Multimed Tools Appl
DOI 10.1007/s11042-016-3494-z

A multi-instance multi-label learning algorithm based on instance correlations

Chanjuan Liu[1,2] **· Tongtong Chen**[1] **· Xinmiao Ding**[3] **·**
Hailin Zou[1] **· Yan Tong**[2]

Received: 7 October 2015 / Revised: 19 March 2016 / Accepted: 23 March 2016

Abstract Existing multi-instance multi-label learning algorithms generally assume that instances in a bag are independent of each other, which is difficult to be guaranteed in practical applications. A novel multi-instance multi-label learning algorithm is proposed by modeling instance correlations in each bag. First, instance correlations are introduced in multi-instance multi-label learning by constructing graphs. Then, different kernel matrices are derived from kernel functions based on graphs at different scales, which are employed to train Multiple Kernel Support Vector Machine (MKSVM) classifiers. Experimental results on different datasets show that the proposed method significantly improves the accuracy of the multi-label classification compared with the state-of-the-art methods.

Keywords Multi-instance multi-label learning · Multi-label classification · Instance correlations · Multiple-kernel fusion

1 Introduction

Artificial intelligence is a branch of computer science. It attempts to understand the essence of intelligence and enables a new kind of “intelligent” machines, which can

✉ Chanjuan Liu
luckycj80@sina.com

Hailin Zou
zhl_8655@sina.com

[1] School of Information and Electrical Engineering, Ludong University, Yantai 264025, China

[2] Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208, USA

[3] School of Information and Electronic Engineering, Shandong Institute of Business and Technology, Yantai 264005, China

Published online: 06 April 2016

sense and response in the way to a human being. Researches in this field include robotics, object recognition and detection, scene understanding, image and video retrieval, natural language processing, expert system, etc.[3, 7, 10, 19, 20, 40, 54]. Machine learning is the essential part of artificial intelligence. In traditional supervised machine learning, an object is represented by an instance and associated with a label. Although this formalization is prevailing and successful, there are many complex real-world problems, which do not fit in this framework well. As a variant of supervised learning, an object can be represented by a bag formed by multiple instances and has a single label in multi-instance learning. Because of its unique properties, multi-instance learning is also regarded as the fourth learning framework in parallel with supervised learning, unsupervised learning, and reinforcement learning.

More recently, multi-instance multi-label learning (MIML) emerged by associating an object with multiple labels. Different from multi-instance learning, a bag is labeled with several kinds of labels in multi-instance multi-label learning. Compared to traditional learning frameworks, the multi-instance multi-label framework is more convenient and natural for representing complicated objects, which have multiple semantic meanings. The multi-instance multi-label learning task can be transformed into a multi-instance learning (MIL) task or a multi-label learning(MLL) task and can be further transformed into a traditional supervised learning task. Based on these two solutions, two basic algorithms have been proposed, i.e., MIMLBOOST and MIMLSVM, and have been successfully applied to many classification problems.

The existing multi-instance multi-label(MIML) algorithms have a common assumption that all instances in a bag are independent of each other. However, it is difficult to guarantee the independence assumption of instances in many practical applications. Thus, the instance correlations cannot be ignored. In addition, there are some issues in MIMLBOOST and MIMLSVM. For MIMLBOOST, when using the multi-instance boosting (MIBOOSTING) method [40] for multi-instance task, the way of assigning the bag's label to each of its instances without considering that the positive bags may contain some negative instances which will cause a large error. For MIMLSVM, it is problematic to employ the Hausdorff distance, which is effective to represent the distance between two instances, to measure the similarity between two bags of instances. For example, if a negative instance in a positive bag is very similar to an instance in a negative bag, adopting the Hausdorff distance can result in a classification error.

In this paper, a novel multi-instance multi-label learning algorithm based on instance correlations (IC-MIML) is proposed. An overview of the proposed IC-MIML algorithm is illustrated in Fig. 1.

The algorithm proceeds in three steps. First, a graph is constructed to describe instance correlations in a bag and mapped to a high-dimensional space to represent features in the bag. At the same time, a multi-instance bag is transformed into a single-instance sample, which avoids not only the problem of labeling instances in the bag, but also the error caused by measuring the distance between two bags of points with Hausdorff distance. Second, IC-MIML employs a concept of multi-kernel fusion to predict different labels according to graphs at different scales by constructing multiple kernel functions according to different parameters and graphs at different scales. A convex combination of basis kernels is employed to implement the multi-kernel fusion. Finally, SVM is employed to perform classification and prediction for image scene classification.

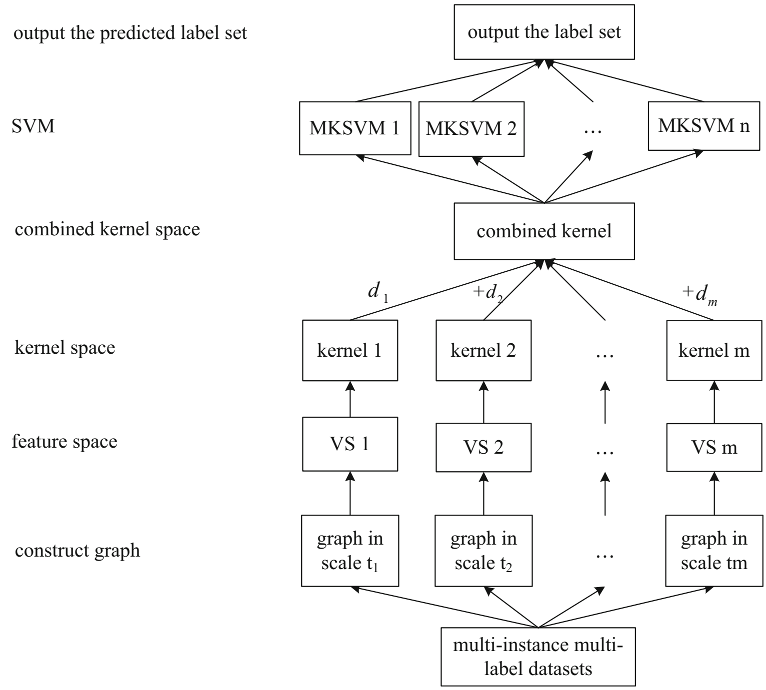

Fig. 1 The system overview of IC-MIML

2 Related work

Multi-instance learning is a deformation of supervised learning and is regarded as a new learning framework in parallel with supervised learning, unsupervised learning, and reinforcement learning. The concept of multi-instance learning was first introduced by Dietterich et al. [9] in 1997, when studying drug activity prediction. Each isomer of a molecule is regarded as an instance, and a set of isomers of a molecule is regarded as a bag. A bag is labeled as a positive bag if there is at least one instance is positive (one isomer of the molecule is suitable for pharmaceutical); otherwise, it is labeled as a negative bag. Through this framework, the problem of drug activity prediction is described more accurately.

2.1 Multi-instance learning

The first multi-instance learning (MIL)algorithm is the axis-parallel-rectangle learning algorithm [9], which adopts iterative search strategy to find the minimum axis-parallel-rectangle containing at least one instance in each positive bag. Then, many multi-instance learning algorithms were proposed.

Maron et al. [25] proposed the Diverse Density (DD) algorithm, where a point in attribute space with the maximum diverse density is found by gradient descent method. Zhang et al. [51] introduced the expectation maximization (EM) algorithm into DD algorithm and proposed the EM-DD algorithm. Wang and Zucker [36] extended the k-nearest neighbor algorithm with the modified Hausdorff distance and proposed the Bayesian-kNN and CitationkNN algorithm. Ruffo [30] modified the decision tree algorithm C4.5 and proposed the Relic algorithm. Andrews et al. [1] applied SVM to multi-instance learning and proposed MI-SVM and mi-SVM algorithms. These two algorithms intend to identify a maximal-margin hyperplane for the bags or the instances. Gärtner [14] took multi-instance bags as a set of feature vectors, which were directly used to build the multi-instance kernel (MI-Kernel) function, to solve the multi-instance problem. Chen et al. [6] transformed the multi-instance learning task into the supervised learning task and proposed DD-SVM algorithm. By building the projection space, bags are mapped to a point in space; then the classifier is trained by SVM. Considering noise caused by the negative instances in positive bags, the Multiple-Instance Learning via Embedded Instance Selection algorithm (MILES) [5] was proposed, which maps bag features into an instance space and trains a l_1 -SVM. Fu et al. [12] improved MILES and proposed Multiple-Instance Learning with Instance Selection (MILIS). MILIS builds a distribution model of negative instances by a Gaussian kernel density estimation function, and selects the instance, whose likelihood is the smallest in a positive bag and the largest in a negative bag as prototypes. Then instance update and SVM training are performed alternatively.

Recently, semi-supervised-based multi-instance learning received an increasing attention. Rahmani et al. [28] proposed the Multiple Instance Semi-Supervised Learning (MISSL)algorithm. Based on the framework of MISSL, Zhou et al. [59] proposed the Multi-Instance learning by Semi-Supervised Support Vector Machine (MissSVM) algorithm, Wang et al. [35] proposed the Graph-based Multiple-Instance Learning (GMIL) algorithm by using three kinds of data, i.e. labeled data, semi-labeled data, and unlabeled data, which simultaneously propagate information on a graph. Zhou et al. [58] proposed MIGraph and miGraph methods, where instances are modeled as non-independent identically distributed and a graph is constructed to represent instance correlations in a bag. Deselaers et al. [8] proposed a multi-instance learning algorithm based on conditional random field, where a bag is modeled as a node and an instance is modeled as a state. Wang et al. [34] proposed a Maximum Margin Multi-instance Learning (M^3L) approach, where the class-to-bag distance is parameterized using the class specific distance metrics and the multi-instance learning problem is solved by solving two maximum margin optimization problems. Li [23] used a sparse ε-graph to construct the inner contextual structure among instances in the same bag and proposed a multi-instance algorithm based on a hierarchical sparse representation.

Liu et al. [24] developed a voting framework to solve key instance detection by exploiting the relationships among instances. Feng [11] proposed a semi-supervised multi-instance learning algorithm with a hierarchical sparse representation. Hu et al. [15–17, 43, 44] proposed a learning discriminative pattern framework for processing surveillance videos.

2.2 Multi-instance multi-label learning

Multi-instance learning techniques have been applied to a variety of applications including image categorization [6], image retrieval [52], text classification [32, 52], web mining [57], spam detection [21], face detection [33, 53], and computer-aided medical diagnosis [13]. These algorithms are mainly used for predicting a single label of the multi-instance bag. However, none of them are used for predicting multiple labels of the multi-instance bag.

Considering the diversity of object's semantics in the real application, Zhou et al. [61, 62] first extended multi-instance learning in a multi-label classification task, and proposed the Multi-Instance Multi-Label learning (MIML) framework, which has become a new research hotspot. In the MIML framework, a bag consists of multiple instances is associated with multiple labels. For example, Fig. 2a contains building, sky, trees, and grass, and thus, it has labels of multiple classes, i.e., building, sky, trees and grass. This image can be segmented into multiple regions shown in Fig. 2b. Obviously, the multi-instance learning framework is difficult to adapt to this situation; while the MIML framework is more suitable.

The multi-instance multi-label learning problem can be transformed into a multi-instance learning task or a multi-label learning task and further transformed into a traditional supervised learning task. Based on this degeneration strategy, Zhou et al. [32, 52] proposed the MIMLBOOST algorithm and the MIMLSVM algorithm and successfully applied them to image scene classification and text classification problems. MIMLBOOST firstly decomposes each multi-instance multi-label sample into multiple multi-instance single-label samples, and then trains a multi-instance classifier for each label by MIBOOSTING algorithm [40]. By replacing the MIBOOSTING by MI-SVM [2] in MIMLBOOST, MIMLSVMmi [62] is proposed. In MIMLSVM, k-medoids clustering is performed on the data set by computing distances between a multi-instance sample and each cluster center. Then, each sample can be represented by a k-dimensional feature vector. A multi-label classifier can be trained from the transformed samples with MLSVM algorithm. By replacing the MLSVM with a two-layer neural network structure [55] in MIMLSVM, MIMLNN [62] is proposed. The degeneration process may lose information. Zhou et al. [62] proposed the Direct MIMLSVM (D-MIMLSVM) algorithm, which tackles MIML problems directly in a regularization framework. Zha et al. [49] proposed an integrated multi-label multi-instance learning (MLMIL) approach based on hidden conditional random fields (HCRFs) for MIML image annotation. By simultaneously modeling the relationships between the labels and regions as well as the correlations among the multiple labels, the MLMIL method solves the multi-label and multi-instance problem in an integrated manner. A generative model for multi-instance multi-label learning was proposed by Yang et al. [48]. Zhang [50] proposed a k-nearest neighbor approach (MIML-kNN) for multi-instance multi-label learning. Given a test example, MIML-kNN considers not only its neighbors, but also its citers, which regard the test sample as their own neighbors. The label set of the test example is determined using the labeling information conveyed by its neighbors and citers. Xu et al. [42] used multi-instance multi-label framework to solve the video annotation task and proposed an ensemble multi-instance multi-label learning approach named En-MIMLSVM considering the class imbalance and training efficiency of most video annotation tasks. In addition, a temporally consistent weighted multi-instance kernel is developed by modeling both the temporal consistency in video data and the significance of instances at different

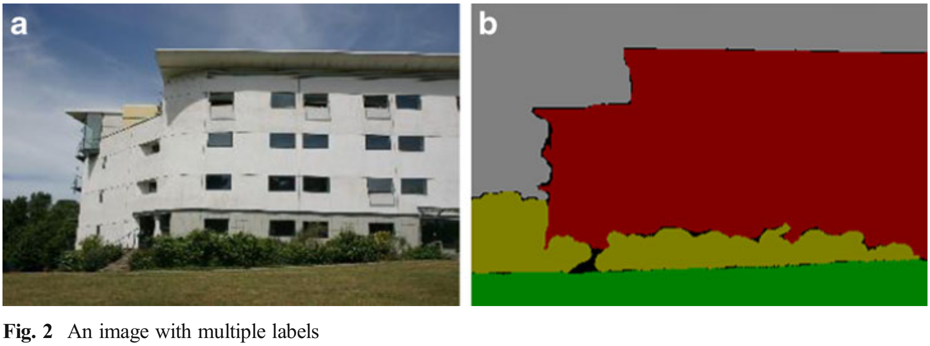

Fig. 2 An image with multiple labels

levels in a pyramid representation. Li et al. [22] proposed a MIML algorithm named Key Instances Sharing among Related labels (KISAR) to discover key instances that trigger different labels in a bag. KISAR first sets key instances as cluster centers via k-means and maps each bag to these key instances with a similarity function. By this means, a bag is represented as a feature vector. By solving a convex optimization problem, KISAR finally gets the prediction function oriented to all labels. Selecting key instances by clustering is not a trivial problem and can be easily affected by noise. In addition, the classifier is sensitive to the number of clusters, which can cause the instability of classifier. Briggs et al. [4] proposed the Rank-Loss Support Instance Machines (RankLossSIM) to optimize the label ranking loss for bag and instance annotation. The loss function is measured at the bag-level and encourages the correct ranking of classes such that scores of classes present in the label set of a bag are higher than those that are not present. This objective can be instantiated with different aggregation models including a max model or a softmax model. Both models can be viewed as representing each bag with a support instance for each class. It alternates between computing the support and optimizing a convex rank-loss objective using an efficient primal sub-gradient descent method. RankLossSIM focuses on ranking loss minimization, but the accuracy of label annotation still needs to improve.

Yang et al. [46] proposed the MIMLwel (Multi-Instance Multi-Label Learning with Weak Label)with assumptions that highly relevant labels generally share common instances and the underlying class means of bags for each label are with a large margin. The MIMLwel first explores a mapping from a bag of instances to cluster centers and gets a feature vector, where each element measures the degree of the bag being associated with a group of similar instances. Then, it employs sparse predictors to learn the labels of bags such that the class means of bags for each label is maximized. Finally, the problem is solved with an efficient block coordinate descent solution. To efficiently handle large data sets, the MIMLfast [18] approach was proposed, which first constructs a low-dimensional subspace shared by all labels, and then trains label-specific linear models to optimize approximated ranking loss via stochastic gradient descent. More recently, MIML is applied into video annotation [41], protein function prediction [39] and so on.

3 IC-MIML method

3.1 A brief review on MIML

Let χ denote the instance space and Y denote the set of class labels. Then, the MIML learning is defined as:

To learn a prediction function $f: 2^x \rightarrow 2^y$ from a given data set $\{(X_1, Y_1), (X_2, Y_2), \cdots, (X_m, Y_m)\}$, where $X_i \subseteq \chi$ is a set of instances $\{x_{i1}, x_{i2}, \cdots, x_{in_i}\}$, $x_{ij} \in \chi (j = 1, 2, \cdots, n_i)$ and $Y_i \subseteq Y$ is a set of labels $\{y_{i1}, y_{i2}, \cdots, y_{il_i}\}$, $y_{ik} \in Y (k = 1, 2, \cdots, l_i)$. Here, n_i denotes the number of instances in X_i; and l_i denotes the number of labels in Y_i.

The MIML framework is shown in Fig. 3. A bag is represented by several instances. By the prediction function $f: 2^x \rightarrow 2^y$, the label set of a bag can be predicted. The main goal of multi-instance multi-label learning is to develop an accurate prediction function, a classifier, given a train data set.

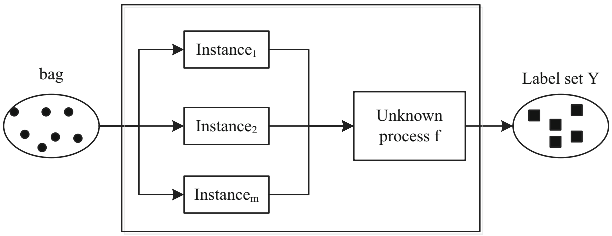

Fig. 3 An illustration of the MIML framework

3.2 Construction of the instance correlations graph

As indicated by Zhou and Xu [60], previous studies on multi-instance learning typically treated the instances in the bags as independently and identically distributed, which neglects the fact that the relationships among the instances convey important structure information. Considering the above image annotation task (Fig. 2) again, modeling the different image parts as inter-correlated samples is more meaningful than treating them as unrelated samples. Actually, the instances in a bag are rarely independent, and thus, a better performance can be expected if the relationships among instances in a bag are considered as the bag features.

In the real world, many objects have inherent structures, which can be used as features to represent objects. For example, Fig. 4a is divided into several areas, as shown in Fig. 4b. The areas in the white box must have strong relationships. Assume that an image corresponds to a bag and each region in the segmentation result corresponds to an instance in the bag. Strong relationships among regions mean that some instances in the bag are highly related. The correlations can be seen as the image features. Motivated by this observation, the proposed method introduces the instance correlations in a bag into the multi-instance multi-label learning.

Inspired by successful applications of MIGraph and miGraph proposed by Zhou [58], IC-MIML adopts the same idea of constructing a graph to model the instance correlations in a bag.

For a bag $X_i \subseteq \chi$, every instance is regarded as a node. Then, the distance of each pair of nodes, e.g., x_{ia} and x_{iu} is computed. If the distance between x_{ia} and x_{iu} is smaller than a predefined threshold, an edge is established between these two nodes, with a weight representing the affinity of the two nodes.

An affinity matrix W^i, which can be regarded as a graph, can be derived to capture the instance correlations in a bag. The size of affinity matrix W^i is $n_i \times n_i$. The element at the a^{th}

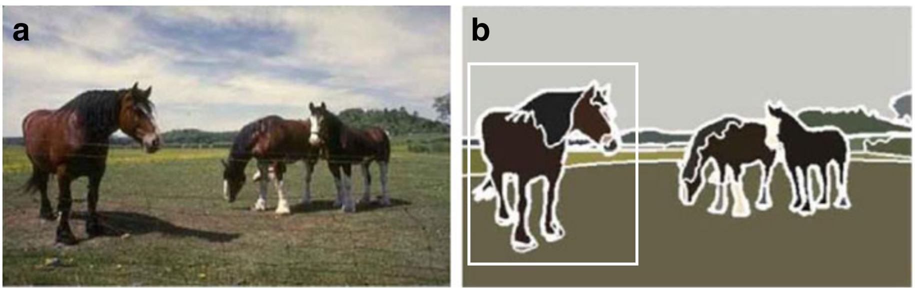

Fig. 4 The result of image segmentation

row and the u^{th} column of W^i, denoted by w^i_{au}, models the correlation between the two instances x_{ia} and x_{iu} is defined as:

$$\begin{matrix} w^i_{au} = 1, \mathit{if}\ d(x_{ia}, x_{iu}) < \delta; \\ w^i_{au} = 0, \mathit{if}\ d(x_{ia}, x_{iu}) \geq \delta. \end{matrix} \tag{1}$$

If the distance between the instance x_{ia} and the instance x_{iu} is smaller than the predefined threshold δ, w^i_{au} is set to 1; otherwise, it is set to 0. Many distance metrics can be used to compute the distances. In this work, we choose the Gaussian distance.

In this way, each bag corresponds to an affinity matrix, which can represent the bag's features. Graphs at different scales can be constructed with different thresholds δ. Different scales of the graph describe different degrees of instance correlations.

3.3 Graph kernel function based on the correlations matrix

However, the affinity matrix describing the instance correlations in a bag cannot be directly used for classification, After mapping the training bags to a set of graphs, we need to map each graph to a vector in a high-dimensional space to represent the bag's features. Hence, we need to use a kernel to capture the similarity among graphs and then solve classification problems by kernel methods such as SVM.

Given two multi-instance bags X_i and X_j which contains n_i and n_j instances, respectively. The graph kernel function is defined as follows [58].

$$k_g(X_i, X_j) = \frac{\sum_{a=1}^{n_i} \sum_{b=1}^{n_j} W_{ia} W_{jb} k(x_{ia}, x_{jb})}{\sum_{a=1}^{n_i} W_{ia} \sum_{b=1}^{n_j} W_{jb}} \tag{2}$$

where W_{ia} and W_{jb} can be obtained based on the affinity matrix W^i and the affinity matrix W^j. Specifically, $W_{ia} = 1/\sum_{u=1}^{n_i} w^i_{au}$ and $W_{jb} = 1/\sum_{v=1}^{n_j} w^j_{bv}$. $k(x_{ia}, x_{jb})$ is obtained by Gaussian radial basis function (RBF) kernel function such that $k(x_{ia}, x_{jb}) = exp(-\gamma \| x_{ia} - x_{jb} \|^2)$. Then, we can perform classification and prediction by the SVM method based on kernels.

3.4 Multi-Kernel fusion for multi-instance multi-label learning

In a multi-label task, the distribution of samples is different for different kinds of labels. It is not appropriate to use a single kernel function for the multi-label task because it use the same mapping function for different kinds of label prediction. Therefore, multi-kernel learning (MKL) is introduced to solve the multi-instance multi-label learning based on instance correlations.

Kernel methods, such as SVMs have been proven to be effective for solving learning problems like classification or regression. The performance of the learning algorithm strongly depends on the kernel. But how to choose an appropriate kernel function to construct a more flexible model is still a problem. Multiple kernel learning aims to learn an appropriate kernel oriented to different kinds of data sets with the iterative optimization method.

Research has shown that using multiple kernels instead of a single one can enhance the interpretability of the decision function and improve performances. The original data set can be mapped to different feature spaces. MKL aims at combining multiple feature spaces with multiple kernels. Based MKL, the flexibility and accuracy of the algorithm is improved. An approach considering the kernel $K(x,x')$ as a convex combination of basis kernels was proposed in [29]:

$$K\left(x,x^{'}\right)=\sum_{m=1}^{M}d_mK_m\left(x,x^{'}\right),\ d_m\geq 0,\ \sum_{m=1}^{M}d_m=1 \tag{3}$$

where $K_m(x,x')$ is the basis kernels, M is the total number of basis kernels, d_m is the weights. The basis kernels $K_m(x,x')$ can use some classical kernels (such as Gaussian kernels) with different parameters. Figure 5 shows a process of combining multiple kernels linearly. Particularly, convex combination is a special form of linear combination. Within the multi-kernel framework, the problem of data representation through the kernel is then transferred to the choice of weights d_m. We can initialize d_m, and get the optimal value by iterative optimization method.

Considering different kinds of labels corresponding to graphs at different scales, IC-MIML constructs several graphs at different scales and derives different basis kernels with different parameters γ. Then, the combined kernel can be obtained as a convex combination of basis kernels. For different kinds of labels $y\in Y$, graph kernel matrices with different parameters and the corresponding graphs are treated as basis kernels. We can get different basis kernels according to Eq. (2). For example, W_{ia} and W_{jb} can be changed by changing the threshold δ for constructing the affinity matrix; and $k(x_{ia},x_{jb})$ can be changed by changing the parameter γ.

An MKSVM h_y can be trained with the SimpleMKL method [29], which is used to solve the multiple kernel learning. Different kinds of labels correspond to different MKSVMs. These MK-SVMs can be used to predict labels of unknown bags. Algorithm 1 summarizes the process of the IC-MIML algorithm. By multi-kernel fusion, the flexibility of the algorithm is greatly increased, the algorithm is more suitable for multi-label learning. As a result, it can deal with many complicate situations such as a heterogeneous sample set.

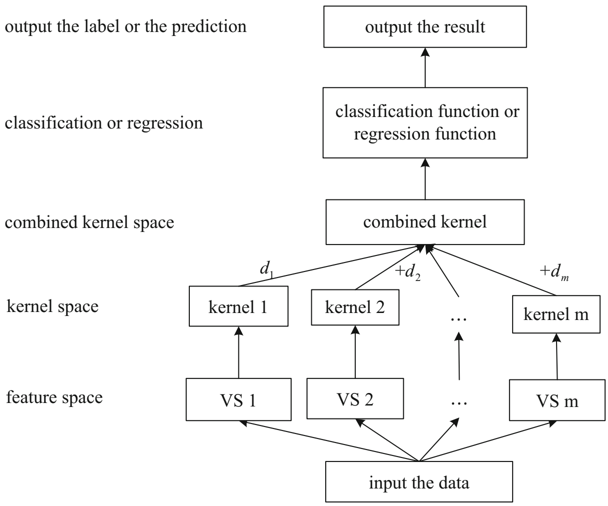

Fig. 5 A process of generating a combined kernel as a linear combination of basis kernels

Algorithm 1 The IC-MIML algorithm

Inputs:

Ω : the MIML training set $\{(X_1, Y_1), (X_2, Y_2), \cdots, (X_m, Y_m)\}$

Y : the label set

X^* : the test MIML sample

threshold: the threshold array for constructing graphs

gam: the parameter array for Gaussian kernel functions

Outputs:

Y^* : the predicted label set of X^*

Process:

(1) For each sample $(X_u, Y_u)(u = 1,2,\cdots.m)$, remove each labels of it to get a data set without labels $\Gamma = \{X_u \mid u = 1,2,\cdots,m\}$. X_u is a set of instance $\{x_{u1}, x_{u2}, \cdots, x_{u,n_u}\}$.

(2) Repeat for $t = 1,2,\cdots,L$ iterations:

2a For each bag X_u , build the affinity matrices W^u in different scales, the threshold is $\delta = threshold(l)(l = 1,2,\cdots,L)$.

2b According to kernel function k_g , compute the kernel matrix k_{train} corresponding to the affinity matrix in different scales, the parameter of the kernel is $\gamma = gam(l)(l = 1,2,\cdots,L)$.

$$k_g(X_i, X_j) = \frac{\sum_{a=1}^{n_i}\sum_{b=1}^{n_j} W_{ia} W_{jb} k(x_{ia}, x_{jb})}{\sum_{a=1}^{n_i} W_{ia} \sum_{b=1}^{n_j} W_{jb}}$$

(3) For each kind of labels $y \in Y$, use simpleMKL method to train a MKSVM classifier h_y , taking the kernel matrix k_{train} as basis kernel.

(4) Return $Y^* = \{\arg\max_{y \in Y} h_y(k(X^*))\} \cup \{y \mid h_y(k(X^*)) \geq 0, y \in Y\}$ ($k(X^*)$ is the kernel matrix of X^*).

4 Experiments and analysis

4.1 Experimental data sets

To evaluate the performance of the proposed method in scene classification, we evaluated it on Bird Song [4] data set, MSRC v2 data set [37], Scene data set [62] and Text data set [62].

Fig. 6 Sample images taken from MSRC v2 data set

Figures 6 and 7 give sample images taken from MSRC v2 data set and Scene data set, respectively. Table 1 summarizes the properties of each dataset used in our experiments.

The Bird Song data set [4] contains 548 audio recordings of bird song belonging to 13 classes corresponding to different species. These audio recordings of bird song are collected at the H.J. Andrews (HJA) Experimental Forest, using unattended microphones. A 10-s audio recording is regarded as a bag with labels corresponding to the set of species present in the recording. The 10-s audio recording is converted to a spectrogram. A series of preprocessing steps are then applied to the spectrogram to reduce noise and to identify bird song segments in the audio [27]. Each segment is considered as an instance and described by a 38-dimensional feature vector characterizing the shape of the segment, its time and frequency statistics, and a histogram of gradients.

A subset of the Microsoft Research Cambridge (MSRC) image dataset [38], named MSRC v2 data set, contains 591 images and 23 classes. The MSRC v2 dataset has been used for the instance annotation problem withpixel-level labels provided. Because there are many images containing several classes, an MIML dataset was constructed from MSRC v2 and employed in MIML experiments [45, 49]. Specifically, each image is treated as a bag. The bag label is a list of all classes present in the ground-truth segmentation result (i.e. the union of the instance labels). Each instance corresponds to each contiguous region in the ground-truth segmentation result and is described by a 16-dimensional histogram of gradients and a 32-dimensional histogram of colors. Hence, each instance is described by a 48 dimensional feature vector. In this way, the data set can be used in multi-instance multi-label learning.

The Scene data set consists of 2000 natural scene images including the classes such as desert, mountains, sea, sunset, and trees. There are over 22 % of images containing multiple classes. Each image has been segmented into 9 blocks by the SBN method [26], which uses a Gaussian filter to smooth the image and then subsamples the image to an 8×8 matrix of color blobs, where each blob is a 2×2 set of pixels. An instance corresponds to a combination of a single blob with its four neighboring blobs (up, down, left, and right) and is described by 15 features. The first three features represent the mean R, G, B values of the central blob; and the remaining twelve features represent the differences between the central blob and other four neighboring blobs in terms of mean color values, respectively.

The Text data set consists of 2000 documents. There are over 14.9 % documents marked with multiple labels of 7 labels. Each document is represented by 2 to 26 instances in a bag by a sliding window technique [2]. The instances are obtained by splitting each document into passages using overlapping windows of maximal 50 words. As a result, there are 2000 bags and

Fig. 7 Sample images taken from Scene data set

Table 1 Data sets used in experiments

Data set	Bags	Instances	Dimension	labels	labels/bags
Bird Song	548	10,232	38	13	2.1
MSRC v2	591	1758	48	23	2.5
Scene	2000	18,000	15	5	1.2
Text	2000	7119	243	7	1.2

the number of instances in each bag varies from 2 to 26 (3.6 on average). The instances are represented based on term frequency. The words with high frequencies are considered, excluding "function words" that have been removed from the vocabulary using the Smart stop-list [31]. It has been found that the dimensionality of the data set can be reduced to 1–10 % without loss of effectiveness based on document frequency [47]. In this work, we use the top 2 % frequent words. Therefore, each instance is represented by a 243-dimensional feature vector.

4.2 Evaluation criteria

In traditional supervised learning, accuracy, defined as the percentage of text samples correctly predicted, is often used to evaluate the performance for prediction of one class label. Accuracy is. But for multi-label task predicting multiple class labels, only using accuracy is no longer persuasive. Five criteria are often used for evaluating the performance of multi-label task, including hamming loss, one-error, coverage, ranking loss, and average precision. The smaller the value of the first four criteria is, the better the performance of the method has. The larger the average precision is, the better the performance of the method has. In addition to the 5 criteria, there are two new multi-label evaluation criteria, i.e., average recall and average F1 [62]. The average recall evaluates the average fraction of labels that have been correctly predicted. The average F1 expresses a tradeoff between the average precision and the average recall. The larger the value of these two criteria is, the better the performance of the method has. In this paper, we adopt all of the 7 criteria to evaluate the performance of the method and analyze the advantages and disadvantages of the method.

4.3 Experimental results and analysis

For the MSRC v2 data set, 2/3 of the data are randomly selected for training, and the remaining data are used for testing. For the other three data sets, 3/4 of the data are randomly used for training while the remaining 1/4 of the data are used as testing examples. We repeat experiments for 30 runs, and compute the average and standard deviation. For the MSRC v2 data set, the threshold array for constructing graphs is set to $threshold = [0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2]$, while the parameter array for Gaussian kernel functions is $gam = [0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2]$. For the Scene data set, $threshold = [0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2]$ and $gam = [0.2, 0.8, 1.6, 200, 3.2, 5.6, 500, 7, 9, 100]$. For the Bird Song data set and the Text data set, $threshold = [1.2, 1.4, 1.6, 1.8, 2]$ and $gam = [0.2, 0.4, 0.6, 0.8, 1]$.

4We further compare the IC-MIML algorithm with several state-of-the-art MIML algorithms including MIMLBOOST [62], MIMLSVM [62], MIMLSVM$_{mi}$ [62], MIMLNN [62], MIMLfast [18], and KISAR [22], as well as an algorithm for single-instance multi-label learning, ML-kNN [56]. For MIMLfast, the step size is $\gamma_t = \gamma_0/(1 + \eta\gamma_0 t)$, where $\gamma_0 = 0.005$, $\eta = 10^{-5}$; and the upper bound of norm is set to 1. The number of boosting iterations in MIMLBOOST is set to 50. The

Table 2 Results (mean ± std.) on Bird Song data set

Compared Algorithms	Evaluation Criteria						
	hamming loss ↓	one-error↓	coverage ↓	ranking loss ↓	average precision ↑	average recall ↑	average F1 ↑
IC-MIML	**0.053±0.002**	0.054±0.014	**1.679±0.232**	**0.027±0.004**	**0.938±0.003**	**0.877±0.016**	**0.907±0.010**
MIMLBOOST	0.139±0.006	0.365±0.031	3.105±0.111	0.126±0.009	0.682±0.032	0.213±0.019	0.324±0.023
MIMLSVM	0.061±0.006	0.088±0.018	1.739±0.156	0.031±0.006	0.921±0.017	0.833±0.020	0.875±0.019
MIMLSVM*mi*	0.067±0.006	0.054±0.021	2.754±0.394	0.075±0.020	0.889±0.024	0.837±0.024	0.862±0.023
MIMLNN	0.068±0.002	0.082±0.019	1.989±0.035	0.041±0.001	0.907±0.006	0.831±0.013	0.867±0.008
MIMLfast	0.092±0.012	0.136±0.047	2.367±0.406	0.061±0.020	0.864±0.037	0.669±0.067	0.753±0.055
KISAR	0.054±0.005	**0.048±0.011**	1.736±0.153	0.028±0.005	0.937±0.008	0.804±0.018	0.865±0.013
MLkNN	0.081±0.007	0.150±0.029	2.162±0.145	0.059±0.007	0.864±0.014	0.695±0.035	0.770±0.025

Table 3 Results (mean ± std.) on MSRC v2 data set

Compared Algorithms	Evaluation Criteria						
	hamming loss ↓	one-error↓	coverage ↓	ranking loss ↓	average precision ↑	average recall ↑	average F1 ↑
IC-MIML	**0.070 ± 0.003**	0.210 ± 0.021	**4.656 ± 0.270**	**0.085 ± 0.008**	**0.770 ± 0.015**	**0.658 ± 0.019**	**0.710 ± 0.015**
MIMLBOOST	0.113 ± 0.004	0.526 ± 0.035	11.144 ± 0.125	0.361 ± 0.025	0.368 ± 0.017	0.046 ± 0.009	0.082 ± 0.015
MIMLSVM	0.081 ± 0.004	0.303 ± 0.030	5.508 ± 0.449	0.115 ± 0.012	0.704 ± 0.019	0.435 ± 0.026	0.537 ± 0.024
MIMLSVM*mi*	0.083 ± 0.004	0.331 ± 0.026	7.665 ± 0.399	0.167 ± 0.011	0.627 ± 0.021	0.364 ± 0.029	0.460 ± 0.027
MIMLNN	0.073 ± 0.002	0.245 ± 0.019	5.256 ± 0.178	0.099 ± 0.005	0.743 ± 0.007	0.466 ± 0.012	0.573 ± 0.010
MIMLfast	0.105 ± 0.006	0.392 ± 0.041	6.973 ± 0.368	0.156 ± 0.012	0.601 ± 0.023	0.361 ± .034	0.451 ± 0.032
KISAR	0.077 ± 0.003	**0.084 ± 0.017**	13.348 ± 0.558	0.565 ± 0.026	0.433 ± 0.018	0.295 ± 0.020	0.351 ± 0.020
MLkNN	0.088 ± 0.003	0.347 ± 0.031	6.133 ± 0.286	0.141 ± 0.011	0.648 ± 0.020	0.306 ± 0.027	0.415 ± 0.026

Table 4 Results (mean ± std.) on Scene data set

Compared Algorithms	Evaluation Criteria						
	hamming loss ↓	one-error↓	coverage ↓	ranking loss ↓	average precision ↑	average recall ↑	average F1 ↑
IC-MIML	**0.172 ± 0.003**	**0.310 ± 0.008**	**0.918 ± 0.026**	**0.161 ± 0.005**	**0.801 ± 0.006**	**0.597 ± 0.008**	**0.684 ± 0.007**
MIMLBOOST	0.191 ± 0.011	0.347 ± 0.013	1.002 ± 0.006	0.186 ± 0.001	0.776 ± 0.005	0.449 ± 0.001	0.568 ± 0.002
MIMLSVM	0.213 ± 0.007	0.419 ± 0.017	1.198 ± 0.029	0.236 ± 0.010	0.726 ± 0.011	0.501 ± 0.018	0.593 ± 0.016
MIMLSVM*mi*	0.199 ± 0.006	0.461 ± 0.023	1.396 ± 0.076	0.281 ± 0.018	0.689 ± 0.015	0.311 ± 0.020	0.428 ± 0.021
MIMLNN	0.200 ± 0.004	0.395 ± 0.011	1.143 ± 0.030	0.221 ± 0.006	0.741 ± 0.007	0.466 ± 0.008	0.572 ± 0.007
MIMLfast	0.242 ± 0.008	0.489 ± 0.021	1.356 ± 0.065	0.272 ± 0.017	0.682 ± 0.014	0.428 ± 0.017	0.526 ± 0.017
KISAR	0.175 ± 0.006	0.322 ± 0.019	0.974 ± 0.045	0.174 ± 0.010	0.790 ± 0.011	0.494 ± 0.018	0.608 ± 0.015
MLkNN	0.201 ± 0.007	0.407 ± 0.020	1.149 ± 0.050	0.220 ± 0.011	0.737 ± 0.012	0.359 ± 0.033	0.482 ± 0.031

Table 5 Results (mean ± std.) on Text data set

Compared Algorithms	Evaluation Criteria						
	hamming loss ↓	one-error↓	coverage ↓	ranking loss ↓	average precision ↑	average recall ↑	average F1 ↑
IC-MIML	**0.037 ± 0.002**	**0.074 ± 0.006**	**0.313 ± 0.028**	**0.025 ± 0.003**	**0.953 ± 0.003**	**0.905 ± 0.006**	**0.928 ± 0.004**
MIMLBOOST	0.087 ± 0.007	0.146 ± 0.022	0.700 ± 0.067	0.107 ± 0.011	0.839 ± 0.016	0.578 ± 0.026	0.684 ± 0.024
MIMLSVM	0.085 ± 0.003	0.238 ± 0.013	0.776 ± 0.037	0.099 ± 0.007	0.843 ± 0.008	0.730 ± 0.011	0.783 ± 0.009
MIMLSVM*mi*	0.063 ± 0.008	0.097 ± 0.018	0.375 ± 0.064	0.035 ± 0.007	0.937 ± 0.012	0.890 ± 0.021	0.913 ± 0.014
MIMLNN	0.039 ± 0.003	0.076 ± 0.006	0.321 ± 0.016	0.026 ± 0.002	0.951 ± 0.004	0.871 ± 0.010	0.910 ± 0.007
MIMLfast	0.081 ± 0.005	0.205 ± 0.019	0.829 ± 0.092	0.106 ± 0.015	0.855 ± 0.014	0.726 ± 0.016	0.785 ± 0.015
KISAR	0.071 ± 0.034	0.168 ± 0.139	0.553 ± 0.392	0.064 ± 0.065	0.893 ± 0.088	0.633 ± 0.220	0.719 ± 0.235
MLkNN	0.091 ± 0.004	0.287 ± 0.019	0.877 ± 0.051	0.117 ± 0.009	0.811 ± 0.012	0.602 ± 0.026	0.691 ± 0.019

number of partitions in MIMLSVM is set to be 20 % of the number of the training set. For ML-kNN, each image is represented by a feature vector, which contains the average feature values of all instances in the bags. The number of nearest neighbors is set to 10.

The comparison results on the four data sets are reported in Tables 2, 3, 4 and 5, where the best results are in bold. As shown in the experimental results on the four data sets, IC-MIML achieves the best performance in most cases. IC-MIML is significantly better than the other algorithms in terms of all criteria except one-error on the Bird Song data set and the MSRC v2 data set. KISAR is better than IC-MIML for one-error and achieves comparable results with IC-MIML for most evaluation criteria. MIMLBOOST yields the worst performance. Experimental results on the Scene data set and the Text data set, shown in Tables 4 and 5 demonstrate that the proposed IC-MIML algorithm is apparently better than the others. KISAR achieves comparable results with IC-MIML on the Scene data set, while MIMLSVMmi achieves comparable results with IC-MIML on the Text data set. MIMLBOOST yields the worst performance on the Text data set.

The number of classes in the Bird Song data set and the MSRC v2 data set is larger compared to the other two data sets is small. Obviously, IC-MIML achieves good performance on both data sets with many classes and data sets with few classes. In conclusion, the IC-MIML method greatly improves the performance of multi-label multi-instance task, compared with other methods.

The improvement is primarily due to two reasons. On the one hand, the instance correlations can represent bags more comprehensively, which improves the classification accuracy. On the other hand, multi-kernel fusion improves the flexibility of the algorithm and makes it more suitable for multi-label learning.

Though IC-MIML offers many advantages, the process of constructing graphs and the introduction of multi-kernel fusion increases the complexity of the algorithm. Therefore, how to reduce the complexity needs to be addressed.

5 Conclusion

It is more suitable and more flexible to predict multiple labels for complex datasets and more importantly, the real world. In this paper, a multi-instance multi-label learning algorithm, IC-MIML, which explicitly models instance correlations, was proposed. Specifically, the instance correlations are modeled by graphs; and multi-kernel fusion is introduced into classifier constructions. Experimental results on four data sets show that the performance of the proposed algorithm is superior to the state-of-the-art multi-instance multi-label algorithms on multi-label classification. Some future directions of IC-MIML can be explored. First, the complexity of IC-MIML should be reduced by improving the process of constructing graphs and multi-kernel fusion. Second, we plan to generalize the instance correlations to other multi-instance learning and multi-instance multi-label learning algorithms.

Acknowledgments This work is supported by the National Science Foundation of China (No.61170161, No.61300155, No.61303086), Shandong Province Scholarship Council, Ph.D. Programs Foundation of Ludong University (No. LY2014033).

References

1. Andrews S, Hofmann T, Tsochantaridis I (2002) Multiple instance learning with generalized support vector machines. Proceedings of the 18th national conference on artificial intelligence, pp 943–944
2. Andrews S, Tsochantaridis I, Hofmann T (2003) Support vector machines for multiple-instance learning. Adv Neural Inf Proces Syst 15:561–568

3. Boutell MR, Luo J, Shen X et al (2004) Learning multi-label scene classification. Pattern Recogn 37(9): 1757–1771
4. Briggs F, Fern XZ, Raich R (2012) Rank-loss support instance machines for MIML instance annotation. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 534–542
5. Chen YX, Bi JB, Wang JZ (2006) MILES: multiple-instance learning via embedded instance selection. IEEE Trans Pattern Anal Mach Intell:1931–1947
6. Chen YX, Wang JZ (2004) Image categorization by learning and reasoning with regions. J Mach Learn Res 5(8):913–939
7. Chowdhury M, Kundu MK (2015) Comparative assessment of efficiency for content based image retrieval systems using different wavelet features and pre-classifier. Multimed Tools Appl 74(24):11595–11630
8. Deselaers T, Ferrari V (2010) A conditional random field for multiple-instance learning. Proceedings of the 27th International Conference on Machine Learning, pp 287–294
9. Dietterich TG, Lathrop RH, Lozano-Pérez T (1997) Solving the multiple instance problem with axis-parallel rectangles. Artif Intell 89(1–2):31–71
10. Dubey SR, Singh SK, Singh RK (2015) A multi-channel based illumination compensation mechanism for brightness invariant image retrieval. Multimed Tools Appl 74(24):11223–11253
11. Feng SH, Xiong WH, Li B, Lang CY, Huang XK (2014) Hierarchical sparse representation based multi-instance semi-supervised learning with application to image categorization. Signal Process 94(1):595–607
12. Fu Z, Robles-Kelly A, Zhou J (2011) MILIS: multiple instance learning with instance selection. IEEE Trans Pattern Anal Mach Intell 33(5):958–977
13. Fung G, Dundar M, Krishnappuram B, Rao RB (2007) Multiple instance learning for computer aided diagnosis. Adv Neural Inf Proces Syst 19:425–432
14. Gärtner T, Flach PA, Kowalczyk A, Smola AJ (2002) Multi-instance kernels. Proceedings of the 19th International Conference on Machine Learning, pp 179–186
15. Hu C, Bai X, Qi L, Wang X, Xue G, Mei L (2015) Learning discriminative pattern for real-time car brand recognition. IEEE Trans Intelli Transp Syst 16(6):3170–3181
16. Hu C, Xu Z et al (2014) Semantic link network based model for organizing multimedia big data. IEEE Trans Emerg Top Comput 2(3):376–387
17. Hu C, Xu Z et al (2015) Video structured description technology for the new generation video surveillance system. Front Comput Sci 9(6):980–989
18. Huang SJ, Gao W, Zhou ZH (2014) Fast multi-instance multi-label learning. Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp 1868–1874
19. Huang YG, Huang HY, Zhang J (2013) A noisy-smoothing relevance feedback method for content-based medical image retrieval. Multimed Tools Appl 73(3):1963–1981
20. Indyk W, Kajdanowicz T, Kazienko P (2013) Relational large scale multi-label classification method for video categorization. Multimed Tools Appl 65(1):63–74
21. Jorgensen Z, Zhou Y, Inge M (2008) A multiple instance learning strategy for combating good word attacks on spam filters. J Mach Learn Res 8:993–1019
22. Li YF, Hu JH, Jiang Y, Zhou ZH (2012) Towards discovering what patterns trigger what labels. Proceedings of the 26th AAAI Conference on Artificial Intelligence, pp 1012–1018
23. Li B, Xiong WH, Hu WM (2011) Context-aware multi-instance learning based on hierarchical sparse representation. IEEE International Conference on Data Mining, pp 370–377
24. Liu G, Wu J, Zhou ZH (2012) Key instance detection in multi-instance learning. Proceedings of the 4th Asian Conference on Machine Learning, pp 253–268
25. Maron O, Lozano-Pérez T (1998) A framework for multiple-instance learning. Adv Neural Inf Proces Syst 10:570–576
26. Maron O, Ratan AL (1998) Multiple-instance learning for natural scene classification. Proceedings of the 15th International Conference on Machine Learning, pp 341–349
27. Neal L, Briggs F, Raich R, Fern XZ (2011) Time-frequency segmentation of bird song in noisy acoustic environments. Proceedings of IEEE International Conference Acoustics, Speech, and Signal Processing, pp 2012–2015
28. Rahmani R, Goldman SA (2006) MISSL: multiple instance semi-supervised learning, Proceedings of the 23rd international conference on machine learning, pp 705–712
29. Rakotomamonjy A, Bach FR, Canu S, Grandvalet Y (2008) SimpleMKL. J Mach Learn 9(11):2491–2521
30. Ruffo G (2000) Learning single and multiple instance decision trees for computer security applications. University of Turin, Torino
31. Salton G (1989) Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison-Wesley, Reading
32. Settles B, Craven M, Ray S (2008) Multiple-instance active learning. Adv Neural Inf Proces Syst 20:1289–1296

33. Viola P, Platt J, Zhang C (2006) Multiple instance boosting for object detection. Adv Neural Inf Proces Syst 18:1419–1426
34. Wang H, Huang H, Kamangar F (2011) Maximum margin multi-instance learning. Proceedings of the 24th Neural Information Processing Systems (NIPS, 2011), pp 153–1613
35. Wang CH, Zhang L, Zhang HJ (2008) Graph-based multiple-instance learning for object-based image retrieval. Proceedings of the 1st ACM international conference on multimedia information retrieval, pp 156–163
36. Wang J, Zucker JD (2000) Solving the multiple-instance problem: A lazy learning approach. Proceedings of the 17th international conference on machine learning, pp 1119–1125
37. Winn J, Criminisi A, Minka T (2005) Object categorization by learned universal visual dictionary. Proceedings of the 10th IEEE International Conference on Computer Vision, pp 1800–1807
38. Winn J, Criminisi A, Minka T (2005) Object categorization by learned universal visual dictionary. Proceedings of 10th IEEE International Conference on Computer Vision, vol. 2. pp 1800–1807
39. Wu JS, Huang SJ, Zhou ZH (2014) Genome-wide protein function prediction through multi-instance multi-label learning. ACM/IEEE Trans Comput Biol Bioinformat 11(5):891–902
40. Xu X, Frank E (2004) Logistic regression and boosting for labeled bags of instances, Advances in knowledge discovery and data mining. Springer, Berlin Heidelberg, pp 272–281
41. Xu XS, Jiang Y, Xue XY, Zhou ZH (2012) Semi-supervised multi-instance multi-label learning for video annotation task. Proceedings of the 20th ACM International Conference on Multimedia, pp 737–740
42. Xu XS, Xue X, Zhou ZH (2011) Ensemble multi-instance multi-label learning approach for video annotation task. Proceedings of the 19th ACM international conference on Multimedia, pp 1153–1156
43. Xu Z et al (2015) Semantic based representing and organizing surveillance big data using video structural description technology. J Syst Softw 102:217–225
44. Xu Z et al (2016) Semantic enhanced cloud environment for surveillance data management using video structural description. Computing 98(1–2):35–54
45. Yakhnenko O (2009) Learning from text and images: generative and discriminative models for partially labeled data, PhD thesis, Iowa State University
46. Yang SJ, Jiang Y, Zhou ZH (2013) Multi-instance multi-label learning with weak label. Proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp 1862–1868
47. Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. Proceedings of the 14th International Conference on Machine Learning, pp 412–420
48. Yang S, Zha H, Hu B (2009) Dirichlet-bernoulli alignment: a generative model for multi-class multi-label multi-instance corpora. Adv Neural Inf Proces Syst 22:2143–2150
49. Zha ZJ, Hua XS, Mei T, Wang JD, Qi GJ, Wang ZF (2008) Joint multi-label multi-instance learning for image classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1–8
50. Zhang ML (2010) A k-nearest neighbor based multi-instance multi-label learning algorithm. Proceedings of the 22th IEEE International Conference on Tools with Artificial Intelligence, pp 207–212
51. Zhang Q, Goldman SA (2002) EM-DD: an improved multiple-instance learning technique. Adv Neural Inf Proces Syst 14:1073–1080
52. Zhang Q, Goldman SA, Yu W, Fritts JE (2002) Content-based image retrieval using multiple-instance learning. Proceedings of the 19th International Conference on Machine Learning, pp 682–689
53. Zhang C, Viola P (2008) Multiple-instance pruning for learning efficient cascade detectors. Adv Neural Inf Proces Syst 20:1681–1688
54. Zhang SQ, Zhao XM (2013) Dimensionality reduction-based spoken emotion recognition. Multimed Tools Appl 63(3):615–646
55. Zhang ML, Zhou ZH (2007) Multi-label learning by instance differentiation. Proceedings of the 22nd AAAI Conference on Artificial Intelligence, pp 669–674
56. Zhang ML, Zhou ZH (2007) ML-kNN: A lazy learning approach to multi-label learning. Pattern Recogn 40(7):2038–2048
57. Zhou ZH, Jiang K, Li M (2005) Multi-instance learning based web mining. Appl Intell 22(2):135–147
58. Zhou ZH, Sun YY, Li YF (2009) Multi-instance learning by treating instances as non-I.I.D. samples. Proceedings of the 26th International Conference on Machine Learning, pp 1249–1256
59. Zhou ZH, Xu JM (2007) On the relation between multi-instance learning and semi-supervised learning. Proceedings of the 24th ICML, pp 1167–1174
60. Zhou ZH, Xu JM (2007) On the relation between multi-instance learning and semi-supervised learning. Proceedings of the 24th International Conference on Machine Learning, pp 1167–1174
61. Zhou ZH, Zhang ML (2007) Multi-instance multi-label learning with application to scene classification. Adv Neural Inf Proces Syst 19:1609–1616
62. Zhou ZH, Zhang ML, Huang SJ, Li YF (2012) Multi-instance multi-label learning. Artif Intell 176(1):2291–2320

Chanjuan Liu received her Ph.D. degree in Computer Application Technology from China University of Mining & Technology (Beijing), 2013. She is currently an associate professor at School of Information and Electrical Engineering, Ludong University and a visiting scholar in Department of Computer Science and Engineering, University of South Carolina. Her current research interests include machine learning, pattern recognition and image processing.

Tongtong Chen received her Bachelor's Degree in School of Information and Electrical Engineering from Ludong University in 2013. She is currently a postgraduate student in School of Information and Electrical Engineering, Ludong University, Yantai, China. Her interests include pattern recognition and image processing.

Xinmiao Ding received her Ph.D. degree in Detection Technology and Automation Equipment from China University of Mining and Technology, Beijing, 2013. She is currently an associate professor at School of Information and Electronic Engineering, Shandong Institute of Business and Technology. Her main research interests include image and video analysis and understanding, machine learning and internet security.

Hailin Zou received the Ph.D. degree in Computer Application Technology from China University of Mining & Technology (Beijing), 2005. He is currently a full professor at School of Information and Electrical Engineering, Ludong University. His current research interests include 3D visualization, pattern recognition and image processing.

Yan Tong received her Ph.D. degree in Electrical Engineering from Rensselaer Polytechnic Institute, Troy, New York. She is currently an associate professor in the Department of Computer Science and Engineering in University of South Carolina. Before joining USC, she has been working in Visualization and Computer Vision Lab of GE Global Research Center since January 2008. Her general research interests are on computer vision, pattern recognition, and human computer interaction.