Multi-Label Sparse Coding for Automatic Image Annotation

Changhu Wang[1]*, Shuicheng Yan[2], Lei Zhang[3], Hong-Jiang Zhang[4]
[1]MOE-MS Key Lab of MCC, University of Science and Technology of China
[2]Department of Electrical and Computer Engineering, National University of Singapore
[3]Microsoft Research Asia, [4]Microsoft Advanced Technology Center, Beijing, China
wch@ustc.edu, eleyans@nus.edu.sg, {leizhang,hjzhang}@microsoft.com

Abstract

In this paper, we present a multi-label sparse coding framework for feature extraction and classification within the context of automatic image annotation. First, each image is encoded into a so-called supervector, derived from the universal Gaussian Mixture Models on orderless image patches. Then, a label sparse coding based subspace learning algorithm is derived to effectively harness multi-label information for dimensionality reduction. Finally, the sparse coding method for multi-label data is proposed to propagate the multi-labels of the training images to the query image with the sparse ℓ^1 reconstruction coefficients. Extensive image annotation experiments on the Corel5k and Corel30k databases both show the superior performance of the proposed multi-label sparse coding framework over the state-of-the-art algorithms.

1. Introduction

Automatic image annotation, whose goal is to automatically assign the images with the keywords, has been an active research topic owing to its great potentials in image retrieval and management systems. Image annotation is essentially a typical multi-label learning problem, where each image could contain multiple objects and therefore could be associated with a set of labels. Since generally it is tedious and time-consuming for humans to manually annotate the keywords in the object/region level for data collection, instead the keywords are usually labeled in the image level, which makes the automatic image annotation problem even more challenging.

The image annotation problem has been extensively studied in recent years. The popular algorithms can be roughly divided into three categories: classification-based methods, probabilistic modeling-based methods, and Web image related methods. The classification-based methods [3][5][6][16] use image classifiers to represent annotation keywords (concepts). The probabilistic modeling-based methods [2][9][10][11][14][17] attempt to infer the correlations or joint probabilities between images and annotation keywords. Web image related methods [20][21][22][23] try to solve image annotation problem in Web environment. There are also some attempts to use multi-label learning algorithms to solve image annotation problem, which are scattered in different categories mentioned above. Most of existing attempts of using multi-label learning algorithms [13][26] to solve image annotation problem mainly focus on mining the label relationship for better annotation performance.

In spite of these many algorithms proposed with different motivations, the underlying question, *i.e.* how to effectively measure the semantic similarity between two images with multiple objects/semantics, is still not well answered. There are mainly three kinds of features for image representation, *i.e.* global features [20][23], region-based features [9][11][14], and patch-based features (or local descriptors) [3][10], out of which the region-based features seem the most reasonable for the above-mentioned multi-label essence of images. However, in practice, on the one hand, it is too time-consuming to manually segment images into regions; on the other hand, without human interaction, the automatic image segmentation algorithms are far from satisfaction. Thus, the existing works based on region-based features [9][11][14] are often inferior to the those patch-based algorithms [3][10].

An inevitable and practical choice for image annotation is then to use global features or patch-based features instead of region-based features. In most existing algorithms, the global features or patch-based features are directly compared to determine the image-to-image similarity. However, there are usually multiple semantic concepts in one image, and two images containing one same object may have additional different objects too. For example, as shown in Fig.

[*] Changhu Wang performed this work while being a Research Engineer at the Department of Electrical and Computer Engineering, National University of Singapore.

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

1, an image with objects “tiger”, “ground”, and “bush” may visually different a lot from the images with only one or two objects in the whole image view. Due to the well-known semantic gap, if there is not a visually similar image with exactly the three objects in the database, it may be difficult to retrieve an image only containing a subset of the three objects through existing image-to-image similarity measure. Therefore, two natural questions to ask are: 1) how we can measure the semantic similarity between two label sets of two images for effective feature extraction, and 2) how we can measure the semantic similarity of a training image to the query image, both with multi-labels.

This work is dedicated to answering the above two questions within the context of automatic image annotation. We claim that the semantic similarity of two images with overlapped labels should be measured in a reconstruction-based way rather than in a one-to-one way, based on which a multi-label sparse coding framework is presented for feature extraction and classification. Beyond the one-to-one similarity, the semantic similarities of label vectors and image features are both measured based on *one-to-all* ℓ^1 sparse reconstruction/coding as introduced afterwards. First, each image is encoded into a so-called supervector, derived from the universal Gaussian Mixture Models on image patches. Second, a label sparse coding based subspace learning algorithm is derived to effectively harness multi-label information for feature extraction. Finally, the sparse coding method for multi-label data is proposed to propagate the multi-labels of the training images to the query image with the sparse ℓ^1 reconstruction coefficients.

An example to show the core idea of this work is illustrated in Fig. 1, where a query image with objects “tiger”, “ground”, and “bush” could be linearly reconstructed by three images with one or two related objects. We can see that all the three “component” images are only partially related with the query image. If we use the direct one-to-one similarity, the “noise” image would be even more similar than the “component” images to the query image, but this “noise” image is removed if the related images are obtained in a *one-to-all* sparse reconstruction way.

2. Multi-Label Sparse Coding Framework

In this section, we introduce the multi-label sparse coding framework for automatic image annotation. The entire framework includes three components: 1) feature representation based on probabilistic patch modeling; 2) label sparse coding for effectively harnessing multi-label information in feature extraction; and 3) data sparse coding for multi-label data to propagate the multi-labels of the training images to the query image with the sparse ℓ^1 reconstruction coefficients.

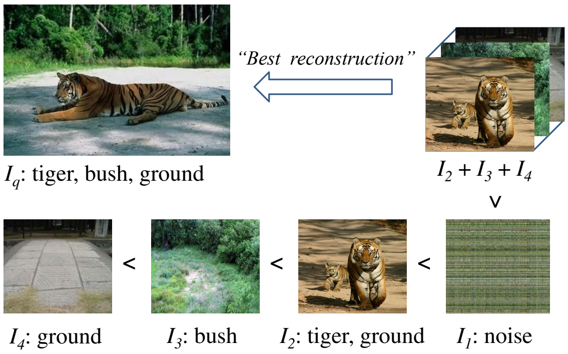

Figure 1. An illustrative example of the *one-to-all* sparse reconstruction/coding for semantic similarity measure. I_q is the query image to be annotated. I_1 - I_4 are four training images out of which I_2 - I_4 are semantically related with I_q. I_1 represents a semantically unrelated image of I_q, which however has similar color histogram features with I_q. Although I_1 is more similar to I_q than the other three ones using color histogram features and based on one-to-one similarities, the linear combination of I_2 - I_4 could result in a more ideal reconstruction of I_q than the noisy image I_1. We therefore use the sparse reconstruction/coding coefficients as the corresponding semantic similarities to the query image.

2.1. Feature Representation

Assume that there are N images in the training set, denoted as $X = [x_1, x_2, \cdots, x_N]$, where each image x_i is encoded as an ensemble of patches $\{x_i^j, x_i^j \in \mathbb{R}^m\}$. Here m is the extracted feature dimension for each patch. First, a global Gaussian Mixture Models (GMM) is estimated based on all patches from the training images, and then each image is encoded as an image-specific GMM, which is adapted from the global GMM, finally, a length-fixed supervector is used to represent an image guided by the Kullback-Leibler (KL) divergence between any two image-specific GMMs.

2.1.1 Universal background GMM model

We first estimate a global GMM, which characterizes the general patch distribution, based on all patches from the training images regardless of their labels. It is similar to the so-called Universal Background Model (UBM) in speech/speaker verification [18]. For ease of presentation, we denote $\mathbf{z}$ as the feature vector of a patch. The distribution of the variable $\mathbf{z}$ is assumed to be

$$p(\mathbf{z}; \Theta) = \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{z}; \mu_k, \Sigma_k), \qquad (1)$$

where $\Theta = \{w_1, \mu_1, \Sigma_1, \cdots\}$, w_k, μ_k and Σ_k are the weight, mean and covariance matrix of the kth Gaussian component, respectively, and K is the total number of Gaussian components. We can obtain a maximum likelihood parameter set for the GMM by using the Expectation-Maximization (EM) algorithm [7] as conventionally.

2.1.2 Image specific GMM by EM adaptation

Based on the patches extracted for each image, an image-specific GMM can be obtained by adapting the mean vectors of the global GMM and retaining the mixture weights and covariance matrices. Mean vectors are adapted using MAP adaptation [15] with conjugate priors [15], and then the image-specific parameters $\hat{\mu}_k$ could be obtained by EM method. Assuming that the $Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_H\}$ are the patches extracted from the image being modeled, then in the E-step, we compute the posterior probability of Gaussian component k given patch $\mathbf{z}_i$ [15],

$$Pr(k|\mathbf{z}_i) = \frac{w_k \mathcal{N}(\mathbf{z}_i; \mu_k, \Sigma_k)}{\sum_{j=1}^{H} w_j \mathcal{N}(\mathbf{z}_i; \mu_j, \Sigma_j)}, \quad (2)$$

$$n_k = \sum_{i=1}^{H} Pr(k|\mathbf{z}_i), \quad (3)$$

and then the M-step updates the mean vectors, namely

$$\overline{\mu}_k = \frac{1}{n_k} \sum_{i=1}^{H} Pr(k|\mathbf{z}_i)\mathbf{z}_i, \quad (4)$$

$$\hat{\mu}_k = \alpha_k \overline{\mu}_k + (1 - \alpha_k)\mu_k, \quad (5)$$

where $\alpha_k = n_k/(n_k + r)$. The parameter r is set to be 4 empirically in this work.

2.1.3 Supervector representation

The KL-divergence is generally used for measuring the similarity between two distributions, and the work in [25] shows that KL-divergence of two adapted GMMs can be approximated with the Euclidean distance between the so-called supervectors as,

$$x_i = [\sqrt{\frac{w_1}{2}} \Sigma_1^{-\frac{1}{2}} \hat{\mu}_1^i; \cdots; \sqrt{\frac{w_K}{2}} \Sigma_K^{-\frac{1}{2}} \hat{\mu}_K^i], \quad (6)$$

where $\hat{\mu}_j^i, j = 1, ..., K$, are the adapted mean vectors for the image x_i, and we use x_i to represent both original image and the corresponding supervector in this work for simplicity. Then, each image is finally represented as a supervector for consequent process.

2.1.4 Implementation details

In this work, the images are first resized to three resolutions, *i.e.* 128×192, 64×96, and 32×48 pixels, and the YBR color space is used. The overlapping 8×8 patches of all resolutions, extracted with a sliding window that moves by four pixels between consecutive patches, compose the bag of patches to represent an image. Each color channel of a patch is represented by a 31-dimensional feature vector, which is selected from a 64-dimensional discrete cosine transformation (DCT) coefficients (the first half except the first one). The features of all three channels are concatenated to compose a 93-dimensional feature vector, followed by Principal Component Analysis (PCA) [12] dimensionality reduction to be a 40-dimensional feature vector. 1024 Gaussian components are used in the global GMM. The dimension of the supervector is represented as m_1, which is usually very large. In this implementation it is $1024 \times 40 = 40960$ dimension. For computation efficiency, we reduce m_1 to be 2000 (for Corel5k) or 1000 (for Corel30k) using PCA.

2.2. Label Sparse Coding for Feature Extraction

2.2.1 Motivations

It is obvious that the above supervector representation does not explicitly utilize label information of images. There is much evidence in the literature of dimensionality reduction [1][12][19] that it would be useful to reduce high-dimensional feature space to a lower-dimensional semantic space oriented by label information. In this subsection, by targeting at the multi-label problem in image annotation, we propose a label sparse coding based subspace learning algorithm to effectively harness multi-label information for feature extraction, which is referred to as multi-label linear embedding (MLE) afterwards.

Assume that the multi-labels of the training images X are represented as an $(N_c \times N)$ matrix C, where the i-th column c_i is the label vector of image x_i, and the j-th element c_i is set to be 1 if x_i is labeled by the j-th label in the vocabulary, 0 otherwise. Since an image could be labeled by multiple keywords, there may exist more than one non-zero elements in a label vector c_i.

The purpose of MLE is to learn a linear transformation matrix $P \in \mathbb{R}^{m_1 \times m_2}, (m_2 < m_1)$ to transform data by $y_i = P^T x_i$ from the original feature space into a lower-dimensional one, in which the semantic relations can be retained. For conventional supervised subspace learning algorithms, e.g. Linear Discriminant Analysis (LDA) [1], an underlying assumption is that data points with the same label should be close to each other, and data points with different labels tend to be faraway. However, in the multi-label setting, this assumption is not valid as shown in Fig. 1, where samples with less similar label sets are even more similar in feature space. Thus, one-to-one similarity is not good for guiding the dimensionality reduction in the multi-label setting.

In this work, we adopt two ways to use multi-label information for guiding feature extraction, which results in two weight matrices. First, the images with exactly the same label set, namely $c_i = c_j$, are considered to be fully semantically related, and then we set $W_{ij}^1 = W_{ji}^1 = 1$; 0, otherwise. Second, as the number of image pairs with exactly the same label set is often small for real-world image set, we propose to use label sparse coding to reveal more semantic relatedness, namely, each label vector c_i is reconstructed with the rest label vectors by ℓ^1-minimization, and then this sparse reconstruction relation is expected also valid for the desired low-dimensional feature space.

Discussion: why not use direct one-to-one label vector similarity for measuring semantic relatedness? It is not valid to directly use the similarity of two label vectors to measure the semantic similarity of the two images, since there might exist visually incompatible objects in the two images yet with one common object, whose features are improper to be forced to be close to each other in the low-dimensional feature space. For example, an image with labels "fish" and "plate", which shows the food on the table, may be visually different a lot from an image with labels "fish" and "coral". Moreover, directly calculating the similarity between two label vectors could not distinguish a polysemous word with different meanings in different images. For example, an image with labels "tiger" (animal) and "forest" is semantically different from an image with labels "tiger" and "cellaret" (a kind of wine). In this work, we use the label vectors of other images in the training set to sparsely reconstruct the label vector of each image, and the reconstruction coefficients could be considered to reveal the semantic relationship between images. This method could potentially avoid the mistakes above-mentioned, since label pairs "plate" and "coral", or pairs "forest" and "cellaret", will scarcely belong to the same image, and it will be impossible to use one vector containing "plate" (or "forest") to reconstruct "coral" (or "cellaret"), and vice versa.

2.2.2 Semantic graph construction by ℓ^1-minimization

Sparse representation is widely used in statistical signal processing community, whose original goal is to represent and compress signals. It is computed with respect to an overcomplete dictionary of base elements or signal atoms [8]. Although the sparsest representation problem is NP-hard in general case, recent results [8] have shown that if the solution is sparse enough, the sparse representation can be recovered by a convex ℓ^1-minimization.

Suppose we have an underdetermined system of linear equations: $x = D\alpha$, where $x \in \mathbb{R}^m$ is the vector to be approximated, $\alpha \in \mathbb{R}^n$ is the vector for unknown reconstruction coefficients, and $D \in \mathbb{R}^{m\times n}(m < n)$ is the overcomplete dictionary with n bases. If the solution for x is sparse enough, it can be recovered by the following convex optimization,

$$\min_{\alpha} \ \|\alpha\|_1, \ \ s.t. \ x = D\alpha. \tag{7}$$

In practice, due to the noise, the exact $x = D\alpha$ may not be satisfied since m may be larger than n. To solve this problem, Wright et al. [24] proposed to reconstruct x by

$$x = D\alpha + \zeta, \tag{8}$$

where ζ is the noise term. Then the sparse representation problem is redefined by minimizing the ℓ^1-norm of both coefficients and reconstruction error, and turned to solve

$$\min_{\alpha'} \ \|\alpha'\|_1, \ \ s.t. \ x = B\alpha', \tag{9}$$

where $B = [D, I] \in \mathbb{R}^{m\times(n+m)}$ and $\alpha' = [\alpha^T, \zeta^T]^T$. This problem could be transformed into a general linear programming problem, where exists a globally optimal solution. In this work, we convert the original constrained optimization problem into an unconstrained one, with an extra regularization coefficient,

$$\min_{\alpha'} \ \lambda\|\alpha'\|_1 + \frac{1}{2}\|x - B\alpha'\|_2^2 \tag{10}$$

Based on the sparse representation described as above, the ℓ^1-oriented semantic graph is given as follows:

1. **Input:** The label matrix of the training data, namely $C = [c_1, c_2, \cdots, c_N], c_i \in \mathbb{R}^{N_c}$.
2. **Sparse Representation:** Each label vector c_i in C is normalized to be $c_i/\|c_i\|$.
   for $i = 1 : N$
   (a) Set $C\backslash c_i = [c_1, c_2, ..., c_{i-1}, c_{i+1}, ..., c_N]$.
   (b) The sparse representation for each label vector c_i is obtained by solving the optimization problems:
   $$\min_{\alpha} \ \lambda\|\alpha\|_1 + \frac{1}{2}\|c_i - B\alpha\|_2^2,$$
   where $B = [C\backslash c_i, I] \in \mathbb{R}^{N_c\times(N-1+N_c)}$ and $\alpha \in \mathbb{R}^{N-1+N_c}$.
   (c) For $1 \le j \le i-1$, we set $W_{ij}^2 = \alpha_j$; for $i+1 \le j \le N$, we set $W_{ij}^2 = \alpha_{j-1}$.
   end
3. **Output:** The semantic graph W^2 with all diagonal elements being zero.

2.2.3 Multi-label linear embedding (MLE)

After the construction of two semantic graphs W^1 and W^2, the transformation matrix P can be derived for two objectives. On the one hand, the images with exactly the same label set should be similar in the low dimensional feature space, which results in the following optimization,

$$\min_P \frac{1}{2}\sum_{ij} \|P^T x_i - P^T x_j\|^2 W_{ij}^1, \ \ s.t. \ P^T P = I. \tag{11}$$

On the other hand, the matrix W^2 characterizes the semantic relations between each image and the rest ones, and these reconstruction relations should also be valid in the low dimensional feature space, which results in

$$\min_P \frac{1}{2}\sum_i \|P^T x_i - \sum_j W_{ij}^2 P^T x_j\|^2, \ \ s.t. \ P^T P = I. \tag{12}$$

By combining these two objectives, the transformation matrix P can be derived by optimizing

$$\min_{P} Tr(P^T XMX^T P), \quad s.t. \; P^T P = I, \tag{13}$$

where the matrix M is defined as

$$M = D^1 - W^1 + \frac{\beta}{2}(I - W^2)^T (I - W^2), \tag{14}$$

and D^1 is a diagonal matrix with $D^1_{ii} = \sum_{j \neq i} W^1_{ij}, \; \forall \; i$. β is a positive parameter for balancing the aforementioned two objectives, which is set to be 0.1 in this experiment. The solution for Eqn. (13) can be obtained with the eigenvalue decomposition method,

$$XMX^T p_k = \lambda_k p_k, \tag{15}$$

where p_k is the eigenvector corresponding to the k-th smallest eigenvalue λ_k of XMX^T, and also the k-th column vector of the matrix P. Then based on P, an m_1-dimensional feature vector of a training or testing image x is reduced to an m_2-dimensional feature vector y via $y = P^T x$.

2.3. Sparse Coding for Multi-Label Annotation

Based on the MLE subspace learning algorithm, all the training images are mapped into the lower-dimensional feature space, denoted as the matrix $Y = [y_1, y_2, \cdots, y_N]$. Similarly, for a query image, it can also be mapped into this feature space, denoted as y^t. The task of multi-label image annotation is to assign a set of labels to this image based on the label information of the training images denoted as C.

2.3.1 Motivations

Similar to the semantic similarity between label vectors as aforementioned, it is crucial to calculate the semantic similarity between two visual features y_i and y_j. In this work, we claim that two images with overlapped labels may not be close to each other in the derived m_2-dimensional feature space, since although they may contain quite similar parts, there may also exist quite different parts from other objects in the images. For example, let y_i be the feature vector of an image with labels “tiger” and “ground”, and y_j is assigned with labels “tiger” and “bush”. It is obvious that, on the one hand, y_i and y_j have common component to represent the semantic concept of “tiger”, but on the other hand, they also contain totally different components to represent “ground” and “bush” respectively. Since we do not have the true segmentation information of the two images, it is difficult to retrieve y_j queried by y_i using traditional one-to-one similarity based on the derived feature space. This observation motivates us to propose a sparse coding method, similar to label sparse coding aforementioned, to build semantic relations based on *one-to-all* reconstruction.

2.3.2 ℓ^1-reconstruction for multi-label image

To avoid the limitation of the one-to-one similarity measure in multi-label context, we use training images Y as the bases to sparsely reconstruct the query image y^t with the ℓ^1 constraint. The training images with non-zero reconstruction coefficients are considered *semantically* related to the query image. This method does not force the retrieved semantically related images be globally similar to the query image, and therefore could retrieve images with partially overlapped objects with the query image. On the one hand, in order to perfectly reconstruct the query image, the retrieved images should reflect all the semantic parts of the query image; on the one hand, the sparse property of ℓ^1-reconstruction forces the algorithm to retrieve only a few semantically related images. The two points above make the retrieved images tend to have sparse but diverse labels which could avoid being dominated by certain concepts and therefore could potentially annotate all the objects in the query image. The sparse coding algorithm for multi-label images is given as follows:

1. **Input:** The training data $Y = [y_1, y_2, \cdots, y_N], y_i \in \mathbb{R}^{m_2}$. A query image $y^t \in \mathbb{R}^{m_2}$.
2. **Sparse coding:** The sparse coding of the query image over all training images is obtained by solving the optimization problem,
$$\min_{\alpha^t} \quad \lambda \|\alpha^t\|_1 + \frac{1}{2}\|y^t - B\alpha^t\|_2^2,$$
where $B = [Y, I] \in \mathbb{R}^{m_2 \times (N+m_2)}$ and $\alpha^t \in \mathbb{R}^{N+m_2}$.
3. **Output:** α^t.

2.3.3 Label propagation to query image

The image annotation process can be considered as the inverse process of the MLE. In MLE, the sparse semantic relations from the label vectors are expected to be transformed to the feature space, while in image annotation process, the sparse semantic relations are transformed from the feature space to the label vector space. Denote the label vector of the query image as c^t, and then its values can be propagated from the training images by,

$$c^t = C\alpha^t, \tag{16}$$

where C is the label matrix of the training images, and α^t is the ℓ^1 sparse reconstruction coefficients. The top labels with the largest values in c^t are considered as the final annotations of the query image, and the values are also stored to facilitate semantic retrieval as described in the next section.

3. Experiments

In this section, we systematically evaluate the effectiveness of the proposed multi-label sparse coding framework

(MSC) for automatic image annotation task by comparing with existing state-of-the-art algorithms on two popular benchmark databases.

3.1. Experimental Setup

3.1.1 Datasets

Two datasets, *i.e*. Corel5k and Corel30k, are used for the comparison evaluations.

Corel5k dataset is a basic comparative dataset for recent research works on image annotation [3][9][10][11][14]. There are 5,000 images from 50 Stock Photo CDs in this dataset. Each CD includes 100 images on the same topic. Each image is annotated with 1 to 5 keywords and totally there are 374 keywords in the dataset. Out of the 5,000 images, 4,500 images are used for model training and the other 500 images are used for testing. The partition of the dataset is the same as that in [11].

Corel30k dataset is an extension of the Corel5k dataset based on a substantially larger database, which tries to correct some of the limitations in Corel5k such as small number of examples and small size of the vocabulary. Corel30k dataset contains 31,695 images and 5,587 keywords. Out of the 31,695 images, 90 percent are used for model training (28,525 images) and 10 percent for testing (3,170 images). As in [3], only the keywords (950 in total) that are used as annotations for at least 10 images are trained.

3.1.2 Evaluation measures

The image annotation performance is evaluated by comparing the results from different algorithms with the human-labeled ground-truths. Similar to existing works [3][9][10][11][14], for each testing image, we use the top five annotations with the largest posterior probability or largest propagation scores as the final annotations. Precision and recall of every keyword in the testing set were used as the performance measures. Recall of a word w_i is defined as the number of images correctly annotated with w_i divided by the number of images that have w_i in the ground truth annotation. Precision of w_i is defined as the number of correctly annotated images divided by the total number of images annotated with w_i. Both measures are averaged over the set of keywords that appear in the testing set as in [3][9][10][11][14]. Moreover, we also consider the number of words with nonzero recalls, which provides an indication of how many words the system has effectively learned.

We also evaluated the semantic retrieval performance as in [3][10]. First, the top five annotations obtained from the annotation algorithm are assigned to the corresponding image. Then, given a query word, the system will return all the images in the testing set whose top five annotations contain the query term, ranked according to the label value propagated based on sparse reconstruction for each testing image (see Eqn. 16, used in MSC) or the probabilities of that word generated by these images (used in SML, et al.). We use a metric called mean average precision to evaluate the retrieval performance. Given the query word and the top n images retrieved from the testing set, precision is the percentage of images which are relevant. Average precision is the average of precision values at the ranks where relevant[1] items occurs, which is further averaged over all single word queries in the testing set to obtain *mean average precision*.

3.2. Results on Corel5k Dataset

3.2.1 Results on automatic image annotation

Table 1 lists the comparison results of automatic image annotation on the Corel5k dataset. Various state-of-the-art algorithms are compared, including the co-occurrence model (Co-occ.) [17], the machine translation model (MT) [9], the cross-media relevance model (CMRM) [11], the continuous relevance model (CRM) [14], CRM with rectangular regions as input (CRM-Rect) [10], the multiple bernoulli relevance model (MBRM) [10], and the supervised multiclass labeling model (SML) [3]. For SML, we adapt the results corresponding to the best parameters in [3]. We also provide the results of MSC without the multi-label linear embedding (MLE) dimensionality reduction part, and the results of k-nearest-neighbors (KNN) algorithm using the supervector representation introduced in Section 2.1. The parameter k in KNN is selected from 1 to 50 corresponding to the best F_1 value ($F_1 = 2 \times precision \times recall/(precision + recall)$). The parameters of MSC are tuned in training set. Results are reported for all 260 words in the testing set. They are also reported for the top 49 annotations to make a direct comparison with the works in [9][10][11][14]. From the results in Table 1, we can draw the following conclusions. First, the proposed MSC algorithm achieves the best performance, exhibiting a gain of 9 and 10 percent in precision and recall respectively compared with SML, which is one of the most popular and effective algorithms in image annotation field. Second, KNN using supervectors outperforms many state-of-the-art algorithms, which shows the effectiveness the feature representation in this paper. Third, base on the same basic feature representation, MSC algorithms with and without MLE both produce superior performance over KNN, which shows the power of the algorithmic part of MSC. Fourth, MSC further improves the performance of the version without MLE dimensionality reduction part, which shows the effectiveness of MLE in the whole framework. Notice that in Co-occ. and SML, there are no statistics for the top 49 keywords in the corresponding papers [3][17].

Fig. 2 presents the precision-recall curves of MSC and

[1] Here “relevant” means that the ground-truth annotations of this image contain the query keyword.

Table 1. Performance comparison of different automatic image annotation algorithms on the Corel5k dataset.

Algorithm	Co-occ. [17]	MT [9]	CMRM [11]	CRM [14]	CRM-Rect [10]	MBRM [10]	SML [3]	KNN (supervector)	MSC (no MLE)	MSC
# words with recall > 0	19	49	66	107	119	122	**137**	133	133	136
Results on all 260 words										
Mean Per-word Recall	0.02	0.04	0.09	0.19	0.23	0.25	0.29	0.30	0.31	**0.32**
Mean Per-word Precision	0.03	0.06	0.10	0.16	0.22	0.24	0.23	0.20	0.24	**0.25**
Results on 49 best words, as in [9][10][11][14]										
Mean Per-word Recall	-	0.34	0.48	0.70	0.75	0.78	-	0.76	**0.83**	0.82
Mean Per-word Precision	-	0.20	0.40	0.59	0.72	0.74	-	0.61	0.72	**0.76**

Table 2. Comparison of MSC annotations with ground-truth annotations on Corel5k (top lines) and Corel30k (bottom lines).

Human Annotation	sky jet plane smoke	sky water ships	buildings light harbor skyline	sky tree ice frost	tree beach people palm
MSC Annotation	sky jet plane *flight* smoke	sky water ships *island rocks*	buildings light harbor skyline *night*	sky tree *snow* ice frost	*water* tree beach people palm
Human Annotation	grass cat lion mane	trees building garden fountain	field horses mare foal	close-up fungus mushroom lichen	sky water elephant bull
MSC Annotation	grass cat *head* lion mane	*sky* trees building *flowers* garden	*grass* field horses mare foal	close-up *plant* fungus mushroom lichen	sky water elephant bull *trunk*

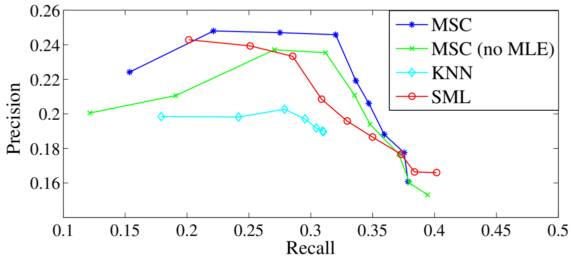

Figure 2. Comparison precision-recall curves of MSC and SML for automatic image annotation on Corel5k dataset.

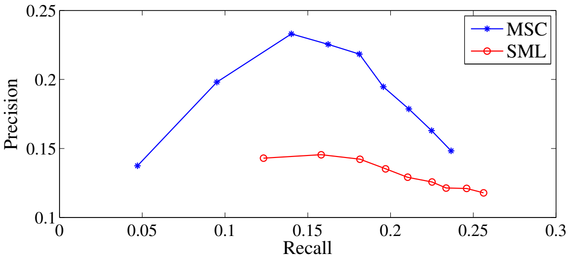

Figure 3. Comparison precision-recall curves of MSC and SML for automatic image annotation on the Corel30k dataset.

SML on the Corel5k dataset, with the number of annotations from 2 to 10. Note that we do not draw the points with the number of fixed annotations larger than 10, since 1) the sparse property of MSC guarantees that there will not be too many semantically related neighbors or labels for each image; and 2) for images in the Corel5k there are at most 5 annotation for one image, and there do not exist many salient objects in real images. From Fig. 2 we can see that MSC consistently outperforms SML. Moreover, the curves of KNN and MSC without MLE are also shown in Fig. 2, from which we can see that MSC also outperforms these two algorithms, which shows the effectiveness of different components of MSC. Notice that the parameter of k in KNN algorithm was tuned to be 2 due to its best F_1 performance compared with other values of k, which however made it only annotate images with a very limited number of labels. Table 2 presents some examples of the annotations produced by MSC, which contain at least one mismatched label compared with ground-truth labels (perfectly matched annotations are not listed here). The results in Table 2 show that, when the system annotates an image with a label not contained in the ground-truth label set, this label is still frequently plausible.

3.2.2 Results on semantic retrieval

Table 3 lists the semantic retrieval results. Notice that another group of SML results (called SML-JSM here) are also listed. SML-JSM results were reported in [4], which used a different group of parameters to achieve superior performance on semantic retrieval task but inferior performance on image annotation task compared with SML in [3]. Therefore we report the semantic retrieval results rather than image annotation results of SML-JSM in this work.

From Table 3 we can observe that, the proposed MSC algorithm significantly outperforms SML and MBRM. More specifically, the MSC algorithm achieves a gain of 40 percent mean average precision on all 260 words over SML-JSM, and a gain of 25 percent on the set of words that have

Table 3. Semantic retrieval results on Corel5k and Corel30k datasets.

	Mean Average Precision for Corel5k Dataset					for Corel30k Dataset	
Algorithm	CRM-Rect[10]	MBRM[10]	SML[3]	SML-JSM[4]	MSC	SML-JSM[4]	MSC
All words	0.26	0.30	0.31	0.30	**0.42**	0.21	**0.32**
Words with recall > 0	0.30	0.35	0.49	0.63	**0.79**	0.47	**0.84**

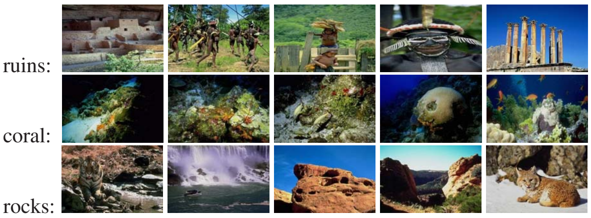

Figure 4. Semantic retrieval results on the Corel30k dataset. Each row shows the top five matches to a semantic query. From top to bottom: “ruins”, “coral”, and “rocks”.

positive recalls. Notice that the annotation results reported in last subsection ignore the rank order of results, and the mean average precision particularly involves the rank order of the semantic retrieval results [10]. The superior performance of MSC in semantic retrieval task shows that MSC could not only annotate images with more correct labels, but also could annotate different images using the same label with relatively proper weights.

3.3. Results on Corel30k Dataset

The Corel30k dataset provides a much larger database size and vocabulary size compared with Corel5k. Corel30k is a very new dataset, and only SML has reported results on it. Since SML has proved its superiority over existing state-of-the-art algorithms, here we only compare the proposed MSC algorithm with SML algorithm on this dataset. Fig. 3 shows the precision-recall curves of MSC and SML on the Corel30k dataset, by selecting fixed number of annotations. From Fig. 3 we can see that, although the recall of MSC is a little smaller than SML, there is huge superiority of MSC over SML on precision. More specifically, with five annotations, the MSC algorithm achieves a gain of 67 percent precision over SML, with only a lose of 18 percent recall. Moreover, the superior performance of MSC on precision directly results in its great semantic retrieval performance. We compare MSC with SML-JSM [4] in semantic retrieval experiments, since there are no semantic retrieval results reported in [3] for Corel30k dataset. From Table 3 we can also see the great improvements of the proposed MSC algorithm over SML-JSM. Besides the strength in annotation precision, another reason of the distinct superiority of MSC on semantic retrieval may be that the label propagation under the reconstruction-based way could be more reasonable than the word-image probability in SML-JSM.

Some annotation results with at least one mismatched label on Corel30k are shown in Table 2, which shows the effectiveness of our proposed MSC framework for automatic image annotation task.

In Fig. 4 we illustrate the retrieval results obtained with several challenging visual concepts being queries, which show the visual appearance diversity of the returned images.

Acknowledgement

This work is supported by NRF/IDM Program, under research Grant NRF2008IDM-IDM004-029.

References

[1] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *TPAMI*, 2002.

[2] D. Blei and M. Jordan. Modeling annotated data. *SIGIR*, 2003.

[3] G. Carneiro, A. Chan, P. Moreno, et al. Supervised Learning of Semantic Classes for Image Annotation and Retrieval. *TPAMI*, 2007

[4] A. Chan, P. Moreno, and N. Vasconcelos. Using Statistics to Search and Annotate Pictures: an Evaluation of Semantic Image Annotation and Retrieval on Large Databases. *Joint Statistical Meetings (JSM)*, Seattle, 2003.

[5] E. Chang, G. Kingshy, G. Sychay, et al.CBSA: content-based soft annotation for multimodal image retrieval using Bayes point machines. *TCSVT*, 2003.

[6] C. Cusano, G. Ciocca, and R. Schettini. Image annotation using SVM. *Proc. of Internet imaging V*, 2004.

[7] A. Dempster, N. Laird and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 1977.

[8] D. Donoho. For most large underdetermined systems of linear equation the minimal l_1-norm solution is also the sparsest solution. *Comm. on Pure and Applied Math*, 2006.

[9] P. Duygulu, and K. Barnard. Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. *ECCV*, 2002.

[10] S. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. *CVPR*, 2004.

[11] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic Image Annotation and Retrieval Using Cross-media Relevance Models. *SIGIR*, 2003.

[12] I. Joliffe. Principal component analysis. *Springer-Verlag, New York*, 1986.

[13] F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. *CVPR*, 2006.

[14] V. Lavrenko, R. Manmatha, and J. Jeon. A Model for Learning the Semantics of Pictures. *NIPS*, 2003.

[15] C. Lee, C. Lin, and B. Juang. A study on speaker adaptation of the parameters of continuous density hidden Markov models. *TASP*, 1991.

[16] J. Li and J. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *TPAMI*, 2003.

[17] Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. *MISRM*, 1999.

[18] D. Reynolds, T. Quatieri, and R. Dunn. Speaker Verification using Adapted Gaussian Mixture Models. *Digital Signal Processing*, 2000.

[19] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 2000.

[20] A. Torralba, R. Fergus, and W. Freeman. Tiny images. *MIT-CSAIL-TR-2007-024*, 2007.

[21] C. Wang, F. Jing, L. Zhang, and H. Zhang. Image Annotation Refinement using Random Walk with Restarts. *ACM Multimedia* 2006.

[22] C. Wang, F. Jing, L. Zhang, and H. Zhang. Scalabel Search-based Image Annotation. *Multimedia Systems*, 2008.

[23] C. Wang, L. Zhang, and H. Zhang. Learning to Reduce the Semantic Gap in Web Image Retrieval and Annotation. *SIGIR*, 2008.

[24] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma Robust Face Recognition via Sparse Representation. *TPAMI*, 2008.

[25] S. Yan, X. Zhou, M. Liu, J. Mark, and T. Huang. Regression from Patch-kernel. *CVPR*, 2008.

[26] X. Zhou, M. Wang, Q. Zhang, J. Zhang, and B. Shi. Automatic image annotation by an iterative approach: incorporating keyword correlations and region matching. *CIVR*, 2007.