Max-Margin Dictionary Learning for Multiclass Image Categorization

Xiao-Chen Lian[1], Zhiwei Li[3], Bao-Liang Lu[1,2], and Lei Zhang[3]

[1] Dept. of Computer Science and Engineering, Shanghai Jiao Tong University, China
[2] MOE-MS Key Lab for BCMI, Shanghai Jiao Tong University, China
[3] Microsoft Research Asia
lianxiaochen@gmail.com, zli@microsoft.com,
bllu@sjtu.edu.cn, leizhang@microsoft.com

Abstract. Visual dictionary learning and base (binary) classifier training are two basic problems for the recently most popular image categorization framework, which is based on the bag-of-visual-terms (BOV) models and multiclass SVM classifiers. In this paper, we study new algorithms to improve performance of this framework from these two aspects. Typically SVM classifiers are trained with dictionaries fixed, and as a result the traditional loss function can only be minimized with respect to hyperplane parameters (w and b). We propose a novel loss function for a binary classifier, which links the hinge-loss term with dictionary learning. By doing so, we can further optimize the loss function with respect to the dictionary parameters. Thus, this framework is able to further increase margins of binary classifiers, and consequently decrease the error bound of the aggregated classifier. On two benchmark dataset, Graz [1] and the fifteen scene category dataset [2], our experiment results significantly outperformed state-of-the-art works.

Keywords: Bag of visual words, Dictionary learning, Max margin.

1 Introduction

Visual recognition is one of the fundamental challenges in computer vision, which targets at automatically assigning class labels to images based on their visual features. In recent years, many methods have been proposed [2,3,4,5], in which the framework that combines bag of visual words (BOV) model with SVM-based multiclass classifiers [3,4] has achieved state-of-the-art performance in various benchmark tasks [2,6,7]. To further improve the performance of this framework, we study two basic problems of it in this paper.

First, how to learn a better BOV model? A core issue of this framework is generating a dictionary that will be effective for classifier training. Most of existing approaches adopt unsupervised clustering manners, whose goals are to keep sufficient information for representing the original features by minimizing a reconstruction error or expected distortion (e.g. K-means [8], manifold learning [9] and sparse coding [4]). Due to the ignorance to supervisory information,

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 157–170, 2010.

the histogram representations of images over the learned dictionary may not be optimal for a classification task. Therefore, a highly probably better choice is to incorporate discriminative information (i.e. class labels) into the dictionary construction process.

Second, how to train a better SVM classifier? SVM-based multiclass classifiers are usually constructed by aggregating results of a collection of binary classifiers. The most popular strategies are *one-vs.-one* where all pairs of classes are compared, and *one-vs.-all* where each class is compared against all others. The performance of the binary classifiers directly affects the performance of the aggregated classifier. Thus, a straightforward idea to improve the multiclass classifiers is improving the individual binary classifier.

Existing approaches typically deal with the above two problems separately: dictionaries are first generated and classifiers are then learned based on them. In this paper, we propose a novel framework for image classification which unifies the dictionary learning process with classifier training. The framework reduces the multiclass problem to a collection of one-vs-one binary problems. For each binary problem, classifier learning and dictionary generation are conducted iteratively by minimizing a unified objective function which adopts the maximum margin criteria. We name this approach Max-Margin Dictionary Learning (MMDL). We evaluate MMDL using two widely used classifier aggregation strategies: majority voting and Decision Directed Acyclic Graph (DDAG) [10]. Experimental results show that by embedding the dictionary learning into classifier training, the performance of the aggregated multiclass classifier is improved. Our results outperformed state-of-the-art results on Graz [1] and the fifteen scene category dataset [2].

2 Related Work

Supervised dictionary learning has attracted much attention in recent years. Existing approaches can be roughly categorized into three categories.

First, constructing multiple dictionaries, e.g. [11] wraps dictionary construction inside a boosting procedure and learns multiple dictionaries with complementary discriminative power, and[12] learns a category-specific dictionary for each category.

Second, learning a dictionary by manipulating an initial dictionary, e.g. merging visual words. The merging process could be guided by mutual information between visual words and classes [1], or trade-off between intra-class compactness and inter-class discrimination power [13]. The performance of such approaches is highly affected by the initial dictionary since only merging operation is considered in them. To ease this problem a large dictionary is required at the beginning to preserve as much discriminative abilities as possible, which is not guaranteed though.

Third, learning a dictionary via pursuing a descriptor-level discriminative ability, e.g. empirical information loss minimization method [14], randomized decision forests [15,16], and sparse coding-based approaches [17,18,19]. Most of these approaches are first motivated from coding of signals, where a sample (or

say signal) is only analogous to a local descriptor in an image rather than a whole image which is composed of a collection of local descriptors. Actually, this requirement is over strong since local descriptors of different objects are often overlapped (i.e. a white patch may appear both in the sky and on a wall).

Moreover, depending on whether dictionary learning and classifier training are unified in a single process or not, the above approaches can be further categorized to two categories. Most of them take two separate processes, e.g. [11,15,16,12,1,13,14], in which a dictionary is first learned and then a classifier is trained over it. Therefore, the objectives of the two processes are likely to be inconsistent. The other category of approaches takes a similar strategy as ours, that is, they combine the two processes by designing a hybrid generative and discriminative energy function. The discrimination criteria used include softmax discriminative cost functions [17,18] and Fisher's discrimination criterion [19]. However, existing approaches put the discrimination criteria on individual local descriptors rather than image-level representations, i.e. histogram representations of images.

After this paper was submitted, two additional related works were published, which also consider learning dictionary with image-level discriminative criteria. Yang *et al.* [20] used sparse coding for dictionary learning and put a classification loss in the model. Boureau *et al.* [21] used regularized logistic cost.

3 Max-Margin Dictionary Learning

In this section, we first introduce the motivation of incorporating max-margin criteria into dictionary learning process. Then the Max-Margin Dictionary Learning (MMDL) algorithm is described and the analysis on how max-margin criterion affects the dictionary construction is given. Finally, we describe the pipeline of the whole classification framework.

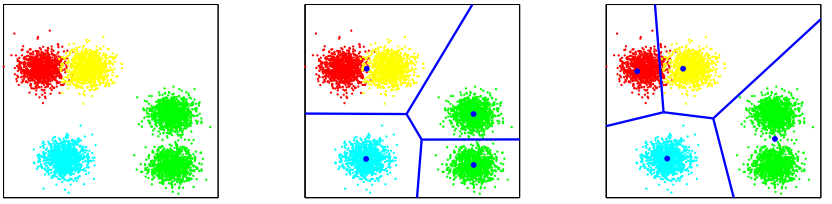
3.1 Problem Formulation

Suppose we are given a corpus of training images $\mathcal{D} = \{(I^d, c^d)\}_{d=1}^{D}$, where $I^d = \{x_1^d, x_2^d \ldots, x_{N_d}^d\}$ is the set of local descriptors (i.e. SIFT [22]) extracted from image d, and $c^d \in \{+1, -1\}$ is the class label associated with I^d. A dictionary learning method will construct a dictionary which consists of K visual words $V = \{v_1, v_2, \ldots, v_K\}$. A descriptor x_i^d from image d is quantized to a K-dimension vector ϕ_i^d where

$$\phi_i^d[k] = \begin{cases} 1, & k = \underset{w}{\operatorname{argmin}} \parallel x_i^d - v_w \parallel_2 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

for hard assignment and

$$\phi_i^d[k] = \frac{\exp(-\gamma \parallel x_i^d - v_k \parallel_2^2)}{\sum_{k'=1}^{K} \exp(-\gamma \parallel x_i^d - v_{k'} \parallel_2^2)}, \quad k = 1, \ldots, K \tag{2}$$



(a) (b) (c)

Fig. 1. (a) The 2-D synthetic data. Colors of points indicate their categories. (b) K-means results and Voronoi boundary of the words. Red and yellow points are grouped onto the same word and as a result cannot be distinguished. (c) A correct partition where categories information are completely preserved at the price of distortion. This figure should best be viewed in color

for soft assignment [23]. Then image d is represented by the histogram

$$\Phi^d = \frac{1}{N_d} \sum_{i=1}^{N_d} \phi_i^d, \tag{3}$$

and the set of couples $\{(\Phi^d, c^d)\}$ are used to train the classifiers.

Traditionally, the dictionary V is learned by minimizing the reconstruction error or overall distortion. For example, K-means clustering algorithm solves the following problem

$$\min_V \sum_{d=1}^{D} \sum_{i=1}^{N_d} \min_{k=1\ldots K} \parallel x_i^d - v_k \parallel_2^2 \tag{4}$$

However, as the learning process does not utilize the category information, the resulted histograms may not be optimal for classification. We illustrate the problem on a toy data shown in Figure 1(a). K-means groups the red and yellow clusters into one word (Figure 1(b)) and separates the two green clusters to two words because it only considers to minimizing the overall distortion. As a result, red and yellow clusters cannot be distinguished through their histogram representations. A correct partition is shown in Figure 1(c); although the dictionary has a bigger distortion, it is more discriminative than the dictionary obtained by K-means.

3.2 MMDL

Our solution to the above problem is to combine classifier training and dictionary learning together. Motivated by the loss function in SVM, we design the following objective function

$$\mathcal{L}(V, W) = \frac{1}{2} \parallel W \parallel_2^2 + C \sum_d \max\left(0, 1 - c_d \langle W, \Phi^d \rangle\right) \tag{5}$$

Algorithm 1. MMDL

Input: A training set $\mathcal{D} = \{(I^d, c^d)\}_{d=1}^{D}$; number of iteration T; convergence threshold ϵ

Initialize the dictionary V
repeat{Alternate dictionary and classifier learning}
  Fix V, $W = \underset{W'}{\operatorname{argmin}}\, \mathcal{L}(V, W')$
  $V_0 = V$; $\mathcal{L}_{\min} = \mathcal{L}(V_0, W)$; $V_{\min} = V_0$
  for $t = 1$ **to** T **do**
    $V_t = V_{t-1} - \lambda_t \nabla \mathcal{L}(V_t, W)$ (∇ denotes subgradient operators)
    if $\mathcal{L}(V_t, W) < \mathcal{L}_{\min}$
      $\mathcal{L}_{\min} = \mathcal{L}(V_t, W)$
      $V_{\min} = V_t$
    end if
  end for
  $V = V_{\min}$
until convergence rate of $\mathcal{L}(V, W)$ is below ϵ
return dictionary V and classifier W

where Φ^d is computed through Eq. (2) and (3), $W = (w_1, \ldots, w_K)^\top$ is a hyperplane classifier and C is the trade-off factor. It is noted that:

1) We omit the offset (i.e. b in a standard SVM classifier) since the L1-norm of Φ^d is always one.

2) In terms of learning a dictionary, the objective of Eq. 5 is different from Eq. 4. Eq. 4 minimizes the distortion of a dictionary, while Eq. 5 aims at finding a dictionary which minimizes a SVM loss.

3) For computational reason, we only support a linear SVM classifier in this framework. The main reason is that using a non-linear kernel in Eq. 5 makes learning difficult since the analytical forms of project functions are usually unknown and computing their derivatives is intractable. However, as later shown in experiments, using the dictionary learned by MMDL, linear SVM classifier outperforms the non-linear SVM classifiers that use dictionary learned by K-means.

By minimizing $\mathcal{L}(V, W)$, we obtain a dictionary V and a binary classifier W which are expected to be with a large margin. The minimization is proceeded as a two-phase iteration. In the first phase, the dictionary V is fixed, and the computation of W becomes a standard linear SVM problem, in which the first term punishing the model complexity and the hinge loss term punishing the training error. In the second phase, V is computed by fixing W. Eq. 5 links the dictionary learning and classifier training processes. In traditional BOV+SVM framework where the two processes are separated, the optimization of Eq. 5 involves only the first phase. While in MMDL, we can further minimize Eq. 5 by doing the second phase, and the margins are further increased. Due to the presence of both the non-linearity of Φ^d and the non-differentiability of the hinge loss, we apply subgradient method [24] which is widely used with non-differentiable objective functions.

The iteration used in subgradient method is quite similar to that of steepest descent, except the following two differences (refer to [24] for details): (1) As the

objective function may not have derivatives at all points, the search direction is the negative of the subgradient; (2) It may happen that the search direction is not a descent direction, therefore a list recording the lowest objective function value found so far is maintained.

Algorithm 1 depicts the complete MMDL algorithm. In line 3, W is computed by a standard SVM solver. In line 7, the dictionary V is updated according to the subgradient and the step size λ_t. The subgradient of a convex function f at point x_0 is a nonempty closed interval $[f^-(x_0), f^+(x_0)]$ where $f^-(x_0)$ and $f^+(x_0)$ are the left- and right-sided derivatives respectively. The interval reduces to a point when f is differentiable at x_0. In this case, $f^-(x_0) = f^+(x_0) = \partial f(x_0)$.

Denote $\langle W^\top, \Phi^d \rangle$ by $h^d(V)$, then the hinge loss term for image d is $\mathcal{L}^d = \max(0, 1 - c^d h^d(V))$. When $c^d h^d(V) < 1$, $\mathcal{L}^d = 1 - c^d h^d(V)$ is differentiable. Its subgradient at $v_k (k = 1 \ldots, K)$ equals to its derivative

$$\begin{aligned}
\frac{\partial \mathcal{L}^d}{\partial v_k} &= \frac{\partial}{\partial v_k}(-\frac{c^d w_k}{N_d} \sum_{i=1}^{N_d} \phi_i^d[k]) \\
&= -\frac{c^d w_k}{N_d} \sum_{i=1}^{N_d} \frac{2\gamma(x_i^d - v_k) \exp(-\gamma \parallel x_i^d - v_k \parallel_2^2)}{\left(\sum_{k'=1}^{K} \exp(-\gamma \parallel x_i^d - v_{k'} \parallel_2^2)\right)} \\
&\quad + \frac{c^d w_k}{N_d} \sum_{i=1}^{N_d} \frac{2\gamma(x_i^d - v_k) \exp(-\gamma \parallel x_i^d - v_k \parallel_2^2)^2}{\left(\sum_{k'=1}^{K} \exp(-\gamma \parallel x_i^d - v_{k'} \parallel_2^2)\right)^2} \\
&= -\frac{2c^d w_k}{N_d} \sum_{i=1}^{N_d} \gamma(x_i^d - v_k) \left(\phi_i^d[k] - (\phi_i^d[k])^2\right).
\end{aligned} \tag{6}$$

When $c^d h^d(V) \geq 1$, the subgradient $\nabla \mathcal{L}^d = 0$ for all $v_k (k = 1 \ldots, K)$, which means we pick the right-sided derivative of the hinge loss term. The visual word v_k is then updated by

$$v_k^{t+1} = v_k^t - \lambda_t \sum_{d \in \mathcal{X}} \frac{\partial \mathcal{L}^d}{\partial v_k^t} \tag{7}$$

where $\mathcal{X} = \{d \mid c^d h^d(V) < 1\}$ is the set of indices of the images that lie in the margin or are misclassified by W. We name these images as *effective images* because only these images are involved in the dictionary update equation.

Analysis. We examine the update rules to see how the *effective images* take effect in the dictionary learning process. For better understanding, we reformat Eq. (7) as:

$$v_k' = v_k + \sum_{d \in \mathcal{X}} \frac{2\gamma\lambda}{N_d} \sum_{i=1}^{N_d} s_i^d[k] \cdot t_i^d[k] \cdot p_i^d[k], \tag{8}$$

where

$$\begin{aligned} s_i^d[k] &= \text{sign}(\text{c}^\text{d}\text{w}_\text{k}) \\ p_i^d[k] &= \frac{x_i^d - v_k}{\parallel x_i^d - v_k \parallel_2^2} \\ t_i^d[k] &= w_k \left(\phi_i^d[k] - (\phi_i^d[k])^2\right) \parallel x_i^d - v_k \parallel_2^2 . \end{aligned} \tag{9}$$

Intuitively, the update of v_k is the net force of all local descriptors in effective images. Each descriptor x_i^d pushes or pulls v_k along a direction. The direction of the force is determined by $s_i^d[k]$ and $p_i^d[k]$. If $s_i^d[k] > 0$, it means that the k-th word is positive to correctly predicting the label of image d (i.e. the image favors larger $\phi_i^d[k]$); otherwise, it means that we expect that the image d should have smaller $\phi_i^d[k]$. As a result, when $s_i^d[k] > 0$, v_k will be pulled to be near to descriptor x_i^d, and when $s_i^d[k] < 0$, it will be pushed to be far away from x_i^d. Therefore moving v_k according to Eq. (8) will decreases the hinge loss $\mathcal{L}(V, W)$. The strength of x_i^d's force on v_k is determined by $t_i^d[k]$, which is proportional to w_k, a quadratic term $\phi_i^d[k] - (\phi_i^d[k])^2$ and $\parallel x_i^d - v_k \parallel_2^2$ (Euclidean distance between x_i^d and v_k). In the following, we give an intuitive explanation about $t_i^d[k]$.

From the feature selection's point of view, hyperplane W plays a role as visual word selector. If the absolute value of w_k is very small, it means the word is not important for the classifier, and thus the update to the corresponding v_k could be minor.

Before analyzing the latter two terms, we first note that $\phi_i^d[k]$ and $e_i^d[k] = \parallel x_i^d - v_k \parallel_2^2$ are both related to the distance between descriptor x_i^d and visual word v_k. The former one measures the relative distance from v_k to $x_i^d[k]$ compared with other visual words, while the latter is the absolute distance. We first consider the case when the force x_i^d exerts on v_k is pull. When $\phi_i^d[k]$ is very large, moving v_k for a distance may not increase the distortion too much. Therefore the quadratic term $\phi_i^d[k] - (\phi_i^d[k])^2$ will be small, indicating that x_i^d does not hold v_k strongly and allows other descriptors to move it. If $\phi_i^d[k]$ is quite small, v_k is relative far from x_i^d, and the quadratic term will also be small which means the force should be small as moving v_k close to x_i^d may cause large distortion. If e_i^d is large but other visual words are much far away from x_i^d, moving v_k close is acceptable. Otherwise x_i^d may not pull v_k over as the distortion may increase. Similar discussion can be made when the force is push.

3.3 Base Classifier and Aggregation Strategy

The hyperplane classifier W obtained during dictionary learning can be used as the base classifier. Although it is a linear classifier, in our experiments it outperforms the SVM classifiers with non-linear kernels which are trained based on unsupervisedly learned dictionaries.

Any strategy that aggregates binary classifiers can be used in our framework, e.g. majority voting (VOTE), DDAG and Error-Correcting Codes (ECC) [25]. In this paper we evaluate the combination of MMDL with VOTE and DDAG.

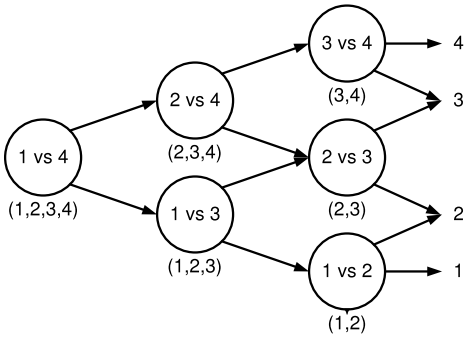


Fig. 2. A four-class DDAG. The list under each node contains the class labels that will remain when the evaluation reaches the node.

We use DDAG as an example to demonstrate how we aggregate MMDL base classifiers.

DDAG is a lightweight architecture which is efficient to evaluate. Besides, the theoretic advantage of DDAG is that when the base classifiers are hyperplanes, enlarging the margin of all nodes in a DDAG will lower the bound on the generalization error. For a C-class problem, DDAG has $C(C-1)/2$ nodes, each one distinguishing a pair of classes a and b. On each such node, MMDL learns a dictionary and a corresponding classifier using the subset of images labeled by a or b. The evaluation of a DDAG $\mathcal{G}$ on a test point is equivalent to operating on a list which initially contains all classes. The point x is first test by the node that corresponds to the first and last classes on the list and one of the class is eliminated from the list if the node prefers the other one. DDAG then proceeds to test the first and last classes on the new list. The process terminates when only one class remains in the list and x is judged to be that class (see Fig. 2).

3.4 Time Complexity

Let C be the number of categories, K be the size of each two-class dictionary, and L be the dimension of descriptor. Suppose the number of descriptors from each categories is N. The time complexity for learning all two-class dictionaries is $O(C \times (C-1) \times N \times K \times L \times T_s \times T_i)$, where T_s and T_i are the number of iterations for subgradient and two-phase iteration respectively. It is comparable to the complexity of learning a same size dictionary by K-means, i.e. $O(C \times N \times \frac{C(C-1)}{2} \times K \times L \times T)$, where T is the number of iterations for K-means to converge.

4 Experiments

In this section, we report results on two benchmark datasets: Graz-02 [1] and fifteen scene dataset [2]. We use a variant setting of SIFT to generate local

Table 1. A comparison of the pixel precision-recall equal error rates on Graz-02 dataset. Dictionary size is 200

	cars	people	bicycles
AIB200-KNN [1]	50.90	49.70	**63.80**
AIB200-SVM [1]	40.10	50.70	59.90
MMDL+HP	**54.27**	**55.81**	63.55

descriptors. In our implementation, a patch is divided into 2×2 subpatches rather than the 4×4 schema in the standard SIFT setting [22]. For each subpatch a 8-bin histogram of oriented gradients is calculated. Thus, our local descriptor is 32-d. We adopt this setting mainly for its computational efficiency and Uijlings *et al.* [26] reported that 2×2 SIFT performed marginally better but never worse than the 4×4 SIFT.

In all experiments, we perform processing in gray scale, even when color images are available. We initialize the dictionary V by randomly selecting descriptors from training data and set the parameters of MMDL as $C = 32$, $\gamma = 1\times10^{-3}$ and $\lambda_t = 1\times10^{-1}$ for all $t = 1\ldots, T$. The number of iterations T for subgradient method is set to be 40, and MMDL converges after about 30 iterations under the convergence threshold $\epsilon = 1 \times 10^{-4}$.

4.1 Object Localization

We first use Graz-02 dataset [1] to evaluate the performance of MMDL for object localization. Graz-02 contains three classes (bicycles, cars and people) with extreme variability in pose, scale and lighting. The task is to label image pixel as either belonging to one of the three classes or background. The baseline approach is another supervised dictionary learning method proposed in [1]. The measure of performance is pixel precision-recall error rate. We follow the same setup as in [1]: for each object, a dictionary that distinguishes foreground objects from background is constructed; when testing, a histogram of frequencies of visual words within the 80×80-pixel window centered at each pixel is computed. A SVM classifier is applied to classify the histogram and a confidence that the pixel belongs to foreground object is returned. Precision and recall are computed according to ground-truth segmentation provided by the dataset. The results when the sizes of dictionaries are 200 are reported in Table 1. MMDL+HP means that we directly used the learned hyperplane classifiers obtained during the dictionary learning. The performance of our approach is significantly better than the baseline approach on the first two classes, and is comparable with [1] on the last class.

4.2 Scene Category Classification

The second dataset we use is the fifteen scene dataset (scene15), which consists of fifteen kinds of scene images, e.g. highway, kitchen and street. As in [2,14], SIFT descriptors of 16×16 patches sampled over a grid with spacing of 8 pixels are computed. 100 images per class are randomly selected for training and the

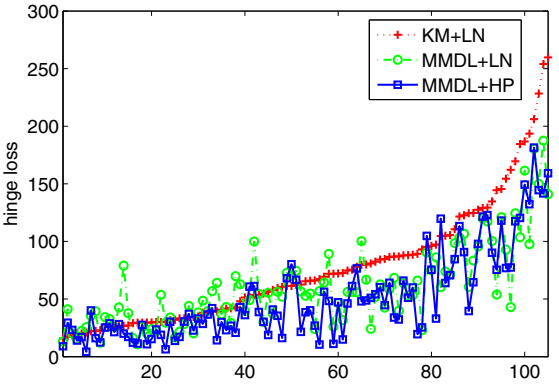


Fig. 3. Comparison of hinge losses on all binary problems obtained by MMDL and K-means on scene15. The hinge losses are computed on test data.

Table 2. Comparison of hinge losses for the top four most confused classes of scene15: *bedroom*, *kitchen*, *living room* and *industrial*

	KM+LN	MMDL+LN	MMDL+HP
bedroom vs. kitchen	137.25 ± 8.23	115.86 ± 3.56	$\mathbf{108.59 \pm 6.35}$
bedroom vs. living room	239.97 ± 9.55	206.62 ± 13.08	$\mathbf{189.93 \pm 32.36}$
bedroom vs. industrial	168.38 ± 2.87	$\mathbf{124.71 \pm 0.49}$	125.25 ± 5.89
kitchen vs. living room	193.47 ± 9.70	173.29 ± 14.23	$\mathbf{166.30 \pm 12.70}$
kitchen vs. industrial	133.34 ± 16.91	95.78 ± 7.56	$\mathbf{88.24 \pm 8.24}$
living room vs. industrial	222.74 ± 24.41	147.55 ± 33.33	$\mathbf{155.82 \pm 16.33}$

rest for testing. We train 105 binary classifiers, one for each pair of classes, with all possible combinations of dictionary learning algorithms and classifier settings. The dictionary learning algorithms are K-means (KM) and MMDL. The quantization of KM uses the soft assignment in Eq. 2 with the same γ as MMDL. Each binary problem use a dictionary with 50 visual words. The classifiers are SVM with linear kernel (LN) and histogram intersection kernel (HI), and the hyperplane-based classifier learned by MMDL (HP). For example, a name "KM+HI+DDAG" means that we adopt K-means to learn a dictionary, histogram intersection kernel to train SVM classifiers, and the DDAG approach to aggregate base classifiers. The experiments are repeated five times and the final result is reported as the mean and standard deviation of the results from the individual runs.

To show the superiority of MMDL over K-means, in Fig. 3 we plot the hinge losses of linear classifiers on all binary problems obtained by the K-means and MMDL. The x-coordinate is the indices of binary classifiers which are sorted in an order that their hinge loss produced by the corresponding KM+LIN method on test set are ascending. We also list the hinge losses of the top four most confused classes (bedroom, kitchen, living room and industrial) in Table 2. In

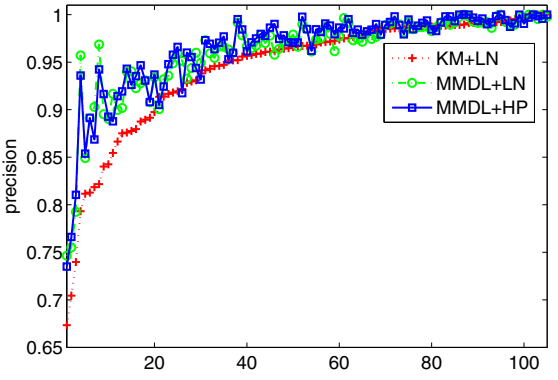


Fig. 4. Comparison of precisions on all binary problems obtained by MMDL and K-means on scene15

Table 3. Comparison of precisions (percentage) for the top four most confused classes of scene15: *bedroom*, *kitchen*, *living room* and *industrial*

	KM+LN	MMDL+LN	MMDL+HP
bedroom vs. kitchen	77.46 ± 2.38	80.04 ± 0.97	$\mathbf{82.69 \pm 1.11}$
bedroom vs. living room	66.70 ± 0.42	70.06 ± 3.26	$\mathbf{72.37 \pm 0.77}$
bedroom vs. industrial	81.96 ± 1.84	86.99 ± 1.43	$\mathbf{87.83 \pm 1.51}$
kitchen vs. living room	72.03 ± 1.43	75.86 ± 3.37	$\mathbf{78.53 \pm 2.26}$
kitchen vs. industrial	86.80 ± 1.92	86.96 ± 0.39	$\mathbf{89.60 \pm 1.41}$
living room vs. industrial	78.66 ± 2.73	$\mathbf{87.55 \pm 3.37}$	85.56 ± 0.94

Table 4. Comparison of precisions (percentage) for all classes of scene15

	KM+HI		MMDL+HP		MMDL+HI	
class	DDAG	VOTE	DDAG	VOTE	DDAG	VOTE
bedroom	34.5 ± 0.9	40.8 ± 1.8	47.1 ± 7.3	$\mathbf{58.0 \pm 5.2}$	46.0 ± 5.7	55.7 ± 5.7
suburb	88.2 ± 3.9	89.4 ± 2.5	91.7 ± 1.1	92.9 ± 2.6	92.9 ± 1.9	$\mathbf{93.9 \pm 1.8}$
kitchen	52.1 ± 1.4	57.0 ± 3.4	$\mathbf{71.2 \pm 0.5}$	69.4 ± 3.4	68.8 ± 6.7	69.1 ± 6.3
living room	49.7 ± 3.8	46.9 ± 4.2	53.3 ± 1.3	51.0 ± 5.0	$\mathbf{61.9 \pm 2.4}$	54.1 ± 3.8
coast	81.0 ± 5.3	82.6 ± 5.5	82.4 ± 1.2	84.6 ± 2.0	86.2 ± 2.8	$\mathbf{90.1 \pm 3.1}$
forest	90.2 ± 1.3	90.8 ± 1.6	$\mathbf{92.3 \pm 1.5}$	$\mathbf{92.3 \pm 1.5}$	90.6 ± 1.8	91.7 ± 1.2
highway	83.8 ± 3.8	84.4 ± 2.5	87.1 ± 1.8	87.1 ± 2.4	87.1 ± 2.5	$\mathbf{88.1 \pm 3.5}$
inside city	65.2 ± 3.9	66.8 ± 3.5	72.1 ± 3.8	72.8 ± 3.6	72.6 ± 1.7	$\mathbf{75.8 \pm 1.1}$
mountain	79.2 ± 1.8	78.5 ± 2.4	83.8 ± 1.3	82.1 ± 1.6	$\mathbf{84.4 \pm 1.1}$	82.5 ± 1.3
open country	68.4 ± 2.6	68.0 ± 2.9	71.5 ± 1.5	73.4 ± 3.2	$\mathbf{80.0 \pm 2.1}$	78.8 ± 2.3
street	84.0 ± 2.4	82.6 ± 2.9	$\mathbf{87.3 \pm 2.1}$	86.5 ± 1.4	86.1 ± 1.5	86.3 ± 2.1
tall building	82.9 ± 0.6	82.0 ± 0.7	77.9 ± 1.0	79.0 ± 5.6	$\mathbf{87.5 \pm 1.0}$	85.3 ± 0.2
office	77.4 ± 1.5	75.9 ± 2.0	82.9 ± 4.8	80.9 ± 3.0	$\mathbf{89.3 \pm 4.0}$	87.5 ± 5.8
store	64.0 ± 5.8	63.1 ± 6.2	68.2 ± 1.2	70.7 ± 3.3	$\mathbf{74.6 \pm 0.5}$	73.8 ± 0.5
industrial	42.3 ± 5.2	42.0 ± 2.8	42.2 ± 5.0	48.3 ± 4.8	55.5 ± 4.3	$\mathbf{58.5 \pm 5.2}$
average	69.5 ± 0.2	70.1 ± 0.1	74.1 ± 1.2	75.3 ± 2.1	77.6 ± 0.3	$\mathbf{78.1 \pm 0.7}$

Table 5. Comparison of average precisions (percentage) on secene15 dataset

	L = 2	L = 3
MMDL+SPM+HP+DDAG	78.34 ± 0.90	82.33 ± 0.39
MMDL+SPM+HP+VOTE	79.15 ± 0.76	83.21 ± 0.45
MMDL+SPM+HI+DDAG	82.23 ± 1.01	$\mathbf{85.98 \pm 0.68}$
MMDL+SPM+HI+VOTE	$\mathbf{82.66 \pm 0.51}$	$\mathbf{86.43 \pm 0.41}$
KM+SPM+HI+DDAG	77.48 ± 1.08	79.65 ± 0.59
KM+SPM+HI+VOTE	77.89 ± 0.50	80.17 ± 0.28
HG [27]	-	85.2
SPM [2]	80.1 ± 0.5	81.4 ± 0.5
ScSPM [4]	-	80.4 ± 0.9
sPACT [3]	-	83.3 ± 0.5

the similar way, we compare their precisions on all binary problems in Fig. 4 and Table 3. We can see that:

1) In terms of both the hinge loss and precision, MMDL based approach is significantly better than K-means based approaches.

2) For the four categories, which KM+LIN does not distinguish well (i.e. classification between the four classes), the improvements obtained by MMDL are significant. For all categories, MMDL outperforms K-means.

Table 4 shows the performance for each category with different dictionary and classifier settings. Our basic approaches, i.e. MMDL+HP+DDAG/VOTE, significantly outperform the baseline approaches (KM+HI+DDAG/VOTE), and with histogram intersection kernel, their performance is even better. With a 200 word universal dictionary, which is obtained by running K-means over SIFT descriptors of randomly sampled 400 images, the linear SVM achieved an average precision at 74.3%[1] which is also lower than our approaches. We also learned a 5250-word universal dictionary by K-means, whose size is equal to the total number of visual words used in MMDL approaches. Its result with histogram intersection kernel is 75.3%. An interesting observation is that without incorporating the max margin term into learning process, using a set of two-class dictionaries is worse than using a single dictionary with enough size. Two-class dictionaries are likely to over fit on training images, and their generalization capabilities are usually weak. While from table 4, we can see that MMDL can boost the performance, which is attributed to the incorporation of max margin criteria.

On scene15, the state-of-the-art results are obtained by applying spatial pyramid matching (SPM) mechanism [2]. We apply it to each binary classifier in our framework. Although our objective function of dictionary learning does not optimize for the SPM representation, our approach achieves the best results as shown in Table 5. To the best of our knowledge, it outperforms all results on

[1] The result is better than the result, $72.2 \pm 0.6\%$, reported in [2]

this dataset reported in recent years [3,4,2,27]. Actually, due to a characteristic of SPM mechanism (i.e. it is a "linear" transform indeed), it can be integrated in our loss function easily.

5 Conclusion

We have proposed a max-margin dictionary learning algorithm, which can be integrated in the training process of a linear SVM classifier to further increase the margin of the learned classifier, and consequently decrease the error bound of the aggregated multi-class classifier. Our preliminary experiment results on two benchmark datasets demonstrate the effectiveness of the proposed approach.

In the future, we are going to study how to directly apply non-linear kernel functions, e.g. histogram intersection kernel and χ^2 kernel, in the SVM classifier. Recently, using spatial information in image classification have drawn much attention. A common problem of these approaches is that the spatial constraints are predetermined and fixed during dictionary learning. We are designing a method that will automatically determine the spatial constraints under the guidance of supervised information.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China (Grant No. 60773090 and Grant No. 90820018), the National Basic Research Program of China (Grant No. 2009CB320901), and the National High-Tech Research Program of China (Grant No. 2008AA02Z315).

References

1. Fulkerson, B., Vedaldi, A., Soatto, S.: Localizing objects with smart dictionaries. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 179–192. Springer, Heidelberg (2008)
2. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proc. CVPR, pp. 2169–2178 (2006)
3. Wu, J., Rehg, J.: Where am I: Place instance and category recognition using spatial PACT. In: Proc. CVPR (2008)
4. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: Proc. CVPR (2009)
5. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: Proc. CVPR, vol. 2, pp. 524–531
6. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Comput. Vis. Image Underst. 106, 59–70 (2007)
7. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge, Results (2009), `http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html`

8. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proc. ICCV., vol. 2, pp. 1470–1477 (2003)
9. Jiang, Y.G., Ngo, C.W.: Visual word proximity and linguistics for semantic video indexing and near-duplicate retrieval. Comput. Vis. Image Underst. 113, 405–414 (2009)
10. Platt, J., Cristianini, N., Shawe-Taylor, J.: Large margin DAGs for multiclass classification. Advances in neural information processing systems 12, 547–553 (2000)
11. Zhang, W., Surve, A., Fern, X., Dietterich, T.: Learning non-redundant codebooks for classifying complex objects. In: Proceedings of the 26th Annual International Conference on Machine Learning (2009)
12. Perronnin, F.: Universal and adapted vocabularies for generic visual categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 1243–1256 (2008)
13. Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. In: Proc. ICCV, pp. 1800–1807 (2005)
14. Lazebnik, S., Raginsky, M.: Supervised learning of quantizer codebooks by information loss minimization. IEEE Trans. Pattern Anal. Mach. Intell. 31, 1294–1309 (2009)
15. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. Advances in neural information processing systems 19, 985 (2007)
16. Shotton, J., Johnson, J., Cipolla, M.: Semantic texton forests for image categorization and segmentation. In: Proc. CVPR (2008)
17. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Supervised dictionary learning. Advances in Neural Information Processing Systems 21 (2009)
18. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Discriminative learned dictionaries for local image analysis. In: Proc. CVPR (2008)
19. Huang, K., Aviyente, S.: Sparse representation for signal classification. Advances in Neural Information Processing Systems 19, 609 (2007)
20. Yang, J., Yu, K., Huang, T.: Supervised Translation-Invariant Sparse Coding. In: Proc. CVPR (2010)
21. Boureau, Y., Bach, F., LeCun, Y., Ponce, J.: Learning Mid-Level Features For Recognition. In: Proc. CVPR (2010)
22. Lowe, D.: Object recognition from local scale-invariant features. In: Proc. ICCV., vol. 2, pp. 1150–1157 (1999)
23. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proc. CVPR (2008)
24. Shor, N., Kiwiel, K., Ruszcayǹski, A.: Minimization methods for non-differentiable functions. Springer, New York (1985)
25. Allwein, E., Schapire, R., Singer, Y.: Reducing multiclass to binary: A unifying approach for margin classifiers. The Journal of Machine Learning Research 1, 113–141 (2001)
26. Uijlings, J., Smeulders, A., Scha, R.: What is the Spatial Extent of an Object? In: Proc. CVPR (2009)
27. Zhou, X., Cui, N., Li, Z., Liang, F., Huang, T.: Hierarchical Gaussianization for Image Classification. In: Proc. ICCV (2009)