Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Concept-Dependent Image Annotation via Existence-Based Multiple-Instance Learning

Xun Yuan*, Meng Wang†, and Yan Song*
*Department of Electronic Engineering, University of Science and Technology of China, Hefei, China
{yuanxun@mail.,songy@}ustc.edu.cn
†Internet Multimedia Group, Microsoft Research Asia, Beijing, China
{mengwang}@microsoft.com

***Abstract*—Conventional multiple-instance learning (MIL) algorithms for image annotation usually neglect concept dependence (i.e., the relationship between positive and negative concepts) and feature selection (i.e., which feature modality is suitable for a specific concept) problems, which have significant influence on the annotation performance. In this paper, we propose a novel concept-dependent algorithm for image annotation, named *existence*-based MIL (EBMIL), aiming at solving the above two problems in one scheme. In our EBMIL scheme, we give a new MIL formulation, named *existence*-based MIL, to explore the concept dependence in image annotation. Moreover, we give an optimization procedure in EBMIL, which is able to select different feature modalities for each concept under MIL settings. EBMIL achieves promising experimental results on the benchmark of COREL dataset with comparison to typical MIL algorithms.**

***Index Terms*—Multiple-Instance Learning; Feature Selection; Image Annotation**

I. Introduction

In content-based image annotation, the user-queried keywords or concepts are usually highly correlated to one or more specific regions in images, while the training data are only labeled in image-level. When we use regional features extracted from the segmented images, it is difficult for standard learning algorithms to directly learn the target concepts. Consequently, multiple-instance learning was proposed to learn the regional-level concepts through training data with image-level labels [1]. In their works, each image is deemed as a labeled bag with multiple instances, and the segmented regions in that image correspond to the instances in the bag.

In traditional multiple-instance formulation, a bag is labeled positive if at least one of its instances is a positive instance, and it is labeled negative if all of its instances are negative. As shown in Figure 1, if one of the regions in the image is a "horse region", the image is positive for the concept *horse*. Recently, it is pointed out that traditional MIL formulation is not sufficient to describe some complex image concepts used by average users [2] in image annotation applications. Therefore, several other MIL formulations are proposed to try to better describe the image content by exploring concurrency (or presence) of certain set of "positive concepts" [2][3]. An intuitive example is shown in Figure 1: an image with regions of *sand*, *sea*, *people* and *sky* is most likely to be positive for concept *beach*. Following the denominations in [3], we regard these new MIL formulations as different kinds of generalized MIL.

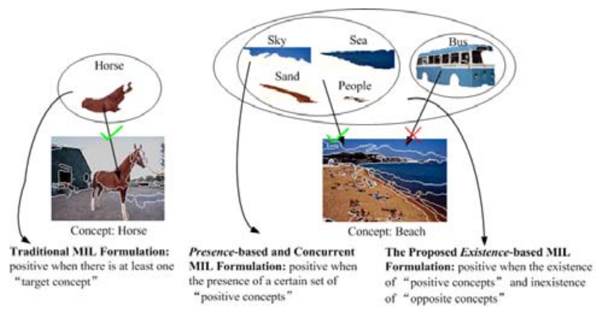

Fig. 1. Comparison of different MIL Formulations

However, only considering the concurrency of concepts is also insufficient to formulate the problems in image annotation. We argue that better performance should be acquired if we take into account the existence of "opposite concepts" additionally. The motivation is also illustrated in Figure 1: the existence of "positive concepts" such as *sand* and *sea* can significantly support the bag is positive for concept *beach*. Meanwhile the existence of "opposite concepts" such as *bus* can help degrade the confidence that one bag is positive for concept *beach*, because these concepts rarely concurrent in one image. Therefore, we formulate image annotation as another new kind of generalized MIL, named *existence*-based MIL, which explores both the "positive concepts" and "opposite concepts" simultaneously. Experimental results show that better performance is achieved by integrating "opposite concepts" into *existence*-based MIL formulation additionally.

According to the *existence*-based MIL formulation, we present a two-stage MIL algorithm named EBMIL. In the first stage, each bag is mapped into a new feature vector in bag-level feature space, thus translating the MIL problem to a standard single-instance learning problem. The mapped bag-level features are often high-dimensional feature vectors with much noise. Therefore, in the second step, an AdaBoost procedure is performed to select the bag-level features and build the final classifier. There are other feature mapping-based MIL algorithms [7][8][9][10]. The main difference between this paper and them is that we point out the concept dependencies in *existence*-based MIL formulation, and perform feature mapping according to the formulation.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

Fig. 2. Different concepts and their corresponding features

Except for the *existence*-based MIL formulation, another issue stressed in our scheme is the feature selection of raw features. It is a key problem in image annotation as there are usually large gaps between different concepts and their corresponding low-level features. For example, among the two concepts shown in Figure 2, color histogram is representative for the concept "snow mountain" because the dominant color is always white, while texture should be more effective for the concept "horse" since horses have various colors.

Generally speaking, typical feature selection algorithms under standard single-instance learning cannot be adapted to MIL settings directly, because of the coarsely labeled training data. Although various MIL algorithms have been proposed by far, the feature selection under MIL settings is rarely investigated. To the best of our knowledge, only the works in [1] and [4] perform feature selection under MIL settings. In [1], Maron et. al use a search-based method to find the best feature weighting in an algorithm called Diverse Density (DD). Zhang et. al [4] use Principal Component Analysis (PCA) to perform feature selection, which improve the performance of final classifier.

However, DD is a search-based method, which searches the best feature weighting based on traditional MIL formulation. PCA is an unsupervised dimension reduction technique which cannot utilize the label information of training data. In our EBMIL scheme, we make a simple modification to typical feature mapping-based MIL algorithm. This modification induces an optimization procedure, which is able to select raw features (instance-level features such as color, texture, etc.) implicitly by selecting the mapped bag-level features. Experimental results show that it outperforms previous feature selection algorithms under MIL settings.

In this paper, we extend our work in [5], and propose a concept-dependent image annotation scheme based on *existence*-based MIL formulation (EBMIL). Our contributions are twofold:

1) We propose *existence*-based MIL formulation to formulate the concept dependence in image annotation. Based on *existence*-based MIL formulation, we present a MIL algorithm named EBMIL. It outperforms other typical MIL algorithms based on other MIL formulations.

2) Typical feature selection algorithms under single-instance settings usually cannot be adapted under MIL settings directly. We give an optimization procedure in EBMIL scheme, which can select raw features implicitly by selecting the bag-level features. This outperforms previous feature selection algorithms under MIL settings.

The rest of this paper is organized as follows: Section II briefly reviews the related works in MIL. In Section III, we detail the basic algorithm of EBMIL, including the *existence*-based MIL formulation, the feature mapping strategy, and the classifier building procedure. In Section IV, we show how to select raw features by selecting bag-level features in EBMIL. Experimental results are provided in Section V, followed by the conclusions in Section VI.

II. Related Work

Multiple-instance learning was firstly introduced in drug activity prediction problem. Dietterich et al. formalize the MIL model and develop an algorithm named axis-parallel rectangles (APR) [6]. Maron et al. [1] firstly formulate content-based image retrieval(CBIR) as a multiple-instance learning problem. They adopt the traditional MIL formulation and develop an algorithm named Diverse Density (DD) to learn the target concepts. After these two pioneering works, extensive research is directed to develop new algorithms for MIL, as well as to explore new applications of MIL. Below we only review the works that are most relevant to this paper.

First, in terms of feature mapping-based MIL algorithms, the most relevant works are [7][8][9][10]. These algorithms attempt to solve MIL problems by mapping each bag to one feature vector in a new bag-level feature space, thus translate the MIL problems into a standard single-instance learning problem. Our algorithm is different from them mainly in the problem formulation and the corresponding feature mapping strategy. It will be detailed in Section III.

Second, considering the feature selection under multiple-instance settings, the most relevant works are [1][4]. In DD [1], Maron et al. use a search-based method to find the best weighting on the initial feature set. Zhang et al. [4] utilize PCA to eliminate the features, and improve the performance of multi-instance neural networks.

Finally, the works in [2][3] also extend traditional MIL formulation to generalized MIL formulation in order to describe more complex multi-instance problems. The differences between their formulations and ours will be detailed in Section III-A.

III. Existence-based Multiple-Instance Learning

In this section we detail the two-stage learning procedure of EMBIL. We start this section by introducing the problem formulation of *existence*-based MIL in Section III-A. In section III-B, we present the first stage, i.e. feature mapping of bags according to the problem formulation. After mapping each bag into a new bag-level feature vector, the second stage, i.e. the classifier training is illustrated in Section III-C.

A. Problem Formulation

To clearly present *existence*-based MIL formulation, we will briefly review the other three different MIL formulations for image annotation, i.e., traditional MIL [1], ConCurrent MIL [2], and *presence*-based MIL [3].

As aforementioned, in traditional MIL formulation, a bag is positive if and only if one of its instances belongs to a concept

c. Specifically, for a given concept c from the concept space, traditional MI is a function v:

$$v(B_i, c) = 1 \Leftrightarrow \Delta(B_i, c) \geq 1 \tag{1}$$

where B_i is a bag and $\triangle$ is a counting function which counts the members of a given concept in a bag.

Weidmann's *presence*-based MIL [3] is defined in terms of the presence of instances of each concept in a bag. For example, a bag is positive only if instances of concept c_1 and instances of concept c_2 are present in the bag. More generally, a *presence*-based MIL is defined as follows: for a given set of concepts C from the concept space, a *presence*-based MI is a function v_{PB}:

$$v_{PB}(B_i, C) = 1 \Leftrightarrow \forall c \in C : \Delta(B_i, c) \geq 1 \tag{2}$$

However, their method has only been performed on a toy dataset, without pointing out its real-world applications. Later, Qi et al. defined a Concurrent MIL [2] to describe the concurrency of concepts which can increase the probability that one bag is positive. Concurrent MIL essentially tells the same thing with *presence*-based MIL in definition, but it find out its real-world applications in image annotation.

However, only considering the concurrency of concepts is also insufficient to formulate the problems in image annotation. It would be more accurate if we take into account the existence of certain "opposite concepts" additionally. Therefore, we formulate image annotation as an *existence*-based MIL, which takes into account the existence of both the "positive concepts" and the "opposite concepts". More specifically, it is defined as follows: given a set of positive concepts C_1 and a set of opposite concepts C_2 from the concept space, an *existence*-based MI is a function v_{EB}:

$$\begin{aligned} &v_{EB}(B_i, C_1, C_2) = 1 \Leftrightarrow \\ &\qquad \forall c_1 \in C_1, \forall c_2 \in C_2 : \Delta(B_i, c_1) \geq 1, \Delta(B_i, c_2) = 0 \end{aligned} \tag{3}$$

The intuitive comparison of the three MIL formulations is illustrated in Figure 1. The "opposite concept" bus is the key-point in *existence*-based MIL. This formulation of MIL problem is closely related to the feature mapping strategy to be adopted. In the next subsection, we will present our feature mapping according to *existence*-based MIL formulation, and show that the feature mapping strategies in [7] and [8] are actually based on the Concurrent MIL or *presence*-based MIL formulations.

B. Feature Mapping

We solve the *existence*-based MIL by mapping each bag to a new feature vector in bag-level feature space, thus translating the MIL problem to a standard single-instance learning problem. The feature mapping is based on some points in the instance-level feature space, which potentially tell where the "positive concepts" and "opposite concepts" distribute. We call these points "instance prototypes".

According to the *existence*-based MIL formulation, there are two types of instance prototypes, i.e., the instance prototypes that represent the "positive concept" (C_1 in Eqn. (3)), named positive instance prototypes), and the instance prototypes that represent the "opposite concept" (C_2 in Eqn. (3)), named opposite instance prototypes). Obviously the positive instance prototypes should come from the positive bags, and the negative instance prototypes should come from the negative bags. We use all the instances gathered from positive bags as positive instance prototypes, denoted by $p_t(t = 1, 2, \cdots, m)$. In image annotation, the training sets are usually very imbalanced between classes, i.e., negative bags are much more than positive bags. Therefore, directly using all instances from negative bags would induce a large number of opposite instance prototypes. We use the clustering centers of instances from negative bags as opposite instance prototypes. For all instances from negative bags, we implement k-means clustering algorithm for n times with different parameters (the number of clustering centers are set to be numbers approximate the number of positive bags). Then we get totally c clustering centers which are denoted by $n_t(t = 1, 2, \cdots, c)$.

Let B_i denote the i^{th} bag, and B_{ij} denote its j^{th} instance. Denote by X and F the instance-level feature space and the mapped bag-level feature space, respectively. We define the distance between a bag B_i and an instance prototype (p_t or n_t) as the minimal distance among the distances between the instances from B_i and the instance prototype, as shown in Eqn. (4).

$$d(p_t, B_i) = \min_j d(p_t, B_{ij}) \;\; d(n_t, B_i) = \min_j d(n_t, B_{ij}), \;\;\; B_{ij} \in B_i \tag{4}$$

The distance between two instances (i.e. $d(p_t, B_i j)$ and $d(n_t, B_{ij})$) can be any distance metric. In our implementation, Euclidean distance is applied:

$$d(p_t, B_{ij}) = \exp\left(-\frac{\|p_t - B_{ij}\|^2}{\delta^2}\right) \quad d(n_t, B_{ij}) = \exp\left(-\frac{\|n_t - B_{ij}\|^2}{\delta^2}\right) \tag{5}$$

Given a bag B_i, it is mapped to a new (m+c)-dimensional feature vector in feature space F, where the first m dimensions take value $d(p_t, B_i)$, and the last c dimensions take value $d(n_t, B_i)$, as shown in Eqn. (6). The right part of Eqn. (6) is the mapped bag-level feature vector, each dimension of which corresponds to an instance prototype (p_t or n_t)

$$\begin{aligned} &B_i \xrightarrow{Feature\ Mapping} \\ &[\;\; d(p_1, B_i), \;\; \cdots, \;\; d(p_m, B_i), \;\; d(n_1, B_i), \;\; \cdots, \;\; d(n_c, B_i) \;\;] \end{aligned} \tag{6}$$

An intuitive way to understand this feature mapping process is illustrated in Figure 3: if an instance prototype p_t approximates the "positive concept" in feature space X, the value $d(p_t, B_i)$ should be small for positive bags while large for negative bags, thus it is useful to distinguish positive and negative bags. Analogously, if an instance prototype n_t approximate the "negative concept" in feature space X, the value $d(n_t, B_i)$ should be large for positive bags while small for negative bags.

However, according to MIL settings, only a small part of the instances in positive bags are the truly positive instances that

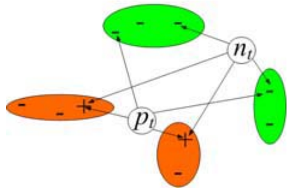

Fig. 3. An intuitive illustration of the Feature Mapping. The radials from p_t and n_t stand for $d(p_t, B_i)$ and $d(n_t, B_i)$. Intuitively $d(p_t, B_i)$ tends to be small for positive bags and large for negative bags, while $d(n_t, B_i)$ is on the contrary

can approximate the "positive concept". Analogously, only a small part of the negative instance prototypes can tell where the "negative concept" is. Therefore, we need a feature selection process to select the useful feature subset from the mapped $(m+c)$-dimensional bag-level feature vector. In our work this is done by AdaBoost with a type of linear weak classifier.

Our feature mapping strategy differs from [7][8][9][10] mainly in that we consider the "opposite concept", which is additionally formulated in *existence*-based MIL, whereas the instance prototypes in [7][8] are all from positive bags. Therefore, they are based on *presence*-based MIL. In [10], the instance prototypes are chosen from both the training bags and test bags, without differentiating "positive concepts" and "negative concepts". We should point out that in [9], the authors use all the instances in training bags to perform feature mapping, but the instances in negative bags are also deemed to reflect the "target concept" in their explanation.

C. AdaBoost Feature Selection and Classifier Building

After feature mapping, each bag is transformed to a new feature vector in bag-level feature space, and the MIL problem is converted to a standard single-instance problem. As aforementioned, the mapped bag-level feature vector is a high-dimensional feature vector with much noise. We use AdaBoost with a type of linear weak classifier to perform feature selection and build the final classifier. Next we will detail the adaboost feature selection procedure, and the learning weak classifier used in it.

1) Learning Weak Classifier for AdaBoost: There are two things need to be considered for selecting the weak classifier. First, according to the intuitive explanation of the feature mapping in Section III-B, if a certain dimension of the mapped bag features is relevant, this single dimension has certain ability to distinguish positive and negative bags. That is, actually each weak classifier corresponds to one single feature dimension. Second, in order to make the whole training procedure efficient, the weak classifier should be cheap to compute. Based on the above two ponts, we construct an efficient linear classifier on each single dimension of the bag-level feature vector. The weak classifier works as Table I shows, which can output confidence-rated real value predictions [13].

Next we will introduce how to integrate this weak classifier into the AdaBoost feature selection procedure.

2) AdaBoost Feature Selection Procedure: AdaBoost can be used to combine feature selection and classifier training into

TABLE I

THE LEARNING WEAK CLASSIFIER

Input:
$(x_1, y_1), \cdots, (x_m, y_m)$, where $y_i = 0, 1$ for negative and positive examples respectively, training sample weight $w(x_i)$

Algorithm:
1) Divide the feature space into n sub spaces $X_1, X_2, \cdots, X_n$
2) Give the training sample weight $w(x_i)$, calculate $W_l^i, l = 0, 1, j = 1, 2, \cdots, n$
$W_l^i = P(x_i \in X_j, y_i = l) = \sum_{i: x_i \in X_j, y_i = l} w(x_i)$
3) The output of the weak learner on the training sample weight $w(x_i)$ is
$h(x) = \frac{1}{2} \log(\frac{W_1^j + \varepsilon}{W_0^j + \varepsilon}), \quad j = 1, ..., n, \quad \forall x \in X_j$
where ϵ is a small positive constant
4) The corresponding training error is calculated as follows:
$err = \sum_{h(x_i) \geq 0, y_i = 0} w(x_i) + \sum_{h(x_i) < 0, y_i = 1} w(x_i)$

TABLE II

THE ADABOOST FEATURE SELECTION AND CLASSIFIER BUILDING ALGORITHM

Input:
$(x_1, y_1), ..., (x_m, y_m)$ where $y_i = -1, 1$ for negative and positive examples

1) Initialize weight $w(x_i) = 1/2n, 1/2p$ for $y_i = 0, 1$ respectively, where n and p are the number of negative and positive samples.
2) For $t = 1, ..., T$
a) According to the weak classifier shown in Table I, train one weak classifier $h_j \in R$ for each feature j using $w(x_i)$, and get the corresponding training error err_j.
b) Choose $h_t = h_j$ with the lowest error, set $err_t = err_j$.
c) Update: $w_{t+1}(x_i) = \frac{w_t(x_i)}{Z_{tp}} \times e^{-y_i h_t}$ for positive samples,
and $w_{t+1}(x_i) = \frac{w_t(x_i)}{Z_{tn}} \times e^{-y_i h_t}$ for negative samples,
where Z_{tp} and Z_{tn} are normalization factors to ensure the weight of positive and negative samples all sum up to $1/2$.

Output:
The final hypothesis is: $H(x) = sign(\sum_{t=1}^{T} h_t(x))$

one procedure. The key idea behind AdaBoost is that a strong classifier can be created by combining many weak classifiers having training errors just above 50%. At each iteration step, one weak classifier is selected, and the training samples are re-weighted to put more emphasis on the misclassified samples. The final strong classifier is a combination of the weak classifiers. Specifically, our AdaBoost feature selection algorithm is shown in Table II.

Note that our AdaBoost algorithm is a little different from the original AdaBoost [12] in the sample weight initialization and updating steps. In our scheme, the sums of sample weights for positive and negative samples are always kept equal to be $1/2$. This is to tackle the unbalance problem in training data when the number of negative training bags is much larger than the number of positive training bags. As the weak classifier we used has folded the its weight into its real value prediction, the final strong classifier is a direct combination of the weak classifiers instead of weighted combination.

In our AdaBoost procedure, one dimension is selected at each iteration step, so the number of iterations (i.e., T) performed is related to the number of dimensions that have enough differentiation abilities in the whole feature vector. In our experiments, it is set to be a little higher than the number

TABLE III
THE LOW-LEVEL FEATURES IN OUR FEATURE POOL

Feature Type	Low-level Feature Sources	Dimension
Color	Color HSV Histogram	64
	Color RGB Histogram	64
	Color Luv Histogram	64
	Color Moment	9
	Color Correlogram	144
Texture	Wavelet PWT Texture	24
	Wavelet TWT Texture	104
Edge	Edge Distribution Histogram	15
Shape	Normalized Inertia	3
Others	Region Size	1
	Region Position	2
In Total		494

TABLE IV
COMPARISON OF TYPICAL PROCEDURE OF FEATURE MAPPING-BASED MIL AND EBMIL

(a) Typical Procedure of Feature Mapping-based MIL:
1. Concatenate each feature source X_k and get instance-level feature vector X
2. F = FeatureMapping(X)
3. Perform feature selection or train a classifier directly on F (note that it cannot select different feature source X_k)

(b) The Procedure of EBMIL to Select Raw Features:
1. For each feature source X_k
   F_k = FeatureMapping(X_k)
2. Concatenate F_k and get a new feature vector F
3. Perform feature selection using adaboost on F, this procedure select different feature source X_k implicitly

of positive bags in the training data.

IV. SELECT RAW FEATURES BY SELECTING BAG-LEVEL FEATURES

As aforementioned, there are large variations among the most effective features for different concepts. Therefore, incorporating feature selection for different concepts is expected to improve performance. We extract various feature sources so that they can cover the effective features for different concepts, as shown in Table III.

Because of the coarsely labeled training data, typical feature selection algorithm under singles-instance settings cannot be adapted directly under MIL settings. To the best of our knowledge, only the works in [1] and [4] investigate the feature selection problems under MIL settings. In [1], Maron et. al employ a search-based method to find best weight for each feature dimension. However, the search process is conducted based on traditional MIL formulation. In [4], Zhang et. al use PCA to perform dimension reduction. PCA is an unsupervised feature reduction algorithm which cannot utilize the label information of training data. Another possible way to avoid dimensionality curse, is to train classifiers on each feature set independently, and then ensemble these classifiers by late fusion [17]. However, the fusion of these classifiers doesn't equal to feature selection; it doesn't select the effective features for each concept. In this paper, we present an optimization procedure in EBMIL, which can select different raw features implicitly during the selection of bag-level features.

The typical working procedure of previous feature mapping-based MIL algorithms cannot provide raw feature selection function for image annotation. Therefore, we utilize a modified procedure to select different raw feature sources from Table III. Let X_k denote the k-th raw feature source, and F_k denote its corresponding mapped bag-level feature. The comparison of typical procedure of feature mapping-based MIL and EBMIL is shown in Table IV.

The main difference between the typical procedure and EBMIL lies in that: the typical procedure concatenates each feature source X_k together and perform feature mapping (this actually corresponds to early fusion of features [17]), while EBMIL perform feature mapping for each feature source X_k and get F_k respectively, and then perform feature selection on the concatenated bag-level feature F. Concatenating all the raw features in Table III will induce a high-dimensional instance-level feature vector, and using it directly for feature mapping may leads to a "curse of dimensionality" problem. Therefore, the typical procedure cannot provide raw feature selection function for image annotation.

However, the procedure of EBMIL is able to select different raw feature sources from Table III implicitly, during the selection of bag-level features. According to the previous intuitive explanation in Section III-B, each dimension of F_k is a certain potentially discriminative feature from feature source X_k. Since F is obtained by concatenating F_k, feature selection on F is a joint feature selection on every F_k, and thus it is a joint feature selection from every feature source X_k as well.

V. EXPERIMENTS

We evaluate EBMIL on the most widely used COREL 2000 benchmark dataset. It has 20 semantically diverse categories, each of which contains 100 images. The images are segmented using JSEG segmentation [14], and different feature sources in Table III are extracted from the segmented regions. We perform 10 trials and in each trial the dataset is randomly split into a training set and a test set with equal sizes. The mean average precision (MAP) [15] measure is adopted as the performance evaluation criterion.

Two sets of experiments are implemented to evaluate the proposed algorithm. In Section V-A, we evaluate the *existence*-based MIL formulation by comparing EBMIL with other typical MIL algorithms, which adopt other MIL formulations. In Section V-B, we demonstrate the feature selection ability of EBMIL by comparing it with other feature selection methods under MIL settings.

A. Existence-based MIL Formulation

We compare our algorithm with three typical existing MIL algorithms: Diverse Density (DD) [1], 1-norm SVM [8], and MI-Boosting [16]. The reason why we choose these three algorithms is that: DD is a typical MIL algorithm adopting traditional MIL formulation, 1-norm SVM represent the Concurrent or *presence*-based MIL formulation (see section III-B), and MI-Boosting is another MIL algorithm utilizing boosting

TABLE V
MAP of The Four Algorithms

Algorithm	MAP
Diverse Density (DD) [1]	0.367
1-norm SVM [8]	0.567
MI-Boosting [16]	0.616
EBMIL	0.699

TABLE VI
Raw Feature Selection under MIL Settings

Scheme		**MAP**
Proposed MIL scheme using typical Feature Sets	Color Correlogram	0.699
	Color Moment	0.351
	Color HSV Histogram	0.573
	Edge Histogram	0.269
Proposed MIL scheme using All Features with PCA [4]		0.521
Proposed MIL scheme using All Features by Late Fusion [17]		0.653
Proposed MIL scheme using All Features Directly (Table IV(a))		0.324
EBMIL (Table IV(b))		0.719

techniques. Table V illustrates the experimental results obtained by the four methods. Since MI-Boosting and 1-norm SVM don't provide feature selection function for raw features, and our aim is only to compare different MIL formulations in this subsection, results on separate feature source instead of all feature sources are reported. There are many different feature sources, only the results on color correlogram are listed due to the limited space; the results are the same on other feature sources as well. The numbers of clustering centers for negative instances are set to 40, 50, and 60 according to the rule in Section III-B. The parameter δ^2 in Eqn. (5) is set by cross-validation in $\{20, 25, 30, 35, 40\}$.

From Table V we can see that our method obtains the best result among the four methods. This demonstrates that the *existence*-based MIL formulation adopted by EBMIL can better describe the image content.

B. Raw Feature Selection under MIL Settings

We compare the performance of the following schemes: (1) the proposed MIL scheme with four typical feature sets; (2) the proposed MIL scheme using all features with PCA for dimension reduction [4]; (3) the proposed MIL scheme using all features by late fusion [17]; (4) the proposed MIL using all features directly (Table IV(a)); and (5) EBMIL(i.e., using feature selection (Table IV(b)). Among the above schemes, (1) is the proposed scheme using separate feature source; (2) is a previous feature selection method utilized under MIL settings; (3) is a fusion method to avoid dimensionality curse under MIL settings; (4) is the proposed MIL scheme with typical procedure of feature mapping-based MIL algorithms; (5) is the proposed EBMIL scheme with feature selection of raw features. Table VI illustrates the results.

From Table VI we can see that EBMIL obtains the best result. This confirms the effectiveness of the proposed feature selection strategy. Without feature selection, using all feature sources significantly degrades performance due to the curse of dimensionality.

VI. Conclusions

This paper presents a concept-dependent image annotation algorithm, called *existence*-based multiple-instance learning (EBMIL). To better describe the image content, we formulate image annotation as *existence*-based generalized MIL, and give a two-stage learning method according to the problem formulation. Then we give a raw feature selection procedure of EBMIL and show how it selects different raw features by selecting bag-level features. To the best of our knowledge, it is the first work which investigates concept-dependence relationship and feature selection under MIL settings. Experimental results on benchmark COREL dataset demonstrate the effectiveness of EBMIL.

Acknowledgment

This work is supported in part by Anhui Provincial Natural Science Foundation under grant 090412056, and Microsoft Internet Service Theme 2009 under grant FY09-RES-THEME-075.

References

[1] O. Maron and A. Ratan. "Multiple-instance learning for natural scene classification." *Proceedings of International Conference on Machine Learning(ICML)*, 1998.

[2] G. -J. Qi, X. -S. Hua, Y. Rui, T. Mei, J. -H. Tang, H. -J. Zhang. "Concurrent multiple-instance learning for image categorization." *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition(CVPR)*, 2007.

[3] N. Weidmann, E. Frank, and B. Pfahringer. "A two-level learning method for generalized multi-instance problem." *Lecture Notes in Artificial Intelligence (LNAI)*, 2003.

[4] M. -L. Zhang, and Z. -H. Zhou. "Improve multi-instance neural networks through feature selection." *Neural Processing Letters*, 2004.

[5] X. Yuan, X. -Sheng Hua, M. Wang, G. -J. Qi, Xiuqing Wu. "A Novel Multiple Instance Learning Approach For Image Retrieval Based on Adaboost Feature Selection." *Proceedings of IEEE International Conference on Multimedia & Expo(ICME)*, 1998.

[6] T. Dietterich, R. Lathrop, and T. Lozano-perez. "Solving the multiple instance problem with the axis-parallel rectangles." *Artificial Intelligence*, 1997.

[7] Y. Chen and J. Wang. "Image categorization by learning and reasoning with regions. " *Journal of Machine Learning Research (JMLR)*, 2004.

[8] J. Bi, Y. Chen and J. Wang. "A sparse support vector machine approach to region-based image categorization." *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition(CVPR)*, 2005.

[9] Yixin Chen, Jinbo Bi and James Z. Wang. "MILES: Multiple-Instance Learning via Embedded Instance Selection." *IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI)*,2006.

[10] Z. -H. Zhou and M. -L. Zhang. "Solving multi-instance problems with classifier ensemble based on constructive clustering." *Knowledge and Information Systems(KAIS)*, 2007.

[11] Joliffe I.T. "Principal Component Analysis." *Springer-Verlag, New York*, 1986.

[12] J. Friedman, T. Hastie, and R. Tibshirani. "Additive logistic regression: A statistical view of boosting." *Annals of Statistics*, 2000.

[13] R.E. Schapire and Y. Singer. "Improved boosting algorithms using confidence-rated predictions." *Proceedings of the eleventh annual conference on computational learning theory(COLT)*, 1998.

[14] Y. Deng and B. S. Manjunath. "Unsupervised segmentation of color-texture regions in images and video." *IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI)*, 2001.

[15] M. R. Naphade, and J. R. Smith. "On the Detection of Semantic Concepts at TRECVID." *Proceedings of ACM Multimedia(ACM MM)*, 2006.

[16] X. Xu, and E. Frank. "Logistic regression and boosting for labeled bags of instances." *Lecture Notes in Artificial Intelligence(LNAI)*, 2004.

[17] C.G. Snoek, M. Worring, A. Smeulders. "Early versus Late Fusion in Semantic Video Analysis." *Proceedings of ACM Multimedia(ACM MM)*, 2005.