A k-Nearest Neighbor Based Multi-Instance Multi-Label Learning Algorithm

Min-Ling Zhang [1,2]
[1] School of Computer Science and Engineering,
Southeast University, Nanjing 210096, China
[2] National Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210093, China
Email: zhangml@seu.edu.cn

***Abstract*—In multi-instance multi-label learning (i.e. MIML), each example is not only represented by *multiple instances* but also associated with *multiple labels*. Most existing algorithms solve MIML problem via the intuitive way of identifying its equivalence in degenerated version of MIML. However, this identification process may lose useful information encoded in training examples and therefore be harmful to the learning algorithm's performance. In this paper, a novel algorithm named MIML-kNN is proposed for MIML by utilizing the popular k-nearest neighbor techniques. Given a test example, MIML-kNN not only considers its neighbors, but also considers its citers which regard it as their own neighbors. The label set of the test example is determined by exploiting the labeling information conveyed by its neighbors and citers. Experiments on two real-world MIML tasks, i.e. scene classification and text categorization, show that MIML-kNN achieves superior performance than some existing MIML algorithms.**

I. Introduction

In traditional supervised learning, each object (or *example*) is represented by a single instance as well as associated with a single label. In other words, let $\mathcal{X} = \mathcal{R}^d$ be the input space and $\mathcal{Y} = \{1, 2, \ldots, Q\}$ be the label space, the task of traditional supervised learning is to learn a function $h : \mathcal{X} \to \mathcal{Y}$ from a number of training examples $\{(\mathbf{x}_i, y_i) | 1 \leq i \leq N\}$, where $\mathbf{x}_i \in \mathcal{X}$ is a single instance and $y_i \in \mathcal{Y}$ is the single label associated with $\mathbf{x}_i$. Although traditional supervised learning has been proven to be very successful in handling diverse real-world problems, it may not work well when the concerned object could not fit the *single-instance single-label* (SISL) formalization well.

For example, in image classification, an image generally contains several naturally partitioned patches each can be represented as an instance, while such an image can be related to multiple semantic classes, such as *clouds*, *grassland* and *lions*; In text categorization, each document usually consists of several sections or paragraphs each can be regarded as an instance, while the document may be assigned to a set of predefined topics, such as *Shanghai World Expo*, *economics* and even *volunteers*; In bioinformatics, an gene sequence generally encodes a number of segments each can be expressed as an instance, while this sequence may be associated with several functional classes, such as *metabolism*, *transcription* and *protein synthesis*.

In view of the above problem, a new learning framework named *multi-instance multi-label learning* (MIML) [17], [18] is recently proposed. In MIML, each example is represented by *multiple instances* and at the same time associated with *multiple labels*. In other words, the task of MIML is to learn a function $h_{\text{MIML}} : 2^{\mathcal{X}} \to 2^{\mathcal{Y}}$ from a number of training examples $\{(X_i, Y_i) | 1 \leq i \leq N\}$, where $X_i \subseteq \mathcal{X}$ is a bag of instances $\{\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_{n_i}^i\}$ and $Y_i \subseteq \mathcal{Y}$ is a set of labels $\{y_1^i, y_2^i, \ldots, y_{l_i}^i\}$ associated with X_i. Here, n_i is the number of instances in X_i and l_i is the number of labels in Y_i.

It is obvious that traditional supervised learning can be regarded as a *degenerated* version of MIML, where each example is restricted to have only single instance and single label. On the other hand, the generality of MIML inevitably makes this learning problem more difficult to be dealt with. Currently, most existing algorithms deal with MIML problem by identifying its equivalence in traditional supervised learning via problem reduction [17], [18]. Although this kind of identification strategy is feasible, the performance of the resultant algorithms may suffer from the information loss incurred during the reduction process.

In this paper, a novel algorithm named MIML-kNN (Multi-Instance Multi-Label k-Nearest Neighbor) is proposed to learn from MIML examples. Given a test example, MIML-kNN not only considers its neighbors, but also considers its *citers* which regard it as their own neighbors. After that, a *label counting vector* is constructed from its neighbors and citers, and then fed to Q trained linear classifiers for prediction. As the correlations between instances and labels are fully exploited in the proposed method, experimental results show that the performance of MIML-kNN is highly competitive to other existing MIML algorithms.

The rest of this paper is organized as follows. Section II reviews related works. Section III presents MIML-kNN. Section 4 reports the experimental results. Finally, Section V concludes and indicates several issues for future work.

II. Related Works

The MIML framework is closely related to the learning frameworks of *multi-instance learning* [3] and *multi-label learning* [7], [10].

Multi-instance learning [3] was coined by Dietterich et al. in their investigation of drug activity prediction problem. The task of multi-instance learning is to learn a function $h_{\text{MIL}} : 2^{\mathcal{X}} \rightarrow \{+1, -1\}$ from a number of training examples $\{(X_i, y_i)|1 \leq i \leq N\}$, where $X_i \subseteq \mathcal{X}$ is a bag of instances $\{\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_{n_i}^i\}$ and $y_i \in \{+1, -1\}$ is the label associated with X_i. Multi-label learning [7], [10] originated from the investigation of text categorization problems. The task of multi-label learning is to learn a function $h_{\text{MLL}} : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$ from a number of training examples $\{(\mathbf{x}_i, Y_i)|1 \leq i \leq N\}$, where $\mathbf{x}_i \in \mathcal{X}$ is an instance and $Y_i \subseteq \mathcal{Y}$ is a set of labels $\{y_1^i, y_2^i, \ldots, y_{l_i}^i\}$ associated with $\mathbf{x}_i$.

It is obvious that traditional supervised learning can be regarded as a degenerated version of either multi-instance learning or multi-label learning, while those three learning frameworks are all degenerated versions of MIML. Therefore, it is natural to solve the MIML problem by identifying its equivalence in traditional supervised learning, using either multi-instance learning or multi-label learning as the bridge. Actually, two MIML algorithms named MIMLBOOST and MIMLSVM have been designed by adopting this strategy.

For the MIMLBOOST algorithm, the original MIML task is firstly transformed into a multi-instance learning task by converting each MIML example (X_i, Y_i) into $|\mathcal{Y}|$ number of multi-instance examples $\{([X_i, y], \Phi[X_i, y]) \,|y \in \mathcal{Y}\}$. Here, $[X_i, y]$ contains n_i instances $\{(\mathbf{x}_1^i, y), \ldots, (\mathbf{x}_{n_i}^i, y)\}$ formed by concatenating each of X_i's instance with label y, while $\Phi[X_i, y] = +1$ if $y \in Y_i$ and -1 otherwise. After that, MIMLBOOST solves the derived multi-instance learning problem by employing the MIBOOSTING algorithm [13], which reduces a multi-instance learning task into a traditional supervised learning task under the assumption that each instance in the bag contributes equally and independently to a bag's label.

For the MIMLSVM algorithm, the original MIML task is firstly transformed into a multi-label learning task by converting each MIML example (X_i, Y_i) into a multi-label example $(\tau(X_i), Y_i)$. Here, the function $\tau(\cdot)$ maps a bag of instances X_i into a single instance $\mathbf{z}_i$ with the help of *constructive clustering* [17]. Specifically, k-medoids clustering is performed on $\Lambda = \{X_1, X_2, \ldots, X_N\}$ at the level of bags and components of $\mathbf{z}_i$ correspond to the distances between X_i and the medoids of the clustered groups. After that, MIMLSVM solves the derived multi-label learning problem by employing the MLSVM algorithm [2], which decomposes the multi-label learning task into a number of independent traditional (binary) supervised learning problems (one per class).

Following the work of MIMLBOOST and MIMLSVM [17], there have been some recent progresses on MIML framework, such as designing MIML algorithms based on regularization [18], [15], metric learning from MIML data [5] and applications of MIML techniques in image analysis [14] and bioinformatics [6].

III. THE PROPOSED METHOD

MIML-kNN solves MIML problem by using the popular k-nearest neighbor techniques. Given an example $X \subseteq \mathcal{X}$ and its associated label set $Y \subseteq \mathcal{Y}$, let $\vec{L}_X$ denote the Q-dimensional category vector of X. Here, its l-th component $\vec{L}_X(l)$ takes the value of $+1$ if $l \in Y$, otherwise $\vec{L}_X(l) = -1$. Furthermore, let $\mathcal{R}_X^r$ denote the set of its r *nearest neighbors* identified in the MIML training set $S = \{(X_i, Y_i)|1 \leq i \leq N\}$, i.e.:

$$\mathcal{R}_X^r = \{A|A \text{ is one of } X'\text{s } r \text{ nearest neighbors in } \Lambda\} \quad (1)$$

Here, $\Lambda = \{X_i|(X_i, Y_i) \in S\}$.

Furthermore, let $\mathcal{C}_X^c$ denote the set of its c *citers* identified in S, i.e.:

$$\mathcal{C}_X^c = \{B|X \text{ is among } B'\text{s } c \text{ nearest neighbors in } \Lambda'\} \quad (2)$$

Here, $\Lambda' = \{X\} \cup \Lambda$.

Intuitively speaking, MIML-kNN not only considers X's neighboring examples in the training set, but also considers those training examples which regard X as their own neighbors (i.e. the citers). Specifically, the idea of utilizing citers to help learn from MIML examples is motivated from the Citation-kNN approach [12], where citers are found to be

TABLE I
PSEUDO-CODE OF MIML-kNN.

```
[$\vec{L}_P, \vec{V}_P$] = MIML-$k$NN($S, r, c, P$)

Inputs:
$S$:    the set of MIML training examples $\{(X_i, Y_i)|1 \leq i \leq N\}$
        ($X_i \subseteq \mathcal{X}, Y_i \subseteq \mathcal{Y}$)
$r$:    the reference parameter considered in Eq.(1)
$c$:    the citation parameter considered in Eq.(2)
$P$:    the test MIML example ($P \subseteq \mathcal{X}$)

Outputs:
$\vec{L}_P$:   the category vector for $P$
$\vec{V}_P$:   the $Q$-dimensional output vector for $P$, where $\vec{V}_P(l) = f(P, l)$
        as shown in Eq.(5)

Process:
1.   for $i$=1 to $N$ do
2.     Identify $\mathcal{R}_{X_i}^r$ for $X_i$ according to Eq.(1);
3.     Identify $\mathcal{C}_{X_i}^c$ for $X_i$ according to Eq.(2);
4.     Calculate label counting vector $\vec{\delta}_{X_i}$ for $X_i$ using Eq.(4);
5.   endfor
6.   Form matrices $\mathbf{\Phi}$ and $\mathbf{T}$ as shown in Eq.(7);
7.   Compute weight matrix $\mathbf{W}$ by solving Eq.(7) with SVD [9];
8.   for $l \in \mathcal{Y}$ do
9.     Identify $\mathcal{R}_P^r$ and $\mathcal{C}_P^c$ for $P$ with Eq.(1) and Eq.(2);
10.  endfor
11.  Calculate label counting vector $\vec{\delta}_P$ for $P$ using Eq.(4);
12.  for $l \in \mathcal{Y}$ do
13.    Set $\vec{V}_P(l)$ to $f(P, l)$ according to Eq.(5)
14.    Set $\vec{L}_P(l)$ to +1 if $\vec{V}_P(l) > 0$;
       Otherwise, set $\vec{L}_P(l) = -1$
15.  endfor
```

TABLE II
CHARACTERISTICS OF THE DATA SETS.

				Instances per bag			Labels per example (k)		
Data set	#examples	#classes	#features	min	max	mean±std.	k=1	k=2	$k\geq$3
Scene	2,000	5	15	9	9	9.00±0.00	1,543	442	15
Reuters	2,000	7	243	2	26	3.56±2.71	1,701	290	9

beneficial to learn from examples with multi-instance representation. Experimental results (Figure 1, Subsection IV-C) show that the learning performance of MIML-kNN could be significantly improved by exploiting citers in addition to neighboring examples.

Note that in MIML, each example is represented by a bag of instances instead of a single instance. Previous researches [12], [16] show that distance between bags could be effectively measured by the Hausdorff metric [4]. Specifically, a variant of Hausdorff metric, i.e. the *average Hausdorff metric* [16] is employed to measure the distance between bags. Given two different bags $U = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{n_u}\}$ and $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n_v}\}$, their average Hausdorff distance is calculated as:

$$\mathrm{aveH}(U,V) = \frac{\sum\limits_{\mathbf{u}\in U}\min\limits_{\mathbf{v}\in V}||\mathbf{u},\mathbf{v}|| + \sum\limits_{\mathbf{v}\in V}\min\limits_{\mathbf{u}\in U}||\mathbf{v},\mathbf{u}||}{|U|+|V|} \quad (3)$$

Here, $|\cdot|$ gives the set cardinality and $||\cdot||$ returns the distance between two instances (Euclidean distance in this paper). Conceptually, for each instance in one bag, $\mathrm{aveH}(\cdot,\cdot)$ calculates the distance to its nearest neighbor in the other bag. After traversing all the instances in both bags, the average value out of the above distances is reported as $\mathrm{aveH}(\cdot,\cdot)$'s output.

Given the example $X \subseteq \mathcal{X}$, we can identify its neighboring examples $\mathcal{R}_X^r$ (Eq.(1)) and citers $\mathcal{C}_X^c$ (Eq.(2)) in S by employing the average Hausdorff metric, and then define the following Q-dimensional *label counting vector* $\vec{\delta}_X$:

$$\vec{\delta}_X(l) = \sum\nolimits_{Z\in\left(\mathcal{R}_X^r\cup\mathcal{C}_X^c\right)} [\![\vec{L}_Z(l) == 1]\!], \quad l \in \mathcal{Y} \quad (4)$$

For any predictor π, $[\![\pi]\!]$ takes the value of 1 if π holds and 0 otherwise. Actually, $\vec{\delta}_X(l)$ records the number of examples in $\mathcal{R}_X^r \cup \mathcal{C}_X^c$ which have label l.

Given a test example $P \subseteq \mathcal{X}$, MIML-kNN firstly determines its label counting vector $\vec{\delta}_P$. Based on the information conveyed by $\vec{\delta}_P$, the following linear classifier is utilized to yield the output of MIML-kNN on the l-th class:

$$f(P,l) = \mathbf{w}_l^{\mathrm{T}} \cdot \vec{\delta}_P, \quad l \in \mathcal{Y} \quad (5)$$

Here, $\mathbf{w}_l$ is a Q-dimensional column vector corresponding to the l-th class and T represents the transpose operation. As shown in the above equation, the prediction of P on the l-th class (i.e. $f(P,l)$) is set to be the *dot product* between $\mathbf{w}_l$ and the label counting vector $\vec{\delta}_P$. The l-th label is deemed to belong to P only if $f(P,l) > 0$. Therefore, MIML-kNN predicts the set of labels associated with P as $h_{\mathrm{MIML}}(P) = \{l | f(P,l) > 0, l \in \mathcal{Y}\}$.

To obtain the column vectors $\mathbf{w}_l$ ($l \in \mathcal{Y}$), MIML-kNN works by minimizing the following sum-of-squares error function:

$$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{l\in\mathcal{Y}}\left(\hat{L}_{X_i}(l) - \vec{L}_{X_i}(l)\right)^2 \quad (6)$$

Here, $\hat{L}_{X_i}(l) = \mathbf{w}_l^{\mathrm{T}} \cdot \vec{\delta}_{X_i}$ represents the actual output of the i-th training example X_i on the l-th class, i.e. the dot product between the column vector $\mathbf{w}_l$ and the label counting vector $\vec{\delta}_{X_i}$. $\vec{L}_{X_i}$ is the label vector of X_i and $\vec{L}_{X_i}(l)$ represents the desired output of X_i on the l-th class. Differentiating the error function of Eq.(6) with respect to each component of $\mathbf{w}_l$, and setting the derivatives to zero gives the normal equations for the least sum-of-squares problem as follows:

$$\left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right) \cdot \mathbf{W} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{T} \quad (7)$$

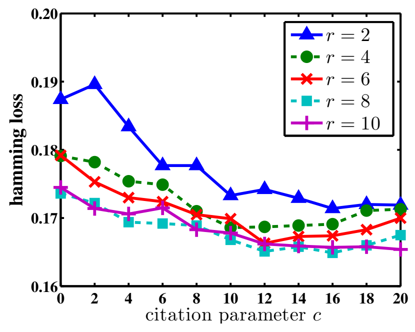
Here, $\mathbf{\Phi} = [\phi_{il}]_{N\times Q}$ with elements $\phi_{il} = \vec{\delta}_{X_i}(l)$, $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_Q]$, and $\mathbf{T} = [t_{il}]_{N\times Q}$ with elements $t_{il} = \vec{L}_{X_i}(l)$. In this paper, the weight matrix $\mathbf{W}$ is computed by solving Eq.(7) using linear matrix inversion techniques of SVD [9].

Table I gives the detailed description of MIML-kNN. Based on the MIML training set, MIML-kNN firstly learns the weight matrix $\mathbf{W}$ used by the linear classifiers (Steps 1 to 7); After that, the label counting vector for the test example P is calculated (Steps 8 to 11); Finally, the derived vector is fed to the trained linear classifiers to yield final predictions (Steps 12 to 15).

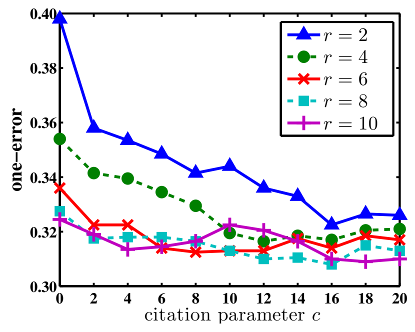
IV. EXPERIMENTS

A. Experimental Setup

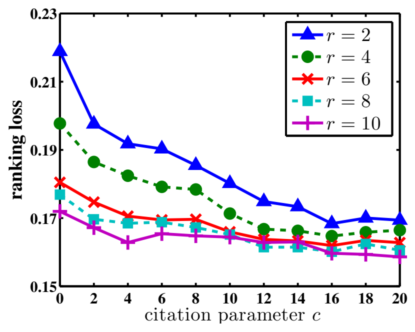
The performance of MIML-kNN is compared with MIMLBOOST and MIMLSVM on two real-world MIML tasks. The first is the benchmark scene classification task studied in seminal works on MIML framework [17]. The scene classification data contains 2,000 natural scene images. All the possible class labels are *desert*, *mountains*, *sea*, *sunset* and *trees* and a set of labels is manually assigned to each image. Images with multiples labels 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-dimensional instances using the SBN image bag generator [8], where each instance corresponds to an image patch. The data set with detailed description can be found at http://lamda.nju.edu.cn/datacode/miml-image-data.htm.



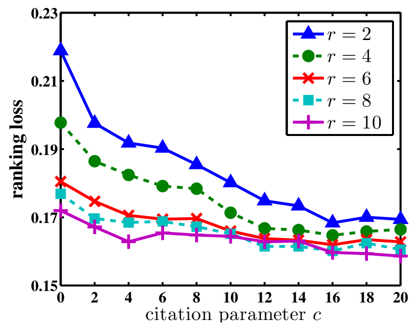
(a) *hamming loss*



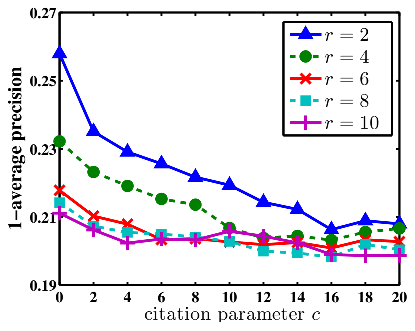
(b) *one-error*



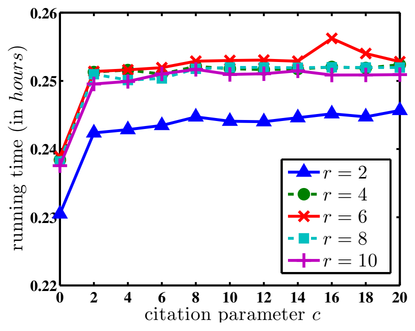
(c) *ranking loss*



(d) *ranking loss*



(e) 1-*average precision*



(f) *running time*

Fig. 1. The performance as well as running time of MIML-kNN changes as the value of c (number of citers) increases, under different values of r (number of nearest neighbors) on the scene data.

In addition to scene classification, we have also studied another MIML task of text categorization. Specifically, the widely used Reuters-21578 collection [11] is employed in experiment. The seven most frequent categories are considered. After removing documents without labels or main texts and randomly removing some documents with only one label, a text categorization data set containing 2,000 documents is obtained. Documents with multiple labels comprise around 15% of the data set and the average number of labels per document is 1.15±0.37. Each document is represented as a bag of instances based on the techniques of sliding windows [1], where each instance corresponds to a text segment enclosed in a sliding window of size 50 (overlapped with 25 words). Function words are excluded from the vocabulary and stemming is performed on the remaining words. Instances in the bags adopt the *"Bag-of-Words"* representation based on term frequency [11]. Without loss of effectiveness, dimensionality reduction is conducted and the top 2% words with highest document frequency are retained. Finally, each instance is represented as a 243-dimensional feature vector. The data set with detailed description can be found at http://lamda.nju.edu.cn/datacode/miml-text-data.htm. Table II summarizes characteristics of both data sets.

B. Evaluation Metrics

Since MIML algorithms make multi-label predictions, their performance are evaluated with five popular multi-label metrics [10], i.e. *hamming loss*, *one-error*, *coverage*, *ranking loss* and *average precision*. Given an MIML data set $\mathcal{D}=\{(X_i,Y_i)|1\leq i\leq p\}$, the five metrics are defined as below. Here, $h(X_i)$ returns a set of proper labels of X_i; $h(X_i,y)$ returns a real-value indicating the confidence for y to be a proper label of X_i; $rank^h(X_i,y)$ returns the rank of y derived from $h(X_i,y)$.

- *Hamming loss*:

$$\text{hloss}_{\mathcal{D}}(h)=\frac{1}{p}\sum_{i=1}^{p}\frac{1}{|\mathcal{Y}|}|h(X_i)\Delta Y_i| \tag{8}$$

Here Δ denotes the symmetric difference between two sets. The hamming loss evaluates how many times an example-label pair is misclassified.

- *One-error*:

$$\text{one−error}_{\mathcal{D}}(h)=\frac{1}{p}\sum_{i=1}^{p}[\![\,[\arg\max_{y\in\mathcal{Y}}h(X_i,y)]\notin Y_i]\!] \tag{9}$$

Here for predicate π, $[\![\pi]\!]$ equals 1 if π holds and 0 otherwise. The one-error evaluates how many times the top-ranked label is not in the set of proper labels of the example.

- *Coverage*:

$$\text{coverage}_{\mathcal{D}}(h)=\frac{1}{p}\sum_{i=1}^{p}\max_{y\in Y_i}rank^h(X_i,y)-1 \tag{10}$$

TABLE III
EXPERIMENTAL RESULTS ON THE SCENE DATA SET.

Evaluation	Compared Algorithm		
Criterion	MIML-kNN	MIMLBOOST	MIMLSVM
hamming loss$^{\downarrow}$	**0.165± 0.007**	0.189±0.007	0.185±0.011
one-error$^{\downarrow}$	**0.310± 0.024**	0.335±0.021	0.347±0.026
coverage$^{\downarrow}$	**0.896± 0.058**	0.947±0.056	1.031±0.068
ranking loss$^{\downarrow}$	**0.159± 0.015**	0.172±0.011	0.191±0.015
average precision$^{\uparrow}$	**0.801± 0.014**	0.785±0.012	0.774±0.015

TABLE IV
EXPERIMENTAL RESULTS ON THE REUTERS DATA SET.

Evaluation	Compared Algorithm		
Criterion	MIML-kNN	MIMLBOOST	MIMLSVM
hamming loss$^{\downarrow}$	**0.039±0.006**	0.053±0.009	0.043±0.007
one-error$^{\downarrow}$	**0.069±0.022**	0.017±0.022	0.101±0.024
coverage$^{\downarrow}$	**0.341±0.067**	0.417±0.047	0.379±0.062
ranking loss$^{\downarrow}$	**0.027±0.008**	0.039±0.007	0.033±0.008
average precision$^{\uparrow}$	**0.954±0.013**	0.930±0.012	0.937±0.015

The coverage evaluates how many steps are need, on average, to move down the label list in order to cover all the proper labels of the example.

- *Ranking loss*:

$$\text{rloss}_{\mathcal{D}}(h) = \frac{1}{p}\sum_{i=1}^{p}\frac{1}{|Y_i||\bar{Y}_i|}\cdot|\mathcal{R}_i|, \quad \text{where}$$
$$\mathcal{R}_i = \{(y_1,y_2)|h(X_i,y_1) \le h(X,y_2),\ (y_1,y_2)\in Y_i\times\bar{Y}_i\} \quad (11)$$

Here $\bar{Y}_i$ denotes the complementary set of Y_i in $\mathcal{Y}$. The ranking loss evaluates the average fraction of label pairs that are misordered for the example.

- *Average precision*:

$$\text{avgprec}_{\mathcal{D}}(h) = \frac{1}{p}\sum_{i=1}^{p}\frac{1}{|Y_i|}\cdot\frac{|\mathcal{P}_i|}{rank^h(X_i,y))}, \quad \text{where}$$
$$\mathcal{P}_i = \{y'|rank^h(X_i,y') \le rank^h(X_i,y),\ y'\in Y_i\} \quad (12)$$

The average precision evaluates the average fraction of proper labels ranked above a particular label $y \in Y_i$.

For the first four metrics, the *smaller* the value the better the performance. For *average precision*, the *bigger* the value the better the performance.

C. Experimental Results

As shown in Table I, the MIML-kNN algorithm involves two different parameters, i.e. r (the number of nearest neighbors considered) and c (the number of citers considered). Figure 1 illustrates how MIML-kNN performs on the scene data under different parameter configuration. Here, r varies from 2 to 10 with an interval of 2 and c increases from 0 to 20 with an interval of 2. In addition, 1-*average precision* is plotted instead of *average precision* such that in Figure 1(a) to Figure 1(e), the lower the curve the better the performance.

As shown in Figure 1(a) to Figure 1(e), when r is fixed, the performance of MIML-kNN improves quickly in the initial increasing phase of c and then gradually levels up; On the other hand, when c is fixed, the performance of MIML-kNN is worse when r is relatively small ($r = 2, 4$), while becomes better and indistinguishable to each other when r is relatively large ($r = 6, 8, 10$). As shown in Figure 1(f), as long as $r \ge 4$ and $c \ge 2$, the running time of MIML-kNN doesn't change significantly. Based on these observations, all the experimental results of MIML-kNN reported in the following are obtained by setting $r = 10$ and $c = 20$.

For MIMLBOOST and MIMLSVM, both of which adopt the best parameters as reported in [17]. Concretely, the number of boosting rounds for MIMLBOOST is set to 25 while Gaussian kernel with width parameter of 0.2 is used to implement MIMLSVM. The performance of each compared algorithm is evaluated by conducting *tenfold cross-validation* on the given data set.

Table III summarizes the experimental results of each compared algorithm on the scene data set. For each evaluation criterion, "↓" indicates "the smaller the better" while "↑" indicates "the bigger the better". Furthermore, the best result on each evaluation criterion is highlighted in boldface. It is obvious from Table III that MIML-kNN performs quite well on this data set. Specifically, pairwise t-tests at 0.05 significance level are further conducted to test whether the performance

TABLE V
TIME COST OF EACH COMPARED ALGORITHM ON BOTH DATA SETS (*measured in minutes*).

		Compared Algorithm		
		MIML-kNN	MIMLBOOST	MIMLSVM
Training Phase	*Scene*	7.72±0.01	6462.40±86.40	6.12±0.09
	Reuters	4.74±0.03	4354.98±64.89	3.01±0.12
Testing Phase	*Scene*	7.34±0.15	467.73±4.42	0.52±0.02
	Reuters	3.38±0.02	369.24±12.18	0.22±0.04

difference between two algorithms is statistically significant. It is impressive that as indicated by the significant tests, MIML-kNN is superior to MIMLBOOST and MIMLSVM in terms of all the evaluation metrics. These results show that MIML-kNN achieves highly competitive generalization performance on the scene data set.

Table IV summarizes the experimental results of each compared algorithm on the Reuters data set. The meanings for the "$\downarrow$" and "$\uparrow$" marks are the same as in Table III, and the best result on each evaluation criterion is also highlighted in boldface. It is obvious from Table IV that MIML-kNN performs quite well on this data set. Pairwise t-tests at 0.05 significance level reveal that: a) MIML-kNN is significantly superior to MIMLBOOST in terms of all the evaluation metrics; b) MIML-kNN is comparable to MIMLSVM in terms of *coverage*, while is significantly superior in terms of the other four evaluation metrics. These results show that MIML-kNN achieves highly competitive generalization performance on the Reuters data set.

In addition, Table V reports the time consumed in the training phase (Steps 1 to 7 in Table I) and testing phase (Steps 8 to 15) of MIML-kNN on both data sets, together with those of the compared algorithms. As shown by Table V, the training and testing efficiency of MIML-kNN is slightly worse than MIMLSVM while far superior than MIMLBOOST.

V. CONCLUSION

In this paper, a novel algorithm MIML-kNN is derived from the popular k-nearest neighbors techniques to learn from MIML examples. MIML-kNN exploits information conveyed by its neighboring examples as well as citing examples to make multi-label predictions for the unseen MIML example. In this way, the correlations between instances and labels of an MIML example are utilized to achieve strong generalization ability. Applications to two real-world MIML tasks show that MIML-kNN is superior to other MIML algorithms based on simple degeneration strategy.

Investigating whether better performance can be achieved by using metrics other than the average Hausdorff distance, and how to choose appropriate parameters for different data sets in a systematic way, are both interesting future works. Furthermore, it is very appealing to design other kinds of MIML algorithms and conduct comparative studies over more real-world MIML tasks.

ACKNOWLEDGEMENTS

The author wishes to thank the anonymous reviewers for their insightful comments which help much in improving the quality of this paper. This work is supported by the National Science Foundation of China (60805022), Ph.D. Programs Foundation of Ministry of Education of China (200802941009), and Open Foundation of National Key Laboratory for Novel Software Technology of China (KFKT2008B12).

REFERENCES

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

[2] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[3] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple-instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 39, no. 1-2, pp. 31–71, 1997.

[4] G. A. Edgar, *Measure, Topology, and Fractal Geometry, 3rd print*. Berlin: Springer-Verlag, 1995.

[5] R. Jin, S. Wang, and Z.-H. Zhou, "Learning a distance metric from multi-instance multi-label data," in *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami, FL, 2009, pp. 896–902.

[6] Y.-X. Li, S. Ji, J. Ye, S. Kumar, and Z.-H. Zhou, "Drosophila gene expression pattern annotation through multi-instance multi-label learning," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, Pasadena, CA, 2009, pp. 1445–1450.

[7] A. McCallum, "Multi-label text classification with a mixture model trained by EM," in *Working Notes of the AAAI'99 Workshop on Text Learning*, Orlando, FL, 1999.

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

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: the Art of Scientific Computing*. New York: Cambridge University Press, 1992.

[10] R. E. Schapire and Y. Singer, "Boostexter: a boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.

[11] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[12] 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*, Stanford, CA, 2000, pp. 1119–1125.

[13] X. Xu and E. Frank, "Logistic regression and boosting for labeled bags of instances," in *Lecture Notes in Computer Science 3056*, H. Dai, R. Srikant, and C.-Q. Zhang, Eds. Berlin: Springer, 2004, pp. 272–281.

[14] Z.-J. Zha, X.-S. Hua, T. Mei, J. Wang, G.-J. Qi, and Z. Wang, "Joint multi-label multi-instance learning for image classification," in *Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, 2008, pp. 1–8.

[15] M.-L. Zhang and Z.-H. Zhou, "M^{3}MIML: A maximum margin method for multi-instance multi-label learning," in *Proceedings of the 8th International Conference on Data Mining*, Pisa, Italy, 2008, pp. 688–697.

[16] ——, "Multi-instance clustering with applications to multi-instance prediction," *Applied Intelligence*, vol. 31, no. 1, pp. 47–68, 2009.

[17] Z.-H. Zhou and M.-L. Zhang, "Multi-instance multi-label learning with application to scene classification," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. Cambridge, MA: MIT Press, 2007, pp. 1609–1616.

[18] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li, "MIML: A framework for learning with ambiguous objects," *CORR abs/0808.3231*, 2008.