Appl Intell (2015) 42:537–543
DOI 10.1007/s10489-014-0608-z

An MIMLSVM algorithm based on ECC

Cunhe Li · Yanli Zhang · Lei Lu

Published online: 21 November 2014

Abstract In the multi-instance multi-label learning framework, an example is described by multiple instances and associated with multiple class labels at the same time. An idea of tackling with multi-instance multi-label problems is to identify its equivalence in the traditional supervised learning framework. However, some useful information such as the correlation between labels may be lost in the process of degeneration, which will influence the classification performance. In E-MIMLSVM$^+$ algorithm, multi-task learning techniques are utilized to incorporate label correlations, while it is time consuming as well as memory consuming. Therefore, we propose an improved algorithm. In our algorithm, the classifier chains method is applied in E-MIMLSVM$^+$ to incorporate label correlations instead of multi-task learning techniques. The experimental results show that the proposed algorithm can reduce time complexity and improve the predictive performance.

Keywords Support vector machine · Multi-instance multi-label · Classifier chains · Label correlations

1 Introduction

In recent years, multi-instance multi-label learning (MIML) has attracted a lot of attention in the machine learning community. In MIML, an example is described by several instances and associated with a set of labels [1]. Many real-world problems can be formalized under the MIML framework, such as text categorization, scene classification, web page classification and gene sequence encoding. For example, in text categorization, each document is usually comprised of several paragraphs of sections, each can be regarded as an instance, while the document can be assigned to a set of predefined topics; multiple links can be extracted from a web page where each link is described by an instance, and thus the web page can be represented by a set of instances meanwhile the web page may belong to many classes, such as news page, sports page, soccer page, etc [2].

The traditional supervised learning (SISL) can be viewed as a degenerated version of MIML where each example is represented by an instance and associated with a class label. Hence, one way to solve MIML problem is to identify its equivalence in SISL via problem degeneration. Although this kind of degeneration strategy is feasible, the performance of the resultant algorithm may be affected by the loss of information during the degenerative process [3].

Support Vector Machine (SVM) [4] has been extensively applied in different areas, such as incremental learning [5], multi-class classification [6], and semi-supervised classification [7]. The traditional SVM can only solve single-instance single-label learning problems. In comparison, E-MIMLSVM$^+$ is a SVM-based algorithm which can solve multi-instance multi-label learning problems. Besides, E-MIMLSVM$^+$ can use the correlations between labels to improve the classification performance [8]. E-MIMLSVM$^+$ is the improved version of MIMLSVM$^+$. In MIMLSVM$^+$, a simple degeneration strategy is firstly employed. It decomposes the learning of multiple labels into a series of binary classification tasks. The algorithm constructs an SVM for

C. Li (✉) · Y. Zhang · L. Lu
College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266580, China
e-mail: licunhe@upc.edu.cn

Y. Zhang
e-mail: civilization2335@163.com

L. Lu
e-mail: leokingofdevil@163.com

each label. Since the kernel function employed here is based on several instances instead of a single feature vector, the well-known multi-instance kernel [9] is adopted. MIMLSVM$^+$ decomposes the multi-label problem into a series of independent binary learning tasks. In this way, the correlation between labels is neglected. To overcome this drawback, E-MIMLSVM$^+$ incorporates the label correlations by utilizing multitask learning techniques which consider the SVM training of each label as a task [10, 11]. The kernel-based multitask learning framework [12] is employed, since MIMLSVM$^+$ is a support vector machine algorithm. While, E-MIMLSVM$^+$ consumes more time and memory than MIMLSVM$^+$ since multitask learning simultaneously will result in many more instances involved in the optimization procedure [8].

In this paper, we propose a new SVM approach to MIML named ECC-MIMLSVM$^+$. Briefly, ECC-MIMLSVM$^+$ employs a degeneration strategy to decompose multiple labels learning into a series of binary classification tasks. Subsequently, we put forward a novel classifier chains method which uses the information of label correlation and meanwhile maintains acceptable computational complexity.

The rest of this paper is organized as follows. Section 2 gives the formal definition of MIML and reviews the related works. Section 3 presents ECC-MIMLSVM$^+$. Section 4 reports experimental results on two real-world MIML data sets. Finally, Section 5 concludes the paper.

2 Related works

In the MIML framework, a learning algorithm typically takes a set of labeled train examples $L = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ as input, where $X_i \subseteq X$ is a bag of instances $\left\{x_1^{(i)}, x_2^{(i)}, \ldots, x_{n_i}^{(i)}\right\}, x_j^{(i)} \in x(j = 1, 2, \ldots, n_i)$ and $Y_i \subseteq Y$ is a set of labels $\{y_1^{(i)}, y_2^{(i)}, \ldots, y_{l_i}^{(i)}\} y_k^{(i)} \in Y(k = 1, 2, \ldots, l_i)$ associated with X_i. The task of MIML is to learn a function $f : 2^x \to 2^y$ from a set of MIML training examples L. The MIML framework can be viewed as a generalization from the learning frameworks of multi-instance learning [13], multi-label learning [14, 15], and traditional supervised learning.

Multi-instance learning [13] or multi-instance single-label learning (MISL) was proposed by Dietterich et al. The goal of MISL is to learn a function $f_{MIL} : 2^x \to \{-1, +1\}$ from a set of MISL training examples $\{(X_1, y_1), (X_2, y_2), \ldots, (X_m, y_m)\}$ where $X_i \subseteq X$ is a bag of instances $\left\{x_1^{(i)}, x_2^{(i)}, \ldots, x_{n_i}^{(i)}\right\}, x_j^{(i)} \in x(j = 1, 2, \ldots, n_i)$ and $y_i \in \{-1, +1\}$ is the binary label of Xi.

Multi-label learning or single-instance multi-label learning (SIML) is originated from the investigation of text categorization problems. The goal of SIML is to learn a function $f_{MLL} \to x \to 2^y$ from $\{(x_1, Y_1), (x_2, Y_2), \ldots, (x_m, Y_m)\}$, where $x_i \in X$ is an instance and $Y_i \subseteq Y$ is a set of labels $\left\{y_1^{(i)}, y_2^{(i)}, \ldots, y_{l_i}^{(i)}\right\} y_k^{(i)} \in Y(k = 1, 2, \ldots, l_i)$ associated with x_i.

Zhi-Hua Zhou and Min-Ling Zhang proposed two MIML algorithms named MIMLBOOST and MIMLSVM [1]. These two algorithms are typically MIML algorithms that based on degeneration strategy. They transformed MIML into traditional supervised learning using MISL and SIML respectively as the bridge. However, useful information encoded between instances and labels may be lost during the degeneration process, which may determine the accuracy of classification.

Zhi-Hua Zhou and Min-Ling Zhang also proposed a direct MIML algorithm named D-MIMLSVM [16] and a maximum margin MIML algorithm named M^{3}MIML[17]. They are both MIML algorithms that based on regularization. In the theory of linear algebra, regularization refers to the question that ill-posed problem is usually caused by a defined set of linear algebra, and this group has a lot of equations usually derived from the condition number of ill-posed inverse problems. D-MIMLSVM assumes that the labels associated with the same example have some relatedness, and the performance of classifying the bags depends on the loss between the labels and the predictions on the bags as well as on the constituent instances [2]. However, it can only deal with moderate training set sizes because of the large optimization problem [16]. M^3 MIML assumes a linear model for each class, where the output on one class is determined by the maximum prediction of all instances with respect to the corresponding linear model. And then, outputs on all possible classes are combined to define the margin of MIML example over the classification system.

Ying-Xin Li and Shuiwang Ji [8] designed a MIML algorithm for large-scale learning problem named MIMLSVM$^+$, and its extended algorithm is named E-MIMLSVM$^+$. MIMLSVM$^+$ simply employs a degeneration strategy which decomposes the learning of multiple labels into a series of binary classification tasks. Different from traditional SVM, the kernel function used in MIMLSVM$^+$ is based on a bag of instances instead of a single feature vector. Theoretically, any kernel defined on the set of instances [18] can be used to compute the kernel function. The famous multi-instance kernel [9] is adopted in MIMLSVM$^+$. As the degeneration strategy of MIMLSVM$^+$ may lose the information of label correlations, E-MIMLSVM$^+$ incorporates label correlations by utilizing multitask learning techniques to extend MIMLSVM$^+$ [10]. For a given label $y \in Y$ suppose its classification function is

$$f_y(X) = \langle w_y, \varphi(X) \rangle + b = \langle (w_0 + v_y), \varphi(X) \rangle + b \qquad (1)$$

Where w_0 is used to reflect the commonalities shared by different learning tasks, v_y is the task-specific model parameter which used to measure the dissimilarity between task y and other tasks, $\varphi(X)$ is a mapping function, and b is the offset [8]. Here, multi-instance multitask kernel is used as follows, which bridges the multi-instance kernel and the multitask kernel.

$$K_{ty}(X_i, X_j) = \left(\frac{1}{\mu} + \delta(t = y)\right) K_{MI} X_i, X_j) \quad (t, y \in Y) \tag{2}$$

Where t and y denote two different tasks respectively, and $\delta(t = y) = 1$ if $t = y$, otherwise $\delta(t = y) = 0$.

In order to avoid this situation in which all the models f_y are forced to be close to a common parameter denoted by w_0, a clustering process to partition the labels into some subgroups based on the correlations between labels is considered before its training. Although E-MIMLSVM$^+$ achieves superior performance over other algorithms, it is time consuming and needs more memory than other algorithms.

Classifier chains (CC) method which is based on the binary relevance (BR) method, overcomes the disadvantages of BR and achieves higher predictive performance, but still retains its important advantages, most importantly is low time complexity. Still like BR, CC's models can both be parallelized and serialized, in addition, there is only a single binary problem in memory at any time, which own an obvious advantage over other methods based on a single large model [19].

3 Proposed method

3.1 The MIMLSVM$^+$ method

Suppose n is the number of training examples. $y \in y$ is a label, X_i is a bag of instances in the training set. For each label y, let $\varphi(X_i, y)$ be the indicator function defined as: $\varphi(X_i, y) = 1$ if y is in Y_i which is corresponding to X_i, and $\varphi(X_i, y) = -1$ otherwise. Hence the SVM classification model involves the following optimization problem:

$$\min_{w_y, b_y, \xi_{iy}} \frac{1}{2} \left\| w_y \right\|^2 + C \sum_{i=1}^{n} \xi_{iy} \tau_{iy}$$

$$\text{s.t.} : \phi(X_i, y) \left(\left\langle w_y, \varphi(X_i) \right\rangle + b_y \right) \geq 1 - \xi_{iy} \tag{3}$$

$$\xi_{iy} \geq 0 \, (i = 1, 2, \ldots, n),$$

Where $\langle \cdot, \cdot \rangle$ denotes the inner product. $\varphi(X_i)$ is the function that maps the bag of instances X_i into a higher dimensional space H. w_y and b_y are the parameters of a linear discriminant function in H. ξ_{iy} is the nonnegative slack variable in the constraints to permit some training bags to be misclassified. $\left\| w_y \right\|^2$ reflects the complexity of the model [4]. C is a parameter to balance the model complexity and the accumulative losses of the training bags. τ_{iy} is the amplification coefficient of the loss ξ_{iy} to handle the class imbalance problem and defined as

$$\tau_{iy} = \frac{1 + \phi(X_i, y)}{2} R_y + \frac{1 - \phi(X_i, y)}{2}, \tag{4}$$

Where R_y is the imbalance level of label y, and it can be evaluated by the number of negative bags divided by the number of positive bags in the training set.

Kernel function is very important in support vector machine and needs to be predefined [4]. Different kernel function will lead to different support vector machine. Here the well-known multi-instance kernel is employed and defined as follows:

$$K_{MI}(X_i, X_j) = \frac{1}{n_i n_j} \sum\nolimits_{(x_{is\_0}, x_{is\_1}) \in X_i} \times \sum\nolimits_{(x_{jz\_0}, x_{jz\_1}) \in X_j^e} {}^{-\gamma_1 \left\| x_{is\_0} - x_{jz\_0} \right\|^2 - \gamma_2 \left\| x_{is\_1} - x_{jz\_1} \right\|^2} \tag{5}$$

Where n_i and n_j denotes the number of instances in the bags X_i and X_j respectively. $\left\| x_{is\_0} - x_{jz\_0} \right\|^2$ is employed to measure the similarity of visual features (low-level features and represented by a local descriptor) between two instances. $\left\| x_{is\_1} - x_{jz\_1} \right\|^2$ is employed to measure the spatial distance (the distance between points, lines and surfaces in three dimensional space) between two instances. γ_1 and γ_2 are the different weights to combine the visual and spatial information.

For a given label y and an example X, the resulting classification model can be defined as

$$\begin{aligned} f_y(X) &= \langle w_Y, \varphi(X) \rangle + b_y \\ &= \sum_{i=1}^{n} a_{iy} \phi(X_i, y) K_{MI}(X_i, X) + b_y \end{aligned} \tag{6}$$

3.2 The improved method: ECC-MIMLSVM$^+$

In order to incorporate the label correlations, E-MIMLSVM$^+$ introduces multitask learning techniques, which make the algorithm consume more time and memory. In this paper we utilize an advanced classifier chains method, the Ensembles of Classifier Chains (ECC) [19], to improve MIMLSVM$^+$. Classifier chains method is the well known binary relevance method for multi-label classification, which considers each label as an independent binary problem.

Y=ECC-MIMLSVM$^+$(S,X,N,O)
Input : S – the training set
X – the test bag of instances
N – the number of ensemble iterations
O – the global label set
Output :Y – the set predicted labels of X

1) Ranking the labels in label set O randomly for N times, then we can get N different sets $L_i(\mathrm{i}=1,2,\ldots,N)$.
2) For training set $S=\{(X_i,Y_i)\}(i=1,2,\cdots,n)$, calculate the multi-instance kernel matrix $\left[K_{MI}(X_i,X_j)\right](i,j=1,2,\ldots,n)$.
3) For each ordered label set $L_k(\mathrm{k}=1,2,\ldots,N)$
   (a) For each label $y_i \in O(\mathrm{i}=1,2,\ldots,L)$ (where L is the number of labels on label set O)
   do $y_i^{'} \leftarrow (y_i,y_i,\ldots,y_i)_d^T$
   $S_i \leftarrow \{\ \}$
   for each $(X_j,Y_j)\in S$
   do $X_j^{'} \leftarrow \left[X_j,y_1^{'},y_2^{'},\ldots,y_{i-1}^{'}\right]$
   $S_i \leftarrow S_i \cup \left(X_j^{'},\phi(X_j,y_i)\right)$
   end for
   end for
   (b) Train an SVM $f_{\mathrm{ki}}^{'}=SVMTrain(S_i)$ based on $\left[K_{MI}(X_i,X_j)\right]$,using The formulation (3)
   end for
4) For a test bag X
   (a) for i=1,….,N
   $\hat{Y}_{\mathrm{i}} \leftarrow \{\ \}$
   for j=1,…..,L
   $\hat{y}_{\mathrm{ij}} \leftarrow f_{ij}(X)$
   $\hat{Y}_{\mathrm{i}} \leftarrow \hat{Y}_{\mathrm{i}} \cup \hat{y}_{ij}$
   end for
   end for
   (b) calculate $\hat{w}$ using $\hat{Y}_{\mathrm{i}}$ $(i=1,2,\ldots,N)$
   (c) obtain the label set of X using the formulation(8):

$$Y=\left\{y_j \middle| \hat{y}_{ij}=1\right\}$$

Fig. 1 ECC-MIMLSVM$^+$ algorithm

As the ECC is originally designed for SIML problems, we improve it to adapt to our MIML problems. Therefore, we first extend each label into a column vector, for example, we let $y_k^{'}=(y_{\mathrm{k}},y_{\mathrm{k}},\ldots,y_{\mathrm{k}})_d^T$ $(k=1,2,\ldots,L)$ be the k-th label, where L is the total number of different labels, and d denotes the dimension of feature vectors. And then we use the new labels above to form a new training set, in which each training example contains the label information. The specific approach is that for each label $y_k(k=1,2,\ldots,L)$, add the corresponding label vector $y_k^{'}$ to each bag, then we get $X_i^{'}=\left[x_{i1},x_{i2},\ldots,x_{in},y_1^{'},y_2^{'},\ldots,y_{k-1}^{'}\right](i=1,2,\ldots,m)$, where x_{ij} is an instance of bag X_i, n is the number of instance in bag X_i, m is the number of examples. Therefore for each label y_k, we can get a set of training data $S_k=\left\{\left(X_{\mathrm{i}}^{'},\phi(X_{\mathrm{i}},y_k)\right)\right\}(k=1,2,\ldots,L)$, and based on that we can train the SVMs according to formulation (3). After the extension of the labels and the training bags, the ECC method can deal with MIML problems.

In classifier chain models, the order of the chain itself will normally have an effect on accuracy. Using an Ensemble of Classifier Chains (ECC), each with a random label order, greatly reduces the risk of these events having an overall negative effect on classification accuracy at only a linear time cost with respect to the number of iterations [19]. ECC trains N CC classifiers $h_1,\ldots,h_{\mathrm{N}}$, and each classifier is given a random chain order. Using the output $\hat{\mathrm{y}}_1,\ldots,\hat{\mathrm{y}}_N$ to calculate the confidence vector $\hat{\mathrm{w}}=\left[\hat{w}_1,\ldots,\hat{w}_L\right]\in R^L$, where L is the total number of different labels, $\hat{w}_j$ is the confidence of the j-th label, R^L is the output space,

$$\hat{w}_j=\frac{1}{N}\sum_{k=1}^{N}\hat{y}_{j,k} \tag{7}$$

A threshold function can be applied to $\hat{\mathrm{W}}$ to predict $\hat{\mathrm{y}}$:

$$\hat{y}_j=\begin{cases}1 & if\ \hat{w}_j \geq t\\ 0 & if\ \hat{w}_j \geq t\end{cases} \tag{8}$$

Where t is the threshold and is calibrated as follows:

$$\mathrm{t}=\arg\min_t\left\|LCARD(S)-\left(\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{\mathrm{L}}1_{\hat{w}_j\geq t}\right)\right\| \tag{9}$$

Where N is the number of test examples. $LCARD(S)=\frac{1}{N}\sum_{\mathrm{i}=1}^{N}\sum_{j=1}^{\mathrm{L}}y_j^i$ denotes the average number of labels associated with each example [20].

Table 1 Characteristics of the data set

Data set	No. of example	No. of class	Instances per bag		Labels per example(k)		Training set size	Test set size	
			Min	Max	k=1	k=2	k=3		
Scene	2000	5	9	9	1543	442	15	1000	1000
Reuters	2000	7	2	26	1701	290	9	1000	1000

Table 2 Experimental results on the data set

Metric	ECC-MIMLSVM$^+$	E-MIMLSVM$^+$	MIMLBOOST	MIMLSVM
hamming loss	$\mathbf{0.095 \pm 0.001}$	0.222 ± 0.065	0.229 ± 0.022	0.194 ± 0.005
one-error	$\mathbf{0.120 \pm 0.006}$	0.316 ± 0.005	0.417 ± 0.009	0.386 ± 0.016
Coverage	$\mathbf{0.502 \pm 0.017}$	0.926 ± 0.004	0.960 ± 0.014	1.034 ± 0.003
ranking loss	$\mathbf{0.056 \pm 0.002}$	0.173 ± 0.004	0.203 ± 0.005	0.217 ± 0.012
average precision	$\mathbf{0.931 \pm 0.013}$	0.792 ± 0.008	0.771 ± 0.012	0.750 ± 0.017

For an unseen bag X, and for each chain order, we make predictions from the first label of the chain. Before making the prediction of the j-th label, let $X' = \left[x_1, x_2, \ldots, x_n, \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{j-1}\right]$, where $\hat{y}_i$ is a vector extended from the prediction of bag X_i on the i-th classifier, and then we use X'_i to predict the j-th label $\hat{y}_j = f_j(X')$. Finally, we use the predictions to calculate the confidence vector and achieve the final prediction set Y through a threshold function.

Figure 1 illustrates the pseudo-code of our proposed algorithm.

The main idea of our proposed algorithm is that firstly we modify the MIML training examples to adapt to the ECC algorithm, secondly use the training process of ECC to train a classifier chains, and then for each classifier in the classifier chains, we exploit the training process of MIMLSVM$^+$ algorithm. In the process of testing, we should modify examples in the test set, and use the predicted values to calculate a confidence vector $\hat{w}$, each dimension of the vector denotes the confidence of a label. Last according to a threshold t and the confidence $\hat{w}_j (j = 1, 2, \ldots, L)$ we can get the final predicted values.

4 Experiment and argumentation

4.1 Experimental setup

In this section, performance of ECC-MIMLSVM$^+$ is compared with MIML-BOOST, MIMLSVM and E-MIMLSVM$^+$ on two real-world MIML learning tasks. The first data set is scene dataset that collected from the COREL image collection and the Internet. Scene classification data contains 2,000 natural scene images. There are 5 possible class labels such as deserted, mountains, sea, sunset and trees and a set of labels are manually assigned to each image. Images belonging to more than one class comprise over 22 % of the data set and the average number of labels per image is 1.24 ± 0.44. Each image is represented as a bag of nine 15-dimension instances.

The second data set is text data which is collected from the widely studied Reuters-21578 collection [21]. The seven most frequent categories are considered. After removing documents whose label sets or main texts are empty, 8,866 documents are retained where only 3.37 % of them are associated with more than one class label. After randomly removing documents with only one label, a text categorization data set containing 2,000 documents is obtained. Around 15 % documents with multiple labels comprise the resultant data set and the average number of labels per document is 1.15 ± 0.37. Each document is represented as a bag of instances using the sliding window techniques [22], where each instance corresponds to a text segment enclosed in one sliding window of size 50. "Function words" on the SMART stop-list [23] are removed from the vocabulary and the remaining words are stemmed. Instances in bags adopt the "Bag-of-Words" representation based on term frequency [21, 24]. Without loss of effectiveness, dimensionality reduction is performed by retaining the top 2 % words with highest document frequency [25]. Thereafter, each instance is represented as a 243-dimensional feature vector. Table 1 summarizes characteristics of both data sets.

ECC-MIMLSVM$^+$ is compared with E-MIMLSVM$^+$, MIMLBOOS and MIMLSVM. The parameters of EMIMLSVM$^+$MIMLBOOST and MIMLSVM are set

Table 3 Experimental results on the Reuters data set

Metric	ECC-MIMLSVM$^+$	E-MIMLSVM$^+$	MIMLBOOST	MIMLSVM
hamming loss	$\mathbf{0.016 \pm 0.001}$	0.033 ± 0.002	0.056 ± 0.001	0.043 ± 0.001
one-error	$\mathbf{0.033 \pm 0.002}$	0.060 ± 0.002	0.113 ± 0.003	0.105 ± 0.003
coverage	$\mathbf{0.217 \pm 0.005}$	0.278 ± 0.010	0.419 ± 0.005	0.389 ± 0.009
ranking loss	$\mathbf{0.009 \pm 0.001}$	0.020 ± 0.003	0.040 ± 0.001	0.033 ± 0.002
average precision	$\mathbf{0.979 \pm 0.004}$	0.964 ± 0.003	0.926 ± 0.007	0.934 ± 0.004

Table 4 Training time of each algorithm on both data sets

		ECC-MIMLSVM$^+$	E-MIMLSVM$^+$	MIMLBOOST	MIMLSVM
Training (minutes)	Scene	19.91 ± 0.28	845.86 ± 15.52	7134.54 ± 34.91	10.48 ± 0.43
	Reuters	10.12 ± 0.13	586.41 ± 6.77	4334.04 ± 11.53	5.18 ± 0.13

according to [1, 8] respectively. Particularly, the number of boosting rounds for MIMLBOOST is set to be 25 and Gaussian kernel with $\gamma = 0.2^2$ is used to implement MIMLSVM. The kernel parameters of E-MIMLSVM$^+$ are $\gamma_1 = 10^{-5}$ and $\gamma_2 = 10^{-2}$, and the cluster parameter q is set to be 0.5. The number of chain labels order is set to be 3 in the ECC-MIMLSVM$^+$. For fair comparison, we employ the same setting with the same partition of data sets and report the average performance.

The performance of the four MIML algorithms is evaluated according to five popular multi-label metrics: hamming loss, one-error, coverage, ranking loss and average precision. Briefly, for hamming loss, one-error, coverage and ranking loss, the smaller value the better performance; for average precision, the bigger value the better performance.

4.2 Experimental results

Tables 2 and 3 show the experimental results of each compared algorithm with the five metrics and running time on the scene data and Reuters data respectively. For hamming loss, one-error, coverage, ranking loss and running time the smaller value the better performance, and for average precision the bigger value the better performance. The best result on each evaluation criterion is highlighted in boldface. As can be seen from Tables 2 and 3, ECC-MIMLSVM$^+$ performs better than other three algorithms on both data sets.

Table 4 shows the time consumed in the four compared algorithm on both data sets. As can be seen from Table 4, the ECC-MIMLSVM$^+$ is slightly worse than MIMLSVM, while far superior than MIMLBOOST and E-MIMLSVM$^+$.

5 Conclusions

In this paper, a novel SVM method for MIML problem named ECC-MIMLSVM$^+$ is proposed. This method considers the connections between labels through classifier chains method in an ensemble framework (ECC). Experiments on both scene classification and text categorization show that our method is more efficient and can produce better performance than other MIML methods.

Acknowledgments This paper is supported by the Fundamental Research Funds for the Central Universities (No. R1407008A, 09CX04031A). The authors are grateful for the anonymous reviewers who made constructive comments.

References

1. Zhou ZH, Zhang ML (2006) Multi-instance multi-label learning with application to scene classification. Neural Inf Process Sys (NIPS):1609–1616
2. Zhou ZH, Zhang ML et al (2012) Multi-instance multi-label learning. Artif Intell 176(1):2291–2320
3. Zhang ML, Wang ZJ (2009) MIMLRBF: RBF neural networks for multi-instance multi-label learning. Neurocomputing 72(16–18):3951–3956
4. Vapnik V (1995) The nature of statistical learning theory. Springer-Verlag, New York
5. Li CH, Liu KW, W HX (2011) The incremental learning algorithm with support vector machine based on hyperplane-distance. Appl Intell 34(1):19–27
6. Qian HM, Mao YB, Xiang WB, Wang ZQ (2010) Recognition of human activities using SVM multi-class classifier. Pattern Recogn Lett 31:100–111
7. Chen WJ, Shao YH, Xu DK (2014) Manifold proximal support vector machine for semi-supervised classification. Appl Intell 40(4):623-638
8. Li YX, Ji SW, Kumar S, Ye JP, Zhou ZH (2012) Drosophila gene expression pattern annotation through multi-instance multi-label learning. Trans Comput Biol Bioinformatics 9(1):98–112
9. Gartner T, Flach PA, Smola AJ (2002) Multi-instance kernels. In: Proceedings of the 19th intenational conference on machine learning, Sydney, Australia, pp 179–186
10. Evgeniou T, Pontil M (2004) Regularized multi-task learning. In: Proceedings of the 10th ACM SIGKDD international conference knowledge discovery data mining, pp 109–117
11. Zhang J, Ghahramani Z, Yang Y (2008) Flexible latent variable models for multi-task learning. Mach Learn 73(3):221–242
12. Evgeniou T, Micchelli CA, Pontil M (2005) Learning multiple tasks with kernel methods. Mach Learn Res 6:615–637
13. Dietterich TG, Lathrop RH, Lozano-Pérez T (1997) Solving the multiple-instance problem with axis-parallel rectangles. Artif Intell 89(1-2):31–71
14. Boutell MR, Luo J, Shen X, Brown CM (2004) Learning multi-label scene classification. Pattern Recogn 37(9):1757–1771
15. Schapire RE, Singer Y (2000) BoosTexter: a boosting-based system for text categorization. Mach Learn 39(2-3):135–168
16. Zhou ZH, Zhang ML, Huang SJ, Li YF (2008) MIML: a framework for learning with ambiguous objects. CORR abs/0808. 3231

17. Zhang ML, Zhou ZH (2008) M^3MIML: A maximum margin method for multi-instance multi-label learning. In: Proceedings of the 8th IEEE international conference on data mining (ICDM'08). Pisa, Italy, pp 688–697
18. Haussler D (1999) Convolution kernels on discrete structures. Technical report UCSC-CRL-99-10, Department of Computer Science, University of California at Santa Cruz, CA, Santa Cruz
19. Jesse R, Bernhard P, Geoff H, Eibe F (2011) Classifier chain for multi-label classification. Mach Learn 85:333–359
20. Tsoumakas G, Katakis (2007) Multi label classification: an overview. Int J Data Warehous Min 3(3):1–13
21. Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47
22. Andrews S, Tsochantaridis I, Hofmann T (2003) Support vector machines for multiple-instance learning. Adv 696 Neural Inf Process Syst 15(15):561–568
23. Salton G (1989) Automatic text processing: The transformation, analysis, and retrieval of information by computer. Addison-Wesley Reading, Pennsylvania
24. Dumais ST, Platt J, Heckerman D, Sahami M (1998) Inductive-learning algorithms and representation for text categorization. In: Proceedings of the 7th ACM international conference on information and knowledge management, Bethesda, MD, pp 148–155
25. Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. In: Proceedings of the 14th international conference on machine learning, pp 412–420