. Here, the maximal *Hausdorff* distance is a natural distance metric for this purpose. Particularly, the distance between bag X_j^+ and X_{ins}^- is given by

$$d(X_j^+, X_{\text{ins}}^-) = \max_{x_k^- \in X_{\text{ins}}^-} \min_{x_j^+ \in X_j^+} \|x_j^+ - x_k^-\|^2 \tag{2}$$

where $\|x_j^+ - x_k^-\|^2$ is the Euclidean distance between the instances x_j^+ and x_k^-.

An instance can be obtained from the j-th positive bag (X_j^+) via Eq. (3).

$$x_j^+ = \arg d(X_j^+, X_{\text{ins}}^-) \tag{3}$$

m^+ initial instances (RSI$^+$) can be selected as the initial positive instances via Eq. (3) from all the training positive bags. These instances are more likely to belong to the true positive instances.

3.1.2 Excluding false positive instances

There are many negative instances around the instances of the positive bags and the instances of the positive bags are probably negative. These false positive instances in the positive bags will be excluded by CCA and an inverse testing process.

We label the instances in X_{ins}^- (RSI$^+$) with $-1(+1)$. CCA is used to construct the cover set of negative instances in X_{ins}^- and RSI$^+$. In order to exclude the false negative instances from the remaining instances in positive bags, an inverse testing process can be employed via C^- to select the instances in X_{ins}^+ that are not covered by C^-. If the instance in the positive bags is covered by the cover of C^-, the instance will more probably be a negative instance. Hence the instance can be excluded from the positive bags. If x_i^+ ($x_i^+ \in X_{\text{ins}}^+$) is covered by C^-, then $X_{\text{ins}}^- = x_i^+ \cup X_{\text{ins}}^-$. The RSI$^+$ is as follows:

$$\text{RSI}^+ = \{x_i^+ | x_i^+ \in X_{\text{ins}}^+, x_i^+ \notin X_{\text{ins}}^-\} \tag{4}$$

3.1.3 Selecting high representative degree instances

In this step, CCA is employed to construct the cover set with the instances in X_{ins}^- and RSI$^+$. All the instances in RSI$^+$ are used as a cover center to create the cover for themselves respectively and top m^+ instances are selected from RSI$^+$ in descending order with no$^+$ to prune RSI$^+$.

At the same time, the instances of C_{ins}^- are regarded as the set of representative negative instances (RSI$^-$). If $|C_{\text{ins}}^-| > m^-$, C_{ins}^- is sorted from large to small according to no_i^- and top m^- instances of C_{ins}^- are selected as RSI$^-$. Because the larger no_i^- of x_i^- makes x_i^- more representative, hence the RSI$^-$ is as follows:

$$\text{RSI}^- = \{x_i^- | x_i^- \in C_{\text{ins}}^-\} \tag{5}$$

Then the set of high degree representative instances is RSI (RSI = RSI$^+ \cup$ RSI$^-$). So $|\text{RSI}^+| = m^+$, $|\text{RSI}^-| = m'^-$ ($m'^- = \min(|C^-|, m^-)$). The outline of the proposed instance selection with CCA can be found in Algorithm 2.

The efficiency of the algorithm is discussed below. The computational complexity of MilCa is $O(E^2)$, where E is the number of the instances in all positive and negative bags. In a word, the time complexity of MilCa is equal to CCA. The complexity of MilCa is mainly determined by Step 1 and Step 3 in Algorithm 2. The computational complexity of other steps is less than $O(E^2)$.

3.2 Transformation and classification

A similarity function will be used to transform a bag to a single sample so that the ML problem becomes a SSL problem. For RSI $= \{x_1^+, x_2^+, \cdots, x_{m^+}^+\} \cup \{x_1^-, x_2^-, \cdots, x_{m'^-}^-\}$ and the numbers of instances in the corresponding covers are no_i^+ and no_j^-, $i = 1, \cdots, m^+$, $j = 1, \cdots, m^-$, respectively. The similarity between x_k ($x_k \in$ RSI) and the i-th bag X_i is

$$s(x_k, X_i) = \begin{cases} \dfrac{\text{no}_k^+}{\sum_{p=1}^{m^+} \text{no}_p^+} \cdot \min\limits_{x_k \in \text{RSI}^+} \exp\left(\dfrac{d(x_k^+, x_{ij})^2}{2\sigma^2}\right), \\ \dfrac{\text{no}_k^-}{\sum_{p=1}^{m^-} \text{no}_p^-} \cdot \min\limits_{x_k \in \text{RSI}^-} \exp\left(\dfrac{d(x_k^-, x_{ij})^2}{2\sigma^2}\right) \end{cases} \tag{6}$$

Equation (6) calculates the minimum distance between x_k and X_i, $x_{ij} \in X_i$. Then, we employ the exponential function and the parameter of σ to adjust

---

Algorithm 2 The outline of instance selection in MilCa

Input: Training bags $X = \{X_1^+, \cdots, X_{m^+}^+, X_1^-, \cdots, X_{m^-}^-\}$.
Output: RSI.

Begin
1: Label all instances of positive(negative)training bags with $+1(-1)$.
2: Select m^+ initial instances (RSI$^+$) from positive bags via Eq. (3).
3: Obtain a negative cover set C^- via Algorithm 1, the instances in RSI$^+$ and X_{ins}^-
4: Use C^- to do an inverse testing process and select the instances from positive bags that are not covered by C^- via Eq. (4)
5: Construct a cover set via Algorithm 1, the instances in RSI$^+$ and X_{ins}^-. Then obtain the new RSI$^+$ with m^+ instances and RSI$^-$ with m'^- instances according to no$^+$ and no$^-$ via Eq. (4) and Eq. (5)
6: Form the high degree representative set of instances, RSI (RSI=RSI$^+ \cup$ RSI$^-$)
End.

the similarity measure function. If $x_k \in \mathrm{RSI}^+$, then a positive bag should be similar to the instance x_k highly. Otherwise the positive bag has a low similarity with the instance. The number of instances no_k is used to measure the weight of x_k.

Then an embedding function $\boldsymbol{\phi}$ is defined, which converts a bag X_i to a $(m^+ + m'^-)$-dimensional sample:

$$\boldsymbol{\phi}(X_i) = [s(x_1^+, X_i), \cdots, s(x_{m^+}^+, X_i), s(x_1^-, X_i), \cdots, s(x_{m'^-}^+, X_i)]^{\mathrm{T}} \quad (7)$$

The converted single sample is also labeled as the label of X_i and the MIL problem can be converted into the SSL problem. For classification, Algorithm 1 can be employed to train the converted samples.

4 Experiments

In this section, the proposed method (MilCa) is evaluated in both synthetic data and benchmark datasets.

4.1 Datesets

The synthetic data is used to describe the effect of false positive instances and to explain the process of instance selection. The benchmark datasets are used to compare with other MIL methods. The benchmark datasets used in this paper can be found at http://www.cs.waikato.ac.nz/~eibe/multi_instance/.

Synthetic data: This data is described in Fig. 3 which each instance is generated by one of the five two-dimensional probability distributions: $N_1 \sim N([5,5]^{\mathrm{T}}, I)$, $N_2 \sim N([5,-5]^{\mathrm{T}}, I)$, $N_3 \sim N([-5,5]^{\mathrm{T}}, I)$, $N_4 \sim N([-5,-5]^{\mathrm{T}}, I)$, $N_5 \sim N([0,0]^{\mathrm{T}}, I)$, where $N([5,5]^{\mathrm{T}}, I)$ denotes the normal distribution with mean $[5,5]^{\mathrm{T}}$ and identity covariance matrix. It contains 6 positive bags and 6 negative bags. Each bag comprises at most 8 instances. A bag is labeled positive if it contains instances from at least two different distributions among N_1, N_2, and N_3. Otherwise, the bag is negative.

Musk1 and Musk2: The task of Musk1 and Musk2 is to predict the drug's activity. The positive bags are active and the remaining negative bags are inactive.

Elephant, Fox, and Tiger: The task of the three datasets is to estimate whether the images contain elephants, tigers, and foxes or not. In these three datasets, each image is considered as a bag, and the interest region of the image as an instance.

More details of the benchmark datasets can be found in Table 1.

Table 1 Information of the MIL benchmark datasets.

Dataset	Bags (pos./neg.)	Total bags	Instances in the bag Total	Instances in the bag Average	Dim.
Musk1	47/45	92	476	5.17	166
Musk2	39/63	102	6598	64.69	166
Elephant	100/100	200	1391	6.96	230
Fox	100/100	200	1320	6.60	230
Tiger	100/100	200	1220	6.10	230

4.2 Experiment setup

In the experiment, the parameter σ of Eq. (6) needs to be adjusted. The values of σ are selected by using 10-fold Cross Validation (CV) from the sets of linespace $(0.6\mu, 1.4\mu, 10)$, where μ is the average distance between each pair of instances in the RSI, and linespace (a, b, n) denotes the set n linearly spaced numbers between and including a and b.

4.3 Experimental and analysis

The synthetic data is used to describe the effect of false positive instances and to introduce the process of instance selection. Then, the five benchmark datasets are used to compare with other MIL methods.

4.3.1 Synthetic data

Figure 3 depicts all the instances on two-dimensional plane. Note that positive and negative instances are represented by circles and equilateral triangles, respectively. Specially, the positive bags are labeled with 1 to 6 and the negative bags are labeled with 7 to 12 in turn. The instances in the same bag are labeled with the bags' label.

The results of selected instances from negative bags in Fig. 4 are the centers of the negative covers I

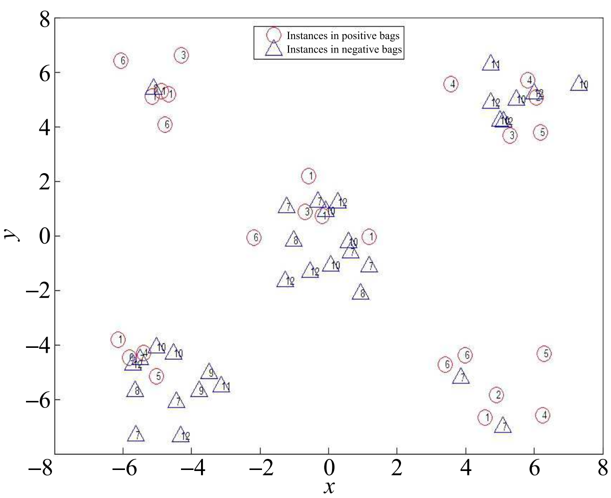

Fig. 3 Raw data.

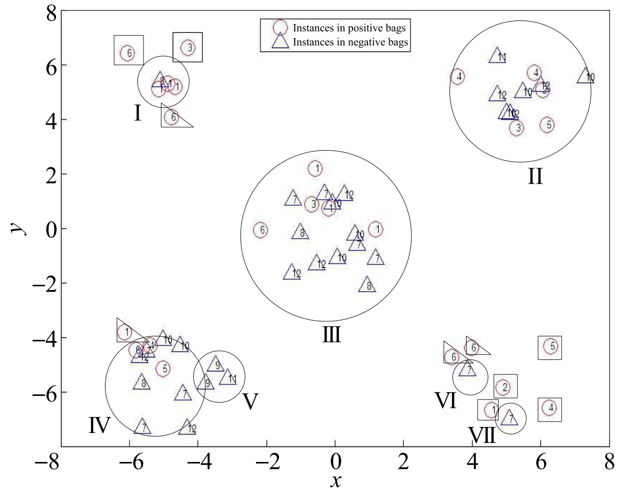

Fig. 4 Result of CCA and the process of instance selection.

to VII. As Fig. 3 depicts, there are many instances in the positive bags distributed around the negative instances. It is difficult to deal with the instance in the positive bag directly. Specifically, if an instance in the positive bag is a true positive instance, there will be less negative instances around the true positive instance. As the number of instances covered by cover III is the biggest one, it can convince that the center of cover III is more important than others. Then the number of instances covered by a cover can be used to measure the representative degree of the cover center. It is important to select the high representative degree instances from the training bags.

At the same time, the labels of rectangles and right triangle are added to the selected instances from the positive bags in Fig. 4. The instances labeled by rectangles represent the initial positive instances via Eq. (3). The labels of right triangles are added to the instances in positive bags not covered by the negative cover I to VII. These instances are mainly distributed around the upper left corner and the lower right corner. There are less negative instances in these two areas. It can convince that the selected instances in positive bags are more likely to the true positive instances.

4.3.2 Benchmark datasets

MilCa is compared with other methods via ten times 10-fold CV (i.e., we repeat 10-fold CV for ten times with different random data partition), with the exception of MILIS, which is over 15 times. The details of experimental results are shown in Table 2. The best performances on each dataset are in bold. The

Table 2 Accuracy (%) on benchmark datasets.

Algorithm	Musk1	Musk2	Elephant	Fox	Tiger
MilCa	88.7	90.0	**85.4**	61.4	81.6
MILDS[16]	**90.9**	86.1	84.8	**64.3**	81.6
MILD_B[15]	88.3	86.8	82.9	55.0	81.6
MILIS[14]	88.6	91.1	*n/a*	*n/a*	*n/a*
MILES[3]	83.3	87.7	84.1	63.0	80.7
DD-SVM[11]	85.8	**91.3**	83.5	56.6	77.2
EM-DD[9]	84.8	84.9	84.9	56.1	72.1
Bag-KI-SVM[18]	88.0	82.0	84.5	60.5	**85.0**
MIGraph[19]	90.0	90.0	85.1	61.2	81.9
MI-SVM[6]	77.9	84.3	81.4	59.4	84.0
MILD_I[15]	89.9	88.7	83.2	49.1	73.4
Average	86.9	86.4	83.7	58.3	79.1

result demonstrates that MilCa is competitive with and often better than the most state-of-the-art MIL algorithms. The first six algorithms are based on instance selection.

The average accuracies of other algorithms on different datasets are listed in the last row. The accuracy of MilCa is higher than the average accuracy on the five benchmark datasets. It increases about 2.54% than the average accuracy on the five datasets.

Especially, in Elephant dataset, MilCa can achieve the highest accuracy among the algorithms in Table 2. MilCa can also achieve better accuracy than most of the algorithms which based on instance selection on the benchmark datasets. Although MilCa has a lower accuracy on Tiger dataset, MilCa can achieve the highest accuracy among the first six algorithms based on instance selection and the average number of instances in RSI extracted from Tiger dataset is also less than other methods in Table 3 except MILD_B.

The average numbers of instances in RSI are given in Table 3. In Musk1 and Fox datasets, MILDS can achieve the best performance. However, the average number of instances in RSI in MILDs is more than our method's and MILDS does not achieve higher accuracy in other three datasets than our method.

Table 3 The average number of instances in RSI.

Algorithm	Musk1	Musk2	Elephant	Fox	Tiger
MilCa	61.1	97.4	133.3	133.8	124.7
MILDS[16]	75.0	92.0	169.4	180.0	139.2
MILD_B[15]	**42.4**	**35.2**	**90.0**	**90.0**	**90.0**
MILIS[14]	83.0	92.0	180.0	180.0	180.0
MILES[3]	429.4	5943.8	1251.9	1188.0	1098.0
DD-SVM[11]	83.0	92.0	180.0	180.0	180.0

The average number of instances in MILD_B is the smallest, because MILD_B only selects one instance from each positive training bag. However, the results are also poorer than many other methods. The average number of instances of our method is only more than MILD_B's and less than other methods except on the dataset Musk2. MilCa decreases average number of selected instances by about 25.6%-30.7%, as compared to MILD_B and MILIS.

The number of the selected instances in MILES is the largest among three methods because its RSI includes all instances in training bags and the performance of MILES is not better than our method except on dataset Fox. In a word, MilCa can achieve better results while decrease the number of selected instances.

5 Conclusions

The false positive instances in positive bags have a negative impact on the MIL problem. It is necessary to exclude the false positive instances and select the representative true positive instances. In this paper, a novel MIL method named MilCa, is proposed which tackles multiple-instance problems by using the clustering of constructive covering algorithm and an inverse testing process. The instance selection approach can select the high representative degree instances from the training bags effectively. Moreover, it can use the number of instances in a cover to measure the representative degree of the selected instances. Experiments show that MilCa is competitive with many existing multi-instance learning algorithms and decreases the number of selected instances.

However, some false positive instances may still exist in RSI. The effects of false positive instances of RSI would be decreased in future work and our research will be extended to multi-class problem and multi-instance multi-label learning problem.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (No. 61175046), the Provincial Natural Science Research Program of Higher Education Institutions of Anhui Province (No. KJ2013A016), the Outstanding Young Talents in Higher Education Institutions of Anhui Province (No. 2011SQRL146), and the Recruitment Project of Anhui University for Academic and Technology Leader.

References

[1] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez, Solving the multiple instance problem with axis-parallel rectangles, *Artificial Intelligence,* vol. 89, pp. 31-71, 1997.

[2] A. Zafra, M. Pechenizkiy, and S. Ventura, ReliefF-MI: An extension of ReliefF to multiple instance learning, *Neurocomputing,* vol. 75, pp. 210-218, 2012.

[3] Y. X. Chen, J. B. Bi, and J. Z. Wang, MILES: Multiple-instance learning via embedded instance selection, *IEEE Transaction Pattern Analysis and Machine Intelligence,* vol. 28, pp. 1931-1947, 2006.

[4] X. F. Song, L. C. Jiao, S. Y. Yang, X. R. Zhang, and F. H. Shang, Sparse coding and classifier ensemble based multi-instance learning for image categorization, *Signal Processing,* vol. 93, pp. 1-11, 2013.

[5] Y. X. Chen and J. Z. Wang, Image categorization by learning and reasoning with regions, *Journal of Machine Learning Research,* vol. 5, pp. 913-939, 2004.

[6] S. Andrews, I. Tsochantaridis, and T. Hofmann, Support vector machines for multiple-instance learning, in *Advance in Neutral Information Processing System 15,* 2003, pp. 561-568.

[7] P. Viola, J. Platt, and C. Zhang, Multiple instance boosting for object detection, in *Advance in Neutral Information Processing System 18,* 2006, pp.1419-1426.

[8] O. Maron and T. Lozano-Perez, A framework for multiple-instance learning, in *Advance in Neutral Information Processing System 10,* 1998, pp. 570-576.

[9] Q. Zhang and S. A. Goldman, EM-DD: An improved multi-instance learning technique, in *Advance in Neutral Information Processing System 14,* 2002, pp. 1073-1080.

[10] R. Rahmani, S. A. Goldman, H. Zhang, S. R. Cholleti, and J. E. Fritts, Localized content based image retrieval, *IEEE Transaction Pattern Analysis and Machine Intelligence,* vol. 30, pp. 1902-2002, 2008.

[11] J. Wang and J. D. Zucker, Solving the multiple-instance problem: A lazy learning approach, in *Processing of the 17th International Conference on Machine Learning,* Morgan Kaufmann Publishers Inc, 2000, pp. 1119-1125.

[12] P. N. Tan, *Introduction to Data Mining*. Pearson Education India, 2007.

[13] G. Q. Liu, J. X. Wu, and Z. H. Zhou, Key instance detection in multi-instance learning, *Journal of Machine Learning Research Proceedings Track,* pp. 253-268, 2012.

[14] Z. Fu, A. Robles-Kelly, and J. Zhou, MILIS: Multiple instance learning with instance selection, *IEEE Transaction Pattern Analysis and Machine Intelligence,* vol. 33, pp. 958-977, 2011.

[15] W. J. Li and D.Y. Yeung, MILD: Multiple-instance learning via disambiguation, *IEEE Trans. on Knowledge and Data Engineer,* vol. 22, pp. 76-89, 2010.

[16] E. Aykut and E. Erkut, Multiple-instance learning with instance selection via dominant sets, *Similarity-Based Pattern Recognition,* vol. 7005, pp. 171-191, 2011.

[17] Y. Li, D. M. J. Tax, R. P. W. Duin, and M. Loog, Multiple-

instance learning as a classifier combining problem, *Pattern Recognition,* vol. 46, pp. 865-874, 2013.
[18] Y. F. Li, J. T. Kwok, I. W. Tsang, and Z. H. Zhou, A convex method for locating regions of interest with multi-instance learning, *ECML PKDD,* vol. 5782, pp.15-30, 2009.
[19] Z. H. Zhou, Y. Y. Sun, and Y. F. Li, Multi-instance learning by treating instance as Non-I.I.D samples, in *Processing of the 26th International Conference on Machine Learning,* pp. 1249-1256, 2009.
[20] A. Zafra, M. Pechenizkiy, and S. Ventura, A condition random field for multiple-instance learning, in *Processing of the 27th International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc, 2010, pp. 287-294.
[21] S. Zhao, R. Chen, and Y. P. Zhang, MICkNN: Multi-instance covering kNN algorithm, *Tsinghua Science and Technology,* vol. 18, pp. 360-368, 2013.
[22] Z. H. Zhou, M. L. Zhang, S. J. Huang, and Y. F. Li, Multi-instance multi-label learning, *Artificial Intelligence,* vol. 176, pp. 2291-2320, 2012.
[23] B. Babenko, N. Verma, P. Dollar, and S. J. Belongie, Multiple instance learning with mainfold bags, in *Processing of the 28th International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc, 2011, pp. 81-88.
[24] 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,* vol. 5, no. 2, pp.1-7, 2013.
[25] L. Zhang and B. Zhang, A geometrical-representation McCulloch-Neural model and its application, *IEEE Transactions on Neural Networks,* vol. 10, pp. 925-929, 1999.
[26] K. R. Md and E. Tsutomu, Recurrent neutral network classifier for three layer conceptual network and performance evaluation, *Journal of Computers,* vol. 5, pp. 40-48, 2010.
[27] L. Mo and Z. Xie, An improved BP neural network based on IPSO and its application, *Journal of Computers,* vol. 8, pp. 1267-1272, 2013.
[28] Y. P. Zhang, Y. H. Wang, and S. Zhao, A novel distributed machine learning method for classification: Parallel covering algorithm, in *7th International Conference of Rough Set and Knowledge Technology,* 2012, pp. 185-193.

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

Heng Zhang is currently a master student at Department of Computer Science and Technology in Anhui University. He received his BEng degree in computer science from Anhui University in 2012. His current research interest is machine learning.

Huazhen Wei is currently an undergraduate student at Department of Computer Science and Technology in Anhui University. Her current research interest is machine learning.

Jie Tang received his PhD degree in computer science and technology from Tsinghua University in 2006. He is now an associate professor in Tsinghua University. His current research interests are social network mining, social influence analysis, and data mining.

Shu Zhao received her PhD degree in computer science from Anhui University in 2007. She is now an associate professor at the Department of Computer Science and Technology, Anhui University. Her current research interests include quotient space theory, granular computing, and machine learning.