Improving Recognition through Object Sub-categorization

Al Mansur and Yoshinori Kuno

Graduate School of Science and Engineering, Saitama University,
255 Shimo-Okubo, Sakura-ku, Saitama-shi, Saitama 338-8570, Japan
{mansur,kuno}@cv.ics.saitama-u.ac.jp

Abstract. We propose a method to improve the recognition rate of Bayesian classifiers by splitting the training data and using separate classifier to learn each sub-category. We use probabilistic Latent Semantic Analysis (pLSA) to split the training set automatically into sub-categories. This sub-categorization is based on the similarity of training images in terms of object's appearance or background content. In some cases, clear separation does not exist in the training set, and splitting results in worse performance. We compute the average difference between posteriors from the pLSA model, and observing this parameter, we can decide whether splitting is useful or not. This approach has been tested on eight object categories. Experimental results validate the benefit of splitting the training set.

1 Introduction

Object category recognition is a challenging research problem in the field of computer vision. To improve the recognition, several promising methods have been proposed. In [1], a bag-of-keypoints method with Naive Bayes and SVM is used to classify different object categories. Bag of keypoint with Bayesian approach has also been used in [2, 3] for learning in their algorithm.

Difficulty of class recognition is mainly due to the unpredictable appearance of a category and the presence of diverse background scenes. Most of the class recognition techniques [1, 2, 3] learn the object model from a set of training images covering the variation of the object appearance in the presence of various backgrounds. It is a common practice to utilize the whole training set of an object category to learn the object model. However, sometimes, such an approach may result in an inaccurate model of the object. Moreover, it is a huge task for a single classifier to construct an accurate decision boundary.

In most cases, clear separations exist between the appearances of the object in training sets. Motivated by this observation, instead of using the whole set of the training images to learn a single model on a single classifier, we attempt to split the training set and to use each subset to learn separate model of the same object. Our method is based on this sub-categorization. Instead of learning in a conventional framework, we divide the training set into different sub-categories, if possible.

G. Bebis et al. (Eds.): ISVC 2008, Part II, LNCS 5359, pp. 851–859, 2008.

Given large quantities of training data set it is very difficult to find out the similarities between the objects manually for the purpose of separating into sub-groups. Therefore, we apply the probabilistic Latent Semantic Analysis (pLSA) proposed by Hofmann et al. [4, 5] to discover the sub-categories. pLSA has been widely used for topic discovery in the field of statistical natural language processing. The object sub-categories obtained from the pLSA learning phase are used as the training images to final learning and recognition algorithms. We have used the Naive Bayesian framework for these purposes. We train Naive Bayesian classifiers separately on sub-categories. In the recognition stage, best decision is chosen using the posteriors associated with each decision. This significantly improves the recognition performance over conventional single classifier learnt on the whole training set.

2 Topic Discovery Using pLSA

pLSA is a generative model widely used in text analysis and in discovering topics in document using the bag-of-words document representation. Suppose we have a collection of N images $D = d_1, ..., d_N$ with words from a visual vocabulary $W = w_1, ..., w_M$. This collection of images is summarized in a M by N co-occurance table T, where $T_{ij} = n(w_i, d_j)$, and $n(w_i, d_j)$ denotes how often the word w_i occurred in an image d_j. In addition, there is also a latent variable model for co-occurrence data which associates an unobserved class variable $z \in Z = z_1, ..., z_K$ with each observation. A joint probability model $P(w, d)$ over $M \times N$ is defined by the mixture:

$$P(w|d) = \sum_{z \in Z} P(w|z)P(z|d) \tag{1}$$

$P(w|z)$ are the distributions of visual words over each topic and, each image is modeled as a mixture of topics, $P(z|d)$. Detail explanation of the model is given in [4, 5, 6].

2.1 Formation of Visual Words

We compute SIFT descriptors [8] at points on a regular grid with spacing of 5 and 10 pixels. At each point, SIFT descriptors are computed over circular regions with radii r=10 and 15 pixels. As a result, each point is represented by 2 SIFT descriptors , each is 128-dimensional vector. Multiple descriptors allow scale invariance between images. The SIFT descriptors are then vector quantized by k-means clustering to produce the visual 'words' for the vocabulary.

2.2 Splitting an Object Category

The technique of sub-categorization is shown visually in Figure 1. During the training step, topic specific distributions $P(w|z)$ are learnt from the set of training images containing an object category. Each training image is then represented

by a Z-vector $P(z|d_{train})$, where Z is the number of topics learnt. Determining both $P(w|z)$ and $P(z|d_{train})$ simply involves fitting the pLSA model to the entire set of training images. As we are interested to split an object category into two sub-categories, the number of topics is 2 in our application. An image, d_j is sub-categorized according to the maximum of $P(z_k|d_j)$ over number of topics, k. It is possible to split a category into more than two. However, in this case, combining decisions from several classifiers may become difficult.

In Figure 1, we illustrate the process of sub-categorization of object category 'rooster'. This category is taken from Caltech dataset [7] and there are 49 objects in this category. It is interesting that one sub-category contains the images of roosters with white or plain background and the other sub-category contains rooster with natural or other backgrounds.

After splitting an object category into two sub-categories, we train two separate Bayesian classifiers. The same negative training set is used in both classifiers (see Figure 2). During classification, a test image is given to both classifiers. If their decisions do not agree, we choose the decision with the higher posterior probability.

In some cases, splitting is not useful. If the $P(z_k|d_j)$ does not vary enough over k, image d_j has nearly similar probabilities to fall into all of the sub-categories. Splitting results in poor performance in such a case. Relying on this observation, we compute the average difference between $P(z_1|d_j)$ and $P(z_2|d_j)$over all images:

$$s = \frac{1}{N}\sum_{i=1}^{N} |P(z_1|d_i) - P(z_2|d_i)| \tag{2}$$

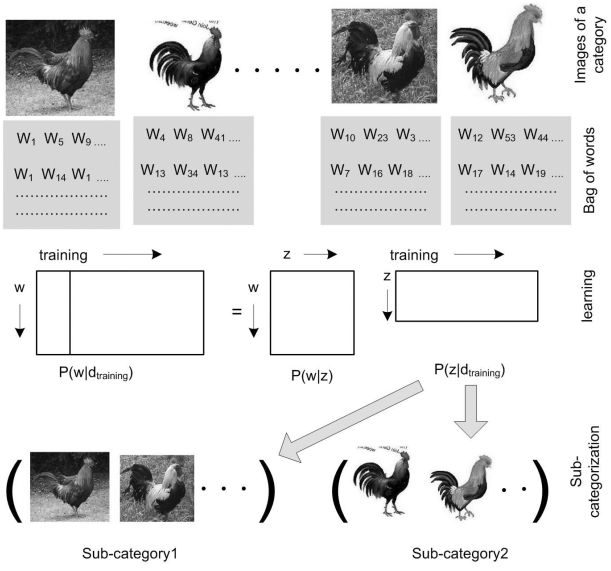

Fig. 1. Object sub-categorization process

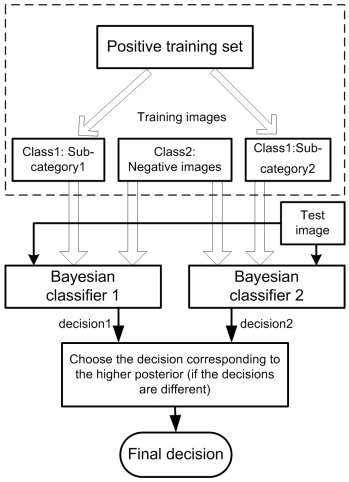

Fig. 2. Classifier for two sub-categories

By observing the value of s we can decide whether the sub-categorization is useful or not. If the value of s is low, we do not split the positive training set and train a single classifier with the whole data. Through experiments we verify that high value of s indicates that the sub-categorization will be useful. In contrast, if we sub-categorize a training set with low value of s, we may get worse results.

3 Classification Using Naive Bayes Classifier

For classification, it is possible to use non-linear classifiers (statistical or non-statistical) which are designed to take into account the class conditional distributions, e.g. SVM with nonlinear kernel. AdaBoost is another example of such a non-linear classifier. Additionally, boosting allows for coupling the feature extraction stage with the classification. However, we choose Naive Bayes framework which is a very simple linear classifier. The reason for this choice is to argue that the sub-categorization enables for very simple representations. In addition, in our proposed approach, although a set of features with proven efficiency is used in the clustering, for the classification, simple grayscale intensities are considered. The reason is same as that mentioned before (simplicity).

In our representation, a class represents the set of all possible appearances of a subcategory of an object category. Our aim is to classify a test image into the most likely class. Let $\mathbf{C} = \{c_1, c_2, ..., c_k\}$ be the set of K possible classes and $\mathbf{x} = \{x_1, x_2, ..., x_d\}$ is the set of continuous features extracted from a patch. Given a feature vector $\{x_1, x_2, ..., x_d\}$, our task is to estimate the most probable class such that

$$\hat{c_i} = \underset{c_i}{\operatorname{argmax}} \; P\left(C = c_i | x_1, x_2, ..., x_d\right) \tag{3}$$

Using Bayes' theorem, we write

$$P\left(C = c_i | x_1, .., x_d\right) = \frac{p\left(x_1, .., x_d | C = c_i\right) P\left(C = c_i\right)}{p\left(x_1, ..., x_d\right)} \tag{4}$$

If the prior P(C) is uniform, our problem is to find

$$\hat{c_i} = \underset{c_i}{\operatorname{argmax}}\ p\left(x_1, x_2, ..., x_d | C = c_i\right) \tag{5}$$

We consider grayscale intensities as the features. We resize each training image into 20×20 pixels. As a result, the length of a feature vector d is 400. Therefore, evaluation of joint probability in Eq. 5 is not feasible. Under the "naive" conditional independence assumption, the conditional distribution over the class variable C can be expressed as:

$$p\left(x_1, x_2, .., x_d | C = c_i\right) = \prod_{j=1}^{d} p\left(x_i | C = c_i\right) \tag{6}$$

However, in the real world, the independence assumption may not be true. In order to meet the independence assumption, we do PCA before applying the data to the Naive Bayes classifier. By decorrelating the features, PCA makes them statistically independent. PCA also reduces the dimension of the feature vectors by removing the irrelevant features.

4 Experiments

We present experimental results to investigate two areas: (i) object splitting - where categories are split into two sub-categories (ii) object detection - where we wish to determine the presence of an object in each image. We use two datasets of objects, one is Caltech [7] and the other from our own. Both datasets depict one object per image. From Caltech image data sets we select six categories. The categories are: faces, motorbikes, watches, sunflower, rooster and background. Background images are used as negative images and the other objects are used as positive images in our binary classifier. The reason for picking these particular categories is pragmatic: they have large number of images per category, some of them can be split into two categories by simple inspection and some of them cannot be split. There are three categories in our own dataset - CD, apple and lychee . These images have been collected from the Internet. In 'CD' category, there are some synthetic CD images which do not have any texture content. This is done intentionally to analyze the sub-categorization result. All images have been converted to grayscale before processing.

4.1 Sub-category Discovery

In this experiments, we learn two topics from each of the eight object categories. Consequently, we split each object into two sub-categories. Then we compute

s for each of the objects. For CD, apple, motorbike, rooster and face, sub-categorization is distinctive. Some of these sub-categorization results are shown in Figure 3. The first sub-category of CD contains images with plain CDs and the textured CDs are moved to sub-category 2. In the case of apple, motorbike and face, the images with plain background are placed into sub-category 1 whether the images with cluttered background are placed into sub-category 2. The value of s is high for CD, apple, motorbike, rooster and face categories. On the contrary, in case of sunflower, value of s is low and therefore two sub-categories are not distinctive (see Figure 4).

In the experiments, we choose the number of sub-category as two. However, it is possible to consider more than two subcategories. However, as we have seen in many of the Caltech dataset objects, when the number of sub-categories is more than two, number of training images falling into some sub-categories become very small. In such cases, object models learnt from these small sub-training sets become imperfect and produces many errors during testing. This results in wrong classification even after the classifier combination.

From Figure 3, it can be noticed that sometimes the sub-categorization process utilize the background information. Although it seems unnatural, it may improve the performance during recognition as the background feature servs as context information.

4.2 Object Classification

We consider binary classification in the experiments. Each of the eight object category is classified against background images from Caltech dataset. Both the single and parallel classifiers are evaluated. On the object categories with high s values (e.g. CD, apple, motorbike and rooster), parallel classifier performs better as the sub-categorization is distinctive. On the other hand, single classifier works better on objects with low s value (such as sunflower). Both methods perform similarly on the objects where s values are moderate (such as face and

Table 1. Comparison of error rates

object	Error rate, %			s	no. of train/ test images
	Naive Bayes (without grouping)	Naive Bayes (with grouping)	SVM		
CD	10	2.5	2	0.89	40/30
apple	15	7	8	0.75	80/70
watch	23	21	23	0.59	100/60
sunflower	24	39	17	0.3	85/74
lychee	30	30	16	0.42	69/61
motorbike	10	8	11	0.66	138/140
face	3	3	3.5	0.57	140/138
rooster	13	9	12	0.6	66/63

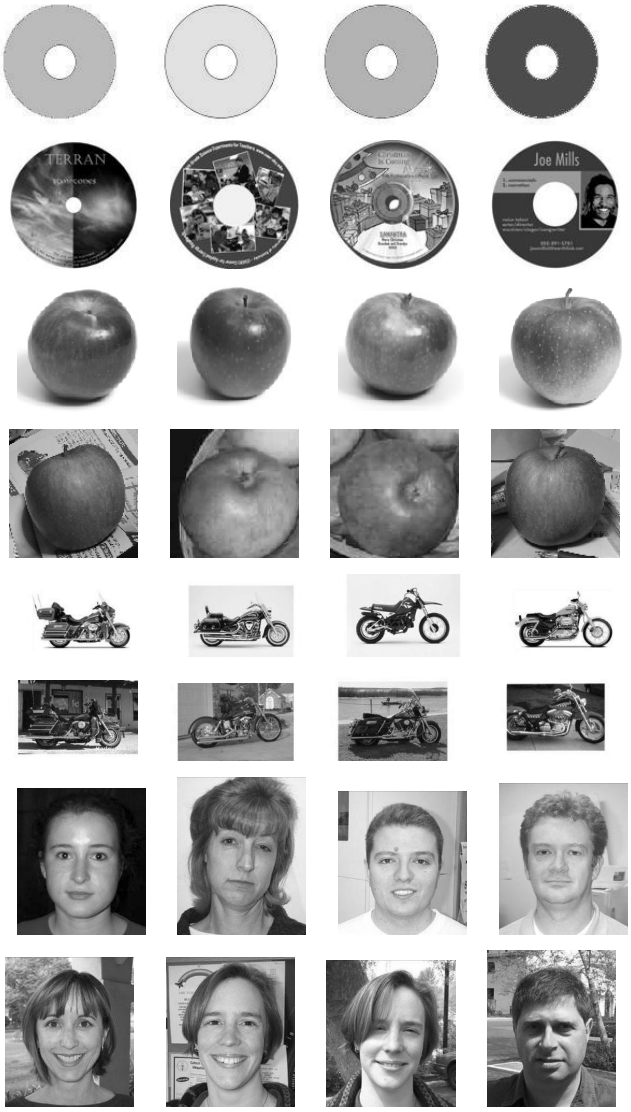

Fig. 3. Distinctive sub-categorization; row 1: CD-subcategory 1, row 2: CD-subcategory 2, row 3: apple-subcategory 1, row 4: apple-subcategory 2, row 5: motorbike-subcategory 1, row 6: motorbike-subcategory 2, row 7: face-subcategory 1, row 8: face-subcategory 2

lychee). Classification results has been given in Table 1. This results validate the advantage of splitting a category into sub-categories and using separate classifier for each sub-category.

Fig. 4. Non-distinctive sub-categorization; row 1: sunflower-subcategory 1, row 2: sunflower-subcategory 2

4.3 Comparison with SVM

In Table 1, we compare the error rates of Bayesian classifier with that of SVM. SVM is known to be one of the best performing classifier in pattern classification problems. Although naive Bayesian framework is one of the simplest and primitive classifier, it performed equally or better than SVM on the cases where splitting of training set is recommended (depending on the value of s). In SVM, we have used radial basis function kernel so that it can learn a non-linear decision surface.

5 Conclusion

We have presented a straightforward and novel approach to improve the recognition rate of a Bayesian classifier by splitting the training data and using separate classifier to learn each sub-category. To split the data automatically, we use pLSA. However, splitting results in poor performances in some case. We compute the average difference between posteriors from the pLSA model, and observing this parameter, one can decide whether splitting is useful or not. This approach has been evaluated on eight object categories. The results demonstrate that the approach can improve the recognition performance considerably.

In this paper, we considered only two sub-categories. In future work, we will evaluate the performance of the system with more than two sub-categories. In our experiments with Caltech dataset and our own dataset, we found that sub-categorization was based on the object textures and the type of background. It will be worthwhile to test the approach with other dataset and to observe the criterion for sub-categorization.

Acknowledgements

This work was supported in part by the Ministry of Internal Affairs and Communications under SCOPE and by the Ministry of Education, Culture, Sports,

Science and Technology under the Grant-in-Aid for Scientific Research (KAKENHI 19300055).

References

1. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV (2004)
2. Fei-Fei, L., Fergus, R., Perona, P.: A Bayesian approach to unsupervised one-shot learning of object categories. In: ICCV, pp. 1134–1141 (2003)
3. Fei-Fei, L., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: CVPR (2005)
4. Hofmann, T.: Probabilistic latent semantic indexing. In: ACM SIGIR (1998)
5. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Machine Learning 41, 177–196 (2001)
6. Sivic, J., Russell, B., Efros, A., Zisserman, A., Freeman, W.T.: Discovering objets and their locations in images. In: ICCV, Beijing, China (2005)
7. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR (2003)
8. Lowe, D.: Distinctive Image Features from Scale-invariant Keypoints. International Journal of Computer Vision 60, 91–110 (2004)