A Boosting, Sparsity-Constrained Bilinear Model for Object Recognition

Chunjie Zhang and Jing Liu
National Lab of Pattern Recognition, Chinese Academy of Sciences

Qi Tian
University of Texas at San Antonio

Yanjun Han
Douban

Hanqing Lu and Songde Ma
National Lab of Pattern Recognition, Chinese Academy of Sciences

Using higher-level visual elements to represent images, the authors have developed a sparsity-constrained bilinear model (SBLM) and have combined a set of SBLMs in a boosting-like procedure to enhance performance.

Classifying images and identifying objects in images is a challenging task in many applications such as image retrieval or annotation. Recent research increasingly relies on the bag-of-words (BoW) representation and its corresponding learning model because this representation has generated promising results in various vision tasks including image and object categorization.[1–13] However, the descriptive ability of a histogram-based representation is limited due to the loss of spatial correlation of local features. In cases with large variations between images belonging to the same class, determining how to extract representative structural descriptors and build a discriminative object model has become a timely research topic.

Previous literature has looked at incorporating spatial information with geometric constraints[1] (at significant computational expense) as well as extending the typical BoW representation by associating a set of visual features or considering spatial information (see the ''Related Work in Object Recognition'' sidebar for more details). Although these methods help obtain an enhanced descriptive image representation, it is uncertain whether that representation is discriminative enough to boost categorization performance. In addition, some approaches have focused on the visual-word-level discriminative power, but they lose the possible visual-word-correlation and spatial information.[10,11] Therefore, given the object-recognition task, jointly enhancing the descriptive and discriminative power of image representation using spatial information and building a discriminative object model is a necessary and challenging research topic.

In this article, we import a *component*, or a set of image regions, as a higher-level element to represent an image jointly with the lower-level visual words. Then, we propose a bilinear model to establish the relationships among an image concept and the two-level visual elements—that is, the concept-to-components and component-to-visual-words relationships. Although this component is an extension of the spatial pyramid in previous work,[3] our method does not use a fixed layout and is thus more flexible. In addition, we impose l^1 constraints on the bilinear model's parameters to solve the joint selection of visual words and components in the image representation, yielding a sparsity-constrained bilinear model (SBLM). The discriminative power of one SBLM is still limited, however, and it is sometimes hard to determine a good set of model parameters because of the SBLM's nonconvexity. To solve this problem, we combine a set of bilinear models along with sparsity constraints in a boosting-like procedure. To determine the model parameters during each boosting round, we propose a modified weighted feature-sign-search algorithm by alternatively optimizing over the two parameter subsets corresponding to the two linear relationships. During the alternative optimizing

1070-986X/12/$31.00 © 2012 IEEE Published by the IEEE Computer Society

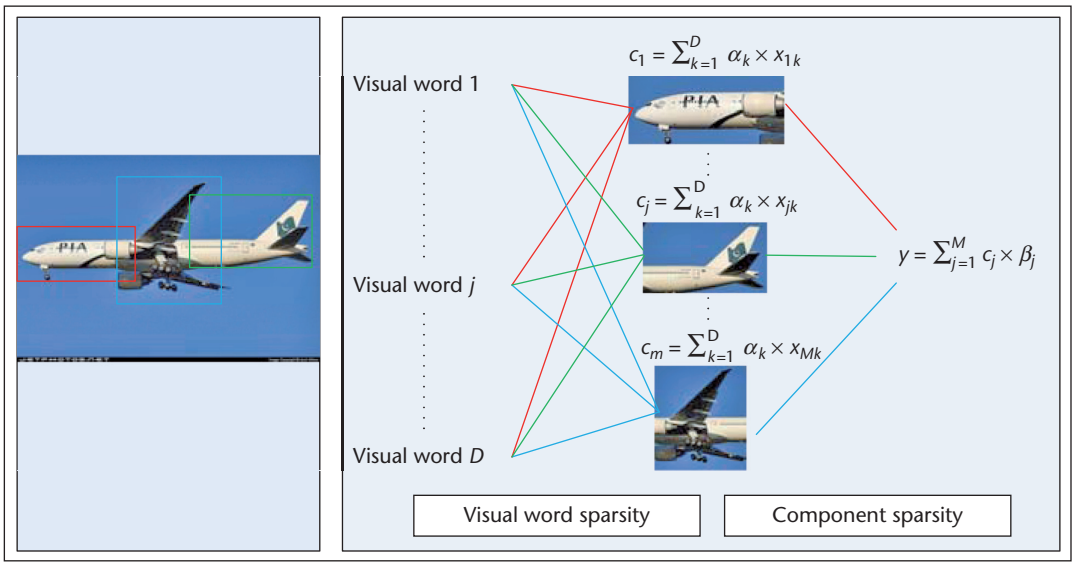

Figure 1. Proposed sparsity-constrained bilinear model. The SBLM models the relationship between two-level visual features and a given image concept.

process, our algorithm can keep refining the two parameter sets by strictly reducing the objective cost.

Boosting SBLM for Object Recognition

Figure 1 illustrates our proposed SBLM for modeling the relationship between the two-level visual elements and an image concept. After presenting the SBLM, we show how to combine a set of SBLMs in a boosting procedure to better recognize images.

Component-Based Image Representation

As we mentioned earlier, a component is a set of image regions that can be generated using various methods, such as sampling, segmentation, or detection. We extract rectangle regions as our components (see Figure 1). For each image, we densely extract overlapping components. The number of overlapped pixels varies depending on each image's size to make sure the sampled components cover the whole image. For each component, we use the frequency distribution of visual words within each component as its feature representation. Hence, a component is a higher-level representation and more descriptive than a single visual word because it combines the spatial correlations among nearby visual words. We use the histogram-based representation because it is invariant to rotation and efficient to compute; however, other more descriptive representation methods (such as graphs) can also be applied.

The visual representations based on the two-level elements are as follows. Formally, let $x_j^n \in R^D$ be the jth component of the nth image, where D is the visual vocabulary size. The kth element x_{jk}^n of x_j^n is the number of occurrences of visual word k within the jth component of the nth image. Components are arranged according to their indexes. We use a matrix $X^n = [x_1^n, x_2^n, ..., x_M^n] \in R^{D \times M}$ to denote the nth image, where M is the number of components. If we regard the whole image as one component, this model will degenerate to the standard BoW representation. Thus, we can view the BoW model as a special case of our model.

SBLM Formulation

To take advantage of the component-based image representation, we propose a novel visual word and component bilinear model. For one image, we believe that each component within the image has a confidence value of the image category; we use a linear combination of these confidence values to model the relationship. Let c_j^n be the confidence values of the jth component for the nth image. To predict the image category, our aim is to determine a component-level linear function:

$$\hat{y}_n = \sum_{j=1}^{M} c_j^n \times \beta^j \quad (1)$$

where $\hat{y}_n$ is the predicted label of the nth image, β^j is the parameter for the jth component, and M is the number of components.

Related Work in Object Recognition

The bag-of-visual-words model has been widely used in object recognition and image/video retrieval because of its simplicity and good performance. However, because a histogram-based image representation discards the spatial information of visual words, its descriptive power for image content is severely limited. Much research has sought to improve performance by using the spatial layout information and correlations of visual words. Approaches using geometric correspondence search achieved robustness at a high computational cost.[1] Attempts to use loose spatial information have shown great promise. Kristen Grauman and Trevor Darrell used a multiresolution histogram pyramid in the feature space to implicitly form a feature matching.[2] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce proposed a spatial-pyramid-matching method for natural scene recognition.[3] Li Fei-Fei, Rob Fergus, and Pietro Perona developed a generative visual model using an incremental Bayesian approach.[4] Gang Wang, Ye Zhang, and Li Fei-Fei used dependent regions to categorize objects in a generative framework.[5] Random forests and ferns were used by Anna Bosch, Andrew Zisserman, and Xavier Munoz for image classification.[6] Hao Zhang and his colleagues proposed a support vector machine and *k*-nearest neighbor (SVM-KNN) method that combined the advantages of both approaches.[7]

To impose topological constraints, Bosch, Zisserman, and Munoz used a spatial-pyramid kernel to represent shape.[8] Manik Varma and Debajyoti Ray leveraged learning methods to determine the discriminative power-invariance trade-off of multiple features.[9] Frank Moosmann, Bill Triggs, and Frederic Jurie proposed using randomized clustering forests to classify images that are robust to background clutter and fast to train and test.[10] Lui Yang and his colleagues combined discriminative-visual-words learning and classifier training into a unified framework.[11] Oren Boiman, Eli Shechtman, and Michal Irani directly used local features for image classification by using nearest-neighbor information.[12]

Not all visual words are useful for recognition for every object-recognition task. John Wright and his colleagues showed that the human visual system employs an effective attention mechanism and can focus on the interesting parts in an image to recognize different object categories robustly. In the past few years, researchers have shown that minimization is effective for object recognition.[13] The use of sparsity constraints makes these algorithms robust to noise and able to select the most useful visual words to help correctly categorize images. The success of sparsity representation lies in the assumption that although images are of high dimensionality, in many cases, images of the same class often exhibit degenerate structure.[13]

In the computational-learning theory literature, Yoav Freund and Robert Schapire proposed boosting,[14] which has since received much attention. The boosting procedure is a way of combining the performance of many weak classifiers to produce a powerful ''committee.'' Jerome Friedman, Trevor Hastie, and Robert Tibshirani interpreted boosting as a gradient descent in function space and proposed many specific algorithms using different loss functions.[15] Recently, more and more researchers have been adapting the boosting principle to efficiently and effectively combine classifiers for visual applications.

The confidence values vary from one component to the other, depending on the visual words within each component. We adopt a linear model of visual words to measure each component's confidence value. Let α^k be the parameter for visual word k. Then, we can write the visual-word-level linear model as

$$c_j^n = \sum_{k=1}^{D} \alpha^k \times x_{jk}^n \tag{2}$$

Let $\alpha = [\alpha^1, \alpha^2, \ldots, \alpha^D]^T$ and $\beta = [\beta^1, \beta^2, \ldots, \beta^M]^T$. We can rewrite Equations 1 and 2 in a unified form:

$$f(X^n) = \hat{y}_n = \alpha^T X^n \beta \tag{3}$$

which is a bilinear model. Given a fixed α or β, the model is linear with respect to β or α. The power of the bilinear model stems from the rich nonlinear interactions that can be represented by varying both α and β simultaneously.

Suppose we have a training image set with labels $\{(X^1, y^1), (X^2, y^2), \ldots, (X^N, y^N)\}$, where N is the number of training images and $y^n = \{-1, 1\}$ is the label of the nth image, $n \in \{1, 2, \ldots, N\}$. We can try to determine α and β by minimizing the summed loss between the predicted labels and the ground truth. The two sets of parameters can be learned by solving the following optimization problem:

$$[\alpha, \beta] = \arg\min_{\alpha,\beta} \sum_{n=1}^{N} L(y^n, \alpha^T X^n \beta) \tag{4}$$

where $L(\cdot,\cdot)$ is the loss function. Here we choose the exponential loss because it is differentiable and efficient to implement. The exponential loss has the following form:

$$L(y, \alpha^T X^n \beta) = \exp(-y \times \alpha^T X^n \beta) \tag{5}$$

We can determine α and β by solving Equation 4. However, when the number of

References

1. A. Berg, T. Berg, and J. Malik, ''Shape Matching and Object Recognition Using Low Distortion Correspondences,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 05), IEEE CS Press, vol. 1, 2005, pp. 26–33.
2. K. Grauman and T. Darrell, ''The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features,'' *Proc. 10th Int'l Conf. Computer Vision* (ICCV 05), IEEE CS Press, 2005, pp. 1458–1465.
3. S. Lazebnik, C. Schmid, and J. Ponce, ''Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 06), IEEE CS Press, 2006, pp. 2169–2178.
4. L. Fei-Fei, R. Fergus, and P. Perona, ''Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 04) *Workshop on Generative Model Based Vision,* IEEE CS Press, 2004, pp 178–186.
5. G. Wang, Y. Zhang, and L. Fei-Fei, ''Using Dependent Regions for Object Categorization in a Generative Framework,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 06), IEEE CS Press, 2006, pp. 1597–1604.
6. A. Bosch, A. Zisserman, and X. Munoz, ''Image Classification Using Random Forests and Ferns,'' *Proc. IEEE 11th Int'l Conf. Computer Vision* (ICCV 07), IEEE CS Press, 2007, pp. 1–8.
7. H. Zhang et al., ''SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 06), IEEE CS Press, 2006, pp. 2126–2136.
8. A. Bosch, A. Zisserman, and X. Munoz, ''Representing Shape with a Spatial Pyramid Kernel,'' *Proc. 6th ACM Int'l Conf. Image and Video Retrieval* (CIVR 07), ACM Press, 2007, pp. 401–408.
9. M. Varma and D. Ray, ''Learning the Discriminative Power-Invariance Trade-off,'' *Proc. IEEE 11th Int'l Conf. Computer Vision* (ICCV 07), IEEE CS Press, 2007, pp. 1–8.
10. F. Moosmann, B. Triggs, and F. Jurie, ''Fast Discriminative Visual Codebooks Using Randomized Clustering Forests,'' *Proc. 20th Ann. Conf. Neural Information Processing Systems* (NIPS 06), *Advances in Neural Information Processing Systems 19,* MIT Press, 2006, pp. 985–992.
11. L. Yang et al., ''Unifying Discriminative Visual Codebook Generation with Classifier Training for Object Category Recognition,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 08), IEEE CS Press, 2008, pp. 1–8.
12. O. Boiman, E. Shechtman, and M. Irani, ''In Defense of Nearest-Neighbor Based Image Classification,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 08), IEEE CS Press, 2008, pp. 1–8.
13. J. Wright et al., ''Robust Face Recognition via Sparse Representation,'' *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 2, 2009, pp. 210–227.
14. Y. Freund and R. Schapire, ''A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting,'' *J. Computer and System Sciences,* vol. 55, no. 1, 1997, pp. 119–139.
15. J. Friedman, T. Hastie, and R. Tibshirani, ''Additive Logistic Regression: A Statistical View of Boosting,'' *The Annals of Statistics,* vol. 28, no. 2, 2000, pp. 337–407.

training images is relatively small, there will be many possible solutions due to the under-constrained nature of the problem. Furthermore, there inevitably exists redundancy and varying usefulness among all the visual words and components. To jointly choose the most discriminative visual words and components, it is reasonable to seek the sparse solution by solving the following optimization problem:

$$[\alpha,\beta] = \arg\min_{\alpha,\beta} \sum_{n=1}^{N} \exp(-y^n \times \alpha^T X^n \beta) + c_1\|\alpha\|_0 + c_1\|\beta\|_0 \quad (6)$$

where $\|\cdot\|_0$ denotes the l^0 norm. It imposes sparsity constraints on both the visual words and components. c_1 and c_2 are penalty parameters that control the sparsity of α and β, respectively. However, this problem is hard to solve. Instead, we simplify Equation 6 to solve an easier problem:

$$[\alpha,\beta] = \arg\min_{\alpha,\beta} \sum_{n=1}^{N} \exp(-y^n \times \alpha^T X^n \beta) + c_1\|\alpha\|_1 + c_1\|\beta\|_1 \quad (7)$$

where $\|\cdot\|_1$ denotes the l^1 norm. After determining the α and β parameter sets, we can predict the image categories using Equation 3.

Boosting SBLM

We can determine the SBLM parameters to predict the image categories, but the discriminative power of one SBLM is limited. In addition, the SBLM is nonconvex, making it difficult to establish a proper set of parameters. The boosting principle can efficiently and

1. Start with weights $w^i = 1/N$, $i = 1, 2, \ldots, N$. The penalty parameters are c_1 and c_2.
2. Repeat for $t = 1, 2, \ldots, T$:
   A. Determine the parameters (α_t, β_t) of the tth SBLM by alternatively optimizing α_t or β_t while keeping β_t or α_t fixed with weights w^i, $i = 1, 2, \ldots, N$.
   B. Set $f_t(X) = \alpha_t^T X \beta_t$.
   C. Set $w^i \leftarrow w^i \exp[-y^i f_t(X_t)]$, $i = 1, 2, \ldots, N$, and renormalize so that $\sum_{i=1}^{N} w^i = 1$.
3. Output the classifier sign $[\sum_{t=1}^{T} f_t(X)]$.

Figure 2. Algorithm 1. Boosting SBLM for object recognition.

effectively combine a set of weak classifiers to produce a powerful "committee." Similarly, it is more efficient and discriminative to combine a set of bilinear models:

$$F_T(X) = \sum_{t=1}^{T} f_t(X) \tag{8}$$

where

$$f_t(X) = \alpha_t^T X \beta_t \tag{9}$$

By adding sparsity constrains to the parameters of each $f_t(\cdot)$ for jointly visual word and component selection, we can determine $F_T(\cdot)$ by solving this optimization problem:

$$F_T(X) = \arg\min_{F_T(X)} \sum_{n=1}^{N} \exp[-y^n \times F_T(X^n)] + c_1 \sum_{t=1}^{T} \|\alpha_t\|_1 + c_2 \sum_{t=1}^{T} \|\beta_t\|_1 \tag{10}$$

where c_1 and c_2 are penalty parameters that control the sparsity of α_t and β_t, respectively, and $t \in \{1, 2, \ldots, T\}$.

Establishing $F_T(X)$ by solving Equation 10 is equal to finding a set of parameters $(\alpha_t, \beta_t)_{t \in \{1,2,\ldots,T\}}$. Because it is hard to find all the parameters simultaneously, we use a greedy forward-stepwise approach[14] to learn one SBLM at a time. For $t = 1, 2, \ldots, T$, where $\{\alpha_p, \beta_p\}_{p=1}^{t-1}$ are fixed at their corresponding solution values at earlier iterations, the classification loss in Equation 5 can be rewritten as

$$\begin{aligned} &\exp[-y \times F_t(X)] \\ &= \exp[-y \times F_{t-1}(X)] \times \exp[-y \times f_t(X)] \\ &= w(X, y) \times \exp[-y \times f_t(X)] \end{aligned} \tag{11}$$

where $w(X, y) = \exp[-y \times F_{t-1}(X)]$ is the weighting parameter. We calculate it on the basis of the learned classifiers at earlier iterations; α_t and β_t can then be found by solving the following optimization problem:

$$\begin{aligned} (\alpha_t, \beta_t) = \arg\min_{\alpha_t, \beta_t} &\sum_{n=1}^{N} w(X^n, y^n) \\ &\times \exp[-y^n \times \alpha_t^T X^n \beta_t)] \\ &+ c_1 \|\alpha_t\|_1 + c_2 \|\beta_t\|_1 \end{aligned} \tag{12}$$

Our algorithm performs in a similar way to Adaboost[14] by training the next SBLM with weighted training samples, giving higher weights to cases that are currently misclassified. This is done for a sequence of weighted samples, and then the final classifier is defined as a linear combination of the SBLM from each stage. Algorithm 1 in Figure 2 gives a sketch description of the proposed boosting SBLM.

Alternative Optimization of Weighted SBLMs

Equation 12 is convex in α_t (with β_t fixed) and β_t (with α_t fixed), but not convex in both simultaneously. Thus, we use an alternative optimization algorithm to solve this problem. We try to find the optimal solution to α_t while keeping β_t fixed. Then we find the optimal solution to β_t while keeping α_t fixed. This process is iterated until either the reduced loss is below a threshold or the iteration process reaches a predefined number of steps.

When solving the optimization problem in Equation 12 over α_t while keeping β_t fixed, the third term in Equation 12 is constant and has no influence on the final result. Because β_t is fixed, we can ignore the third constant term. Equation 12 then equals

$$\begin{aligned} \alpha_t = \arg\min_{\alpha_t} &\sum_{n=1}^{N} w(X^n, y^n) \\ &\exp(-y^n \times \alpha_t^T X^n \beta_t) + c_1 \|\alpha_t\|_1 \end{aligned} \tag{13}$$

We adopt the same principle as Honglak Lee and his colleagues[15] to solve Equation 13. Although the proposed Algorithm 2 in Figure 3 looks like the algorithm in their work, the two are fundamentally different. First, we use a different loss function for different applications. They adopt the least-squares loss for reconstruction, whereas our algorithm uses the summed exponential loss with weights for classification. Second, the feature-sign step in their work is an unconstrained quadratic optimization problem, and an analytical solution to the problem exists. However, although the objective function is

still convex in our algorithm, there is no analytical solution and we have to search for the optimal solution by gradient descent.

As long as we know the sign of each $.\alpha_t^i.$ at the optimal value, then $\|\alpha_t\|_1$ can be replaced with either α_t^i (if $\alpha_t^i \geq 0$) or $-\alpha_t^i$ (if $\alpha_t^i < 0$). If we only consider the nonzero parameters of α_t, Equation 13 is reduced to an unconstrained convex optimization problem, which can be solved efficiently. Therefore, we try to guess the optimal sign of the parameter α_t^i. Given such a guess, we will be able to solve the resulting convex optimization problem. Furthermore, the algorithm systematically refines the guess if it is initialized with incorrect values.

Algorithm 2 shows the details of the weighted feature-sign search algorithm for α according to

$$\frac{\partial L(y, \alpha_t{}^T X \beta_t)}{\partial \alpha_t} = -y \times X\beta_t \times \exp(-y \times \alpha_t{}^T X \beta_t) \tag{14}$$

For simplicity, we omit the indices of t in Algorithm 2.

Algorithm 2 systematically searches for the optimal active set and parameter sign by proceeding in a series of feature-sign steps. During each step, given a current guess of the active set and the parameter signs, a better solution $\hat{\alpha}_{\text{new}}$ to the convex optimization problem is obtained by gradient descent. Then the active set and the parameter signs are updated using an efficient discrete line search between the current solution and $\hat{\alpha}_{\text{new}}$. As we show later on, each step will strictly reduce the objective cost.

When we try to solve the optimization problem in Equation 12 over β_t while keeping α_t fixed, the second term in Equation 12 is constant and has no influence on the final result. If we ignore the second constant term, Equation 12 becomes

$$\beta_t = \arg\min_{\beta_t} \sum_{n=1}^{N} w(X^n, y^n) \times \exp(-y^n \times \alpha_t^T X^n \beta_t) + c_2 \|\beta_t\|_1 \tag{15}$$

We can solve this problem using the same procedure as Algorithm 2 by replacing α_t and c_1 with β_t and c_2, respectively:

$$\frac{\partial L(y, \alpha_t{}^T X \beta_t)}{\partial \beta_t} = -y \times \alpha_t^T X \times \exp(-y \times \alpha_t{}^T X \beta_t) \tag{16}$$

1. Initialize $\alpha := \vec{0}, \theta := \vec{0}$, and active set ={ }, where $\theta^j \in \{-1, 0, 1\}$ denotes sign(α^j).
2. From zero parameters of α, select $j = \arg\max_j \left| \sum_{n=1}^{N} w^n \times \frac{\partial L(y^n, \alpha^T X^n \beta)}{\partial \alpha^j} \right|$
   Activate α^j (add j to the active set) only if it can locally improve the objective:
   If $\sum_{n=1}^{N} w^n \times \frac{\partial L(y^n, \alpha^T X^n \beta)}{\partial \alpha^j} > c_1$, then set $\theta^j := -1$, active set := $\{j\} \cup$ active set.
   If $\sum_{n=1}^{N} w^n \times \frac{\partial L(y^n, \alpha^T X^n \beta)}{\partial \alpha^j} < -c_1$, then set $\theta^j := 1$, active set := $\{j\} \cup$ active set.
3. Feature-sign step:
   For each training image n
   Let $\hat{X}^n$ be a submatrix of X^n that contains only the rows corresponding to the active set.
   Let $\hat{\alpha}$ and $\hat{\theta}$ be subvectors of α and θ corresponding to the active set.
   Compute the solution $\hat{\alpha}_{\text{new}}$ to the optimization problem by gradient descent
   $$\arg\min_{\hat{\alpha}} \sum_{n=1}^{N} w^n \times L(y^n, \hat{\alpha}^T \hat{X}^n \beta) + c_1 \left\| \hat{\theta}^T . \times \hat{\alpha} \right\|_1$$
   Perform a discrete line search on the closed line segment from $\hat{\alpha}$ to $\hat{\alpha}_{\text{new}}$:
   Check the object value at $\hat{\alpha}_{\text{new}}$ and all points where any coefficient changes sign.
   Update $\hat{\alpha}$ (and the corresponding entries in α) to the point with the lowest objective value.
   Remove zero parameters of $\hat{\alpha}$ from theactive set and update $\theta := \text{sign}(\alpha)$.
4. Check the optimality conditions:
   A. Optimality condition for nonzero parameters:
   $$\sum_{n=1}^{N} w^n \times \frac{\partial L(y^n, \alpha^T X^n \beta)}{\partial \alpha^j} + c_1 \times \text{sign}(\alpha^j) = 0, \forall \alpha^j \neq 0.$$
   If condition A is not satisfied, go to step 3 (without any new activation). Otherwise, check condition B.
   B. Optimality condition for zero parameters:
   $$\left| \sum_{n=1}^{N} w^n \times \frac{\partial L(y^n, \alpha^T X^n \beta)}{\partial \alpha^j} \right| \leq c_1, \forall \alpha^j = 0.$$
   If condition B is not satisfied, go to step 2; otherwise return α as the solution.

Figure 3. Algorithm 2. Weighted feature-sign search algorithm for α.

Theoretical Foundations

To simplify notation, we omit the indices of t and let

$$g(\alpha) = \sum_{n=1}^{N} w^n \exp(-y^n \times \alpha^T X^n \beta) + c_1 \|\alpha\|_1$$

Let a parameter vector α be regarded as consistent with a given active set and sign vector θ if the following two conditions hold for all j:

- If j is in the active set, then $\text{sign}(\alpha^j) = \theta^j$.
- If j is not in the active set, then $\text{sign}(\alpha^j) = -\theta^j$.

Lemma 1. Consider the optimization problem in Equation 13 augmented with the additional constraint that α is consistent with a given active set and sign vector. Then, if the current parameters α_{old} are consistent with the active set and sign vector but are not optimal for the augmented problem at the start of step 3, the weighted searching for the feature-sign

step for α is guaranteed to strictly reduce the objective cost.

Proof. Let $\hat{\alpha}_{\text{old}}$ be the subvector of α_{old} corresponding to coefficients in the given active set. In step 3 of Algorithm 2, we minimize the following function:

$$\hat{g}(\hat{\alpha}) = \sum_{n=1}^{N} w^n \exp(-y^n \times \hat{\alpha}^T \hat{X}^n \beta) + c_1 \hat{\theta}^T \hat{\alpha}$$

Because $\hat{\alpha}_{\text{old}}$ is not an optimal point, we have $\hat{g}(\hat{\alpha}_{\text{new}}) < \hat{g}(\hat{\alpha}_{\text{old}})$. Now we consider two possible cases. First, if $\hat{\alpha}_{\text{new}}$ is consistent with the given active set and sign vector, updating $\hat{\alpha}_{\text{old}}$ to $\hat{\alpha}_{\text{new}}$ will strictly decrease the objective. Second, if $\hat{\alpha}_{\text{new}}$ is not consistent with the given active set and sign vector, let $\hat{\alpha}_d$ be the first crossing zero point (where any coefficient of $\hat{\alpha}$ changes sign) on a line segment from $\hat{\alpha}_{\text{old}}$ to $\hat{\alpha}_{\text{new}}$. Then, of course, $\hat{\alpha}_{\text{old}} \neq \hat{\alpha}_{\text{new}}$, and $\hat{g}(\hat{\alpha}_d) < \hat{g}(\hat{\alpha}_{\text{old}})$. Because $g(\alpha)$ is convex, we have $g(\hat{\alpha}_d) = \hat{g}(\hat{\alpha}_d) < g(\hat{\alpha}_{\text{old}}) = g(\hat{\alpha}_{\text{old}})$. Therefore, the discrete line search described in step 3 of Algorithm 2 ensures a decrease in the objective cost.

Lemma 2. Consider the optimization problem in Equation 13 augmented with the additional constraint that α is consistent with a given active set and sign vector. If the coefficients α_d at the start of step 2 are optimal for the augmented problem but are not optimal for Equation 13, the weighted feature-sign step for α is guaranteed to strictly reduce the objective cost.

Proof. Because α_d is optimal for the augmented problem, it satisfies optimality condition A, but not condition B. Thus, in step 2 of Algorithm 2, there is some j, such that

$$\left| \sum_{n=1}^{N} w^n \times \frac{\partial L(y^n, \alpha^T X^n \beta)}{\partial \alpha^j} \right| > c_1$$

This jth parameter is activated, and j is added to the active set. In step 3, because a Taylor expansion of the convex objective function $\hat{g}(\hat{\alpha})$ around $\hat{\alpha} = \hat{\alpha}_d$ has a first-order term in α^j only (using condition 4A for the other parameters), any direction that locally decreases $\hat{g}(\hat{\alpha})$ must be consistent with the sign of the activated α^j. In addition, because α_d is not an optimal point of the objective function $\hat{g}(\hat{\alpha})$, the objective function must decrease locally near $\hat{\alpha}_d$ along the direction from $\hat{\alpha}_d$ to $\hat{\alpha}_{\text{new}}$. Hence, the line-search direction $\hat{\alpha}_d$ to $\hat{\alpha}_{\text{new}}$ must be consistent with the sign of the active set α^j. Finally, because $\hat{g}(\hat{\alpha}) = g(\hat{\alpha})$, when $\hat{\alpha}$ is consistent, either $\hat{\alpha}_{\text{new}}$ is consistent or the first zero-crossing from $\hat{\alpha}_d$ to $\hat{\alpha}_{\text{new}}$ has a lower objective cost.

Based on Lemmas 1 and 2, we can now prove the convergence of Algorithm 2.

Theorem 1. The weighted feature-sign search algorithm for α converges to a global optimum of the optimization problem in Equation 13 in a finite number of steps.

Proof sketch. From Lemmas 1 and 2, it follows that the weighted searching for the feature-sign step in Algorithm 2 always strictly reduces $g(\alpha)$. At the start of step 2, α either satisfies the optimality condition 4A or is $\vec{0}$. In either cases, α is consistent with the current active set and sign vector and must be optimal for the augmented problem described in the lemmas. Because the number of all possible active sets and parameter signs is finite, and because no pair can be repeated, the outer loop in steps 2 through 4 cannot repeat indefinitely. Thus, a finite number of steps are needed to reach step 4B from step 2. This is true because the inner loop of steps 3 and 4A always results in either an exit to step 4B or a decrease in the size of the active set.

Theorem 2. Step 2A in Algorithm 1 alternatively refines the two parameter sets by strictly reducing the objective cost of Equation 12 and is guaranteed to stop in a finite number of steps.

Proof sketch. From Lemmas 1 and 2 and Theorem 1, it follows that the weighted feature-sign search algorithm for α and the weighted feature-sign search algorithm for β alternatively reduce the objective cost. The number of all possible active sets and coefficient signs are finite for both of the two subproblems, and no pair can be repeated (since the objective cost is strictly decreasing). Hence, step 2A in Algorithm 1 can stop in a finite number of steps.

Experiments

We evaluated the proposed boosting SBLM method on three diverse datasets: the 15 natural

scene (Scene-15) dataset,[3] the Caltech-101 dataset,[4] and the Caltech-256 dataset.[16] The 4,485 images in the Scene-15 dataset consist of categories that range from natural scenes (such as forests and mountains) to man-made environments (like offices and kitchens). The Caltech-101 dataset contains 8,677 images from 101 object categories, where the number of images in each category varies from 31 to 800. The Caltech-256 dataset holds 29,780 images of 256 classes, where each class has at least 80 images.

We performed all processing on grayscale images. For feature extraction, we followed the setup in previous work that has been proven effective on these datasets,[3,17] and we densely computed SIFT descriptors on overlapping 16 × 16 pixels with an overlap of 8 pixels. The components are densely extracted with the size of 64 × 64 pixels. We extracted $M \times N$ components (20 × 15 for the Scene-15 dataset and the Caltech-101 dataset, and 10 × 10 for the Caltech-256 dataset) with overlap for each image. The number of overlapped pixels varied, depending on each image's size, to make sure the sampled components covered the whole image. We empirically set the penalty parameters of both c_1 and c_2 to 1,000. Multiclass classification was done via the one-versus-all rule. That is, a boosting SBLM separates each class from the rest, and a test image is assigned the label of the classifier with the highest response. We measured the classification performance quantitatively by the average of per-class classification rates.

Experiment 1: Scene-15 Dataset

The major picture sources in the Scene-15 dataset include the Corel collection, personal photographs, and Google image search. Each category has 200 to 400 images, and the average image size is 300 × 250 pixels. We used the same number of training images per category as previous research.[3,17] We then randomly chose 100 images per category as the training set and used the remaining images as the test set. We repeated this process five times, creating a codebook with 1,000 clusters by k-means clustering.

Table 1 shows the performance comparison among the boosting SBLM (B-SBLM), the one without sparsity constraints by using L^2 norm (instead of L^1 norm) in B-L^2BLM, and the previous research methods.[3,17] We also show the performance of a single SBLM to demonstrate the effectiveness of B-SBLM by combining a set of SBLMs in a boosting procedure. B-SBLM produced the best results.

Table 1. Performance comparison of different methods and the proposed boosting SBLM on the Scene-15 dataset (100 training images per class).

Method	Performance
Lazebnik[3]	81.4 ± 0.5
Gemert[17]	76.3 ± 0.4
L^2BLM	83.4 ± 1.3
SBLM	85.6 ± 1.5
B-L^2BLM	86.8 ± 1.5
B-SBLM	90.5 ± 1.6

The component-based solutions (L^2BLM, SBLM, B-L^2BLM, and B-SBLM) performed better than the BoW methods.[3,17] This is because the component-based image representation preserves more spatial relationships and correlations of visual words. In addition, the methods with sparse constraints (such as L^1 norm) are better than the ones using L^2 norm—that is, SBLM versus L^2BLM, and B-SBLM versus B-L^2BLM. For a specified object-recognition task, SBLM and B-SBLM can jointly choose the most discriminative visual words and components by imposing sparsity constraints. Lastly, by establishing a set of SBLM models using a boosting-based reweighting scheme, B-SBLM can further improve the performance of a single SBLM.

Experiment 2: Caltech Datasets

We conducted the second set of experiments on the Caltech-101 and Caltech-256 datasets. The Caltech-101 dataset has 101 classes with high intraclass appearance and shape variability. The number of images in each category varies from 31 to 800 images. Most of these images are approximately 300 × 300 pixels. The Caltech-256 dataset is diverse and challenging, with 256 classes of 29,780 images. Each class contains at least 80 images, and each image is not manually rotated to face one direction. We created a codebook with 1,000 clusters by k-means clustering for the two datasets, respectively.

We randomly split the images into training and test samples five times and present the average performance in our evaluations. Figure 4 compares the proposed boosting SBLM with

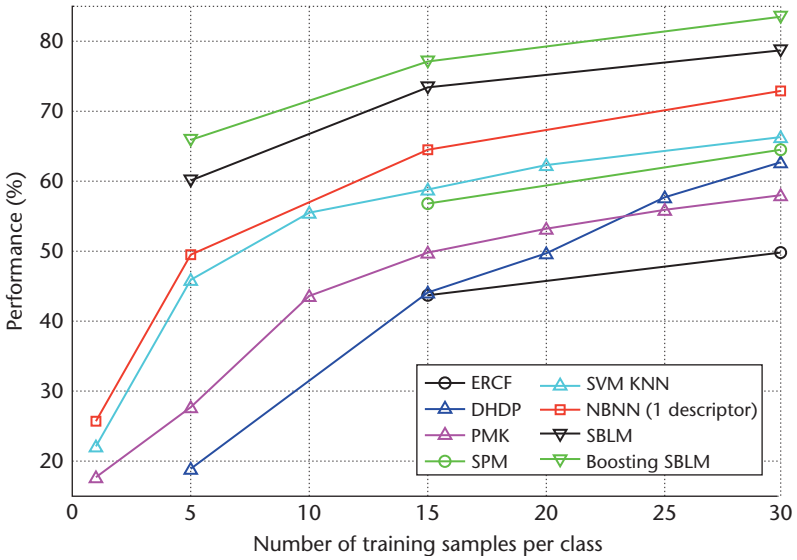

Figure 4. Performance comparison on the Caltech-101 dataset. We compared the proposed boosting SBLM with pyramid match kernel (PMK),[2] spatial-pyramid match (SPM) kernel,[3] Dependent Hierarchical Dirichlet Process (DHDP),[5] support vector machine and k-nearest neighbor (SVM-KNN),[7] extremely random clustering forest (ERCF),[10] nearest-neighbor classifier (NBNN) with one descriptor,[12] and single SBLM approaches.

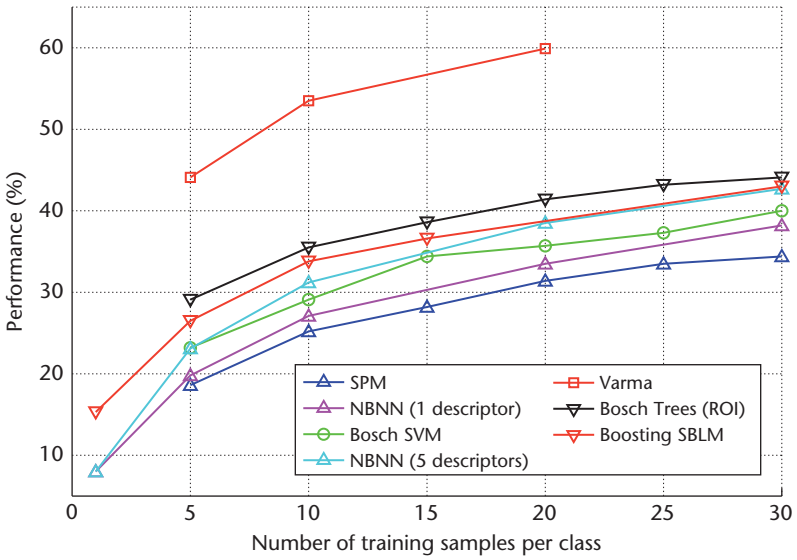

Figure 5. Performance comparison on the Caltech-256 dataset. We compared the proposed boosting SBLM with the Bosch SVM,[6] Bosch Trees (ROI),[8] Varma,[9] NBNN with one descriptor,[12] NBNN with five descriptors,[12] and SPM approaches.[16]

other related methods on the Caltech-101 dataset. The PMK leveraged the pyramid match kernel in a feature space,[2] while SPM used the spatial pyramid kernel.[3] The Dependent Hierarchical Dirichlet Process (DHDP) used dependent regions in a generative framework to classify images.[5] The SVM-KNN leveraged a hybrid support-vector-machine- and k-nearest-neighbor-based method,[7] which was considered state of the art until recently. The ERCF used the extremely random clustering forest to generate discriminative visual words.[10] The NBNN leveraged a nearest-neighbor classifier on local features directly,[12] which avoids the quantization loss of local features.

Figure 4 shows that the proposed B-SBLM outperforms many of the previously reported methods, even the state-of-the-art work that combines the benefits of NN-based and SVM-based methods.[7] Although several methods used the BoW representation of images,[2,3,5,7] the method by Oren Boiman, Eli Shechtman, and Michal Irani[12] used the NN relationship of unquantized descriptors between images and classes. The performance improvement over these methods demonstrates the effectiveness of our component-based image representation.

The ERCF work[10] leveraged the random-clustering forest to generate discriminative visual words for classification. The performance of ERCF with 30 training images per class was 49.8 percent. This is better than simply using k-means clustering, which had a 41.2 percent performance.[3] However, ERCF did not consider the spatial information of local features, and our method jointly combines the spatial information with visual words and component selection and, hence, outperforms the ERCF.

For class-level results, the proposed SBLM and B-SBLM performed well on the categories that were either dominated by rotation artifacts (such as a minaret), had little clutter (for example, a revolver or umbrella), or represented coherent natural ''scenes'' (such as an okapi). The less successful classes were influenced by different rotation (such as a platypus) or textureless (such as a pigeon) animals. However, by jointly choosing the most discriminative visual words and components, and by determining a set of SBLM using a reweighting scheme, the proposed method performed better.

Figure 5 shows the performance comparison for the B-SBLM and some related methods[6,8,9,12,16] on the Caltech-256 dataset. Anna Bosch, Andrew Zisserman, and Xavier Munoz addressed the positional variability of objects in images using the region of interest (ROI).[6] They also represented shape with a spatial-pyramid kernel,[8] whereas Manik Varma and Debajyoti Ray tried to determine the optimal

descriptor for object recognition.[9] Gregory Griffin, Alex Holub, and Pietro Perona reimplemented the SPM on the Caltech-256 dataset.[16] Similar to the Caltech-101 dataset, we randomly split the images into training and test sets three times.

Figure 5 shows that our algorithm outperforms SPM and NBNN with one descriptor, when considering only one type of feature. The B-SBLM even outperforms the NBNN with five descriptors, which uses five types of descriptors by simple averaging. The ROI optimization[8] effectively addressed the positional variability of objects in images. Hence, it performed better than B-SBLM for a few of the training samples. This gap closes when the number of training samples increases, however.

Other methods[6,8,9,12] tried to leverage multiple types of features by establishing class-adaptive combinations of different features, whereas our algorithm uses a single feature type. Better performance is possible if multiple types of features are fused together rather than using a single feature type.

Conclusion

In this article, we used components to represent images and proposed a B-SBLM for efficient classification. Our experiments demonstrate the proposed method's effectiveness, and we believe considering spatial information and correlations of visual words is important for image classification. In future work, we will study how to speed up the proposed B-SBLM method. **MM**

Acknowledgments

This work was supported by the 973 Program (project 2010CB327905) and the National Natural Science Foundation of China (under grants 60903146, 60835002, and 90920303).

References

1. A. Berg, T. Berg, and J. Malik, ''Shape Matching and Object Recognition Using Low Distortion Correspondences,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 05), IEEE CS Press, vol. 1, 2005, pp. 26–33.
2. K. Grauman and T. Darrell, ''The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features,'' *Proc. 10th Int'l Conf. Computer Vision* (ICCV 05), IEEE CS Press, 2005, pp. 1458–1465.
3. S. Lazebnik, C. Schmid, and J. Ponce, ''Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 06), IEEE CS Press, 2006, pp. 2169–2178.
4. L. Fei-Fei, R. Fergus, and P. Perona, ''Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 04) *Workshop on Generative Model Based Vision,* IEEE CS Press, 2004, pp 178–186.
5. G. Wang, Y. Zhang, and L. Fei-Fei, ''Using Dependent Regions for Object Categorization in a Generative Framework,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 06), IEEE CS Press, 2006, pp. 1597–1604.
6. A. Bosch, A. Zisserman, and X. Munoz, ''Image Classification Using Random Forests and Ferns,'' *Proc. IEEE 11th Int'l Conf. Computer Vision* (ICCV 07), IEEE CS Press, 2007, pp. 1–8.
7. H. Zhang et al., ''SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 06), IEEE CS Press, 2006, pp. 2126–2136.
8. A. Bosch, A. Zisserman, and X. Munoz, ''Representing Shape with a Spatial Pyramid Kernel,'' *Proc. 6th ACM Int'l Conf. Image and Video Retrieval* (CIVR 07), ACM Press, 2007, pp. 401–408.
9. M. Varma and D. Ray, ''Learning the Discriminative Power-Invariance Trade-Off,'' *Proc. IEEE 11th Int'l Conf. Computer Vision* (ICCV 07), IEEE CS Press, 2007, pp. 1–8.
10. F. Moosmann, B. Triggs, and F. Jurie, ''Fast Discriminative Visual Codebooks Using Randomized Clustering Forests,'' *Proc. 20th Ann. Conf. Neural Information Processing Systems* (NIPS 06), *Advances in Neural Information Processing Systems 19,* MIT Press, 2006, pp. 985–992.
11. L. Yang et al., ''Unifying Discriminative Visual Codebook Generation with Classifier Training for Object Category Recognition,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 08), IEEE CS Press, 2008, pp. 1–8.
12. O. Boiman, E. Shechtman, and M. Irani, ''In Defense of Nearest-Neighbor Based Image Classification,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 08), IEEE CS Press, 2008, pp. 1–8.
13. J. Wright et al., ''Robust Face Recognition via Sparse Representation,'' *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 2, 2009, pp. 210–227.

14. J. Friedman, T. Hastie, and R. Tibshirani, ''Additive Logistic Regression: A Statistical View of Boosting,'' *The Annals of Statistics,* vol. 28, no. 2, 2000, pp. 337–407.
15. H. Lee et al., ''Efficient Sparse Coding Algorithms,'' *Advances in Neural Information Processing Systems* (NIPS 07), vol. 19, 2007, pp. 801–808.
16. G. Griffin, A. Holub, and P. Perona, ''Caltech-256 Object Category Dataset,'' tech. report, Calif. Inst. of Technology, 2007.
17. J. Gemert et al., ''Visual Word Ambiguity,'' *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 7, 2010, pp. 1271–1283.

Chunjie Zhang is a doctoral student in the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include machine learning, image content analysis, and object categorization. Zhang has a BE from Nanjing University of Posts and Telecommunications, China. Contact him at cjzhang@nlpr.ia.ac.cn.

Jing Liu (the corresponding author) is an associate professor in the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. Her research interests include machine learning, image content analysis, and multimedia information retrieval. Liu has a PhD in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences. Contact her at jliu@nlpr.ia.ac.cn.

Qi Tian is an associate professor in the Department of Computer Science at the University of Texas at San Antonio. His research interests include multimedia information retrieval, computational systems biology, biometrics, and computer vision. Tian has a PhD in electrical and computer engineering from the University of Illinois at Urbana-Champaign. Contact him at qitian@cs.utsa.edu.

Yanjun Han is a researcher at Douban. His research interests include machine learning, optimization, and social network analysis. Han has a PhD in complex systems from the Institute of Automation, Chinese Academy of Sciences. Contact him at yanjun.han@ia.ac.cn.

Hanqing Lu is a professor in the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include video analysis and multimedia technology and systems. Lu has a PhD in pattern recognition and intelligent systems from Huazhong University of Sciences and Technology. Contact him at luhq@nlpr.ia.ac.cn.

Songde Ma is a professor at the National Laboratory of Pattern Recognition, Chinese Academy of Sciences. His research interests include computer vision, multimedia analysis and intelligent systems. Ma has a PhD in computer vision from the University of Paris. Contact him at masd@most.cn.