Combining Hiearachical Clustering and Naive-Bayes Nearest-Neighbor For Image Classification

Chen Fu
International School of Beijing University of Posts and Telecommunications

Shijie Jia
College of Electrical & Information
Dalian Jiaotong University

Abstract—This paper addresses the problem of image classification, which has successful applications in many fields, such as image retrieve, object detection and recognition. As a simple nonparametric methods, Naive-Bayes Nearest-Neighbor(NBNN) employs NN distances in the space of the local image descriptors and computes direct 'Image-to-Class' distances without descriptor quantization. However, the high compute cost is the severe bottleneck for the generation of NBNN, which is particular for large scale image classification. Under the naïve bayes assumptation (the local features are considered i.i.d.), this paper combines hierarchical clustering with Naive-Bayes nearest-neighbor for image classification, which provides high scalability to the tradeoff between accuracy and efficiency. Furthermore, this paper proposes the scheme of category cutting, which could improve the test speed at the cost of little accuracy dropping, which is particularly efficient for large scale image classification. Experiments showed that our scheme is the most efficiency among LSH, kd-tree, and linear search algorithms.

Keywords-Naïve Nearst Neighbour Classifier; image classification;Imageto-Class distance hierarchical clustering; category cutting ;

I. INTRODUCTION

The author addresses the problem of image classification, which has successful applications in many fields, such as image retrieve, object detection and recognition. Until now, there are numerous of solutions for image classification[1-6], which are roughly divided into two groups, i.e. the parameter family[1-4, 7-9] and no-parameter family[5, 6]. The first group requires an intensive learning/training phase of the classifier parameters such as parameters of SVM. It is also known as learning-based classifier family. Conversely, the non-parametric classifiers make the decisions based directly on the data, needless of parameters learning/training. For example, NN-based classifiers make decision simplify relying on Nearest-Neighbor (NN) distance estimation. In practice, the parametric classifiers perform much effective than the non-parametric methods. However, the learning based approaches are hard to avoid over fitting of parameters, which is none-existence for the non-parametric classifiers (as there is no parameter to learn!). Furthermore, as the number of classes increase, the parameters would explode, which would bring a tremendous burden to learning machines. Conversely, the latter can naturally handle a huge number of classes.

For the state of art approach "bags-of-words[10]", the local features are extracted and represented as local (such as SIFT[11]) descriptors; the descriptors are clustered into some "visual words" and each descriptor corresponds to the least-distance visual word of the codebook. The visual word histogram representation for an image gives rise to a significant dimensionality reduction, but also to significant degradation because of the quantification. The learning based methods have a training phase to compensate for this loss of information, while the nonparametric classifiers have no training phase and thereafter cause the performances degradation of classification. In addition, the nonparametric methods employ 'Image-to-Image' distance, instead of 'Image-to-Class' distance. Only when the query image is similar to one of the database images, it makes good performances, but does not generalize much beyond the labeled images.

Ref [6] proposes a remarkably simple nonparametric Naive-Bayes Nearest-Neighbor(NBNN) based classifier, which requires no descriptor quantization, and employs a direct "Image-to-Class" distance. They `further show that under the Naive-Bayes assumption, the theoretically optimal image classifier can be accurately approximated by NBNN. Empirical comparisons are shown its performance ranks among the top leading learning-based image classifiers. However, the high compute cost is the severe bottleneck for the generation of NBNN. As the brute force exhaustive linear search is too costly for many large scale applications, Ref [6] employed kd-tree to make search efficiency. In fact, the kd tree is a useful tool in Euclidian spaces of only moderate dimension (<20) [12]. For the high dimensional spaces (such as 128- dimension SIFT descriptors), kd tree provides no speedup over exhaustive search, whose complexity is O(nD).

This paper explores efficient ways to degrade the compute complexity of NBNN. People are intensely interested in the approximate nearest neighbor search algorithms, which can be orders of magnitude faster than exact search, while can still provide near optimal accuracy. Inspired by the clustering process of vocabulary tree [13], hierarchical clustering is employed for the local descriptors, which provides high scalability to the tradeoff between accuracy and efficiency. Furthermore, a novel scheme of category cutting is proposed to improve the test speed greatly at the cost of little accuracy dropping.

The rest of the paper is organized as follows. Section 2 describes the framework of NBNN approach, including some improving schemes. Experiment setup and typical results are described in Section 3. The final part concludes suggestions for future research.

II. OUR APPROACH

The proposed approach employes hierarchical clustering technology to computer the image-to-class distance, which makes Naive-Bayes Nearest-Neighbor be much more efficiency for image classification. The diagram is illustrated in Fig.1.

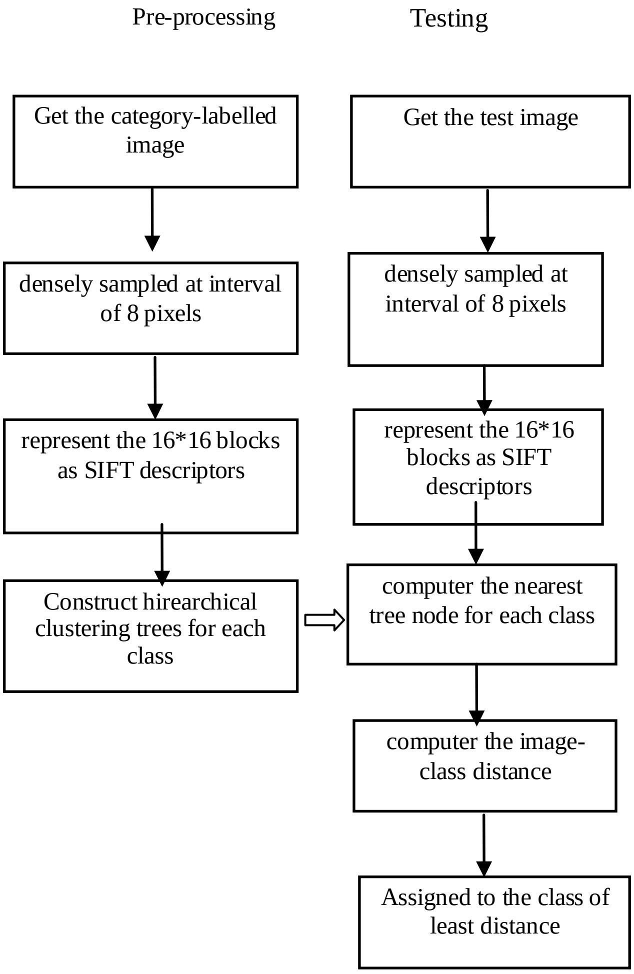
Pre-processing
Testing
Get the category-labelled image
densely sampled at interval of 8 pixels
represent the 16*16 blocks as SIFT descriptors
Construct hirearchical clustering trees for each class
Get the test image
densely sampled at interval of 8 pixels
represent the 16*16 blocks as SIFT descriptors
computer the nearest tree node for each class
computer the image-class distance
Assigned to the class of least distance

Figure 1. The flow-process diagram of our appoarch

A. *Representation of images and image categories*

Each labeled image is densely sampled at 8 pixels interval (spaced by rows and columns) and the region of 16*16 pixels size around the sample points are represented as 128-dimensions SIFT descriptors, in which the chrominance information is ignored. After the processing for all the labeled images, each image category is represented as a specified unordered pooling of category-labeled local features. The query image is also densely sampled at 8 pixels interval and represented as a number of SIFT descriptors.

B. *Hierarchical clustering*

The hierarchical k-means clustering scheme is employed to partition the class-specific feature space iteratively. On the first clustering level, all the class-specific descriptors are clustered into several partitions with K-means clustering algorithm. On the next level, the descriptors of each partition are divided into some sub-partitions independently applying K-means clustering. The splitting processing can be continued iteratively until the descriptors affiliated with the sub-partition are less enough for search efficiency. The number of hierarchical levels and the clustering nodes at different levels are chosen according to the scale of descriptors, and the parameters setup can be varied from class to class. For the large scale feature space, the K-means clustering process can inevitably become time-costing. To degrade the computer complexity, an alternative scheme is to randomly choose some descriptors rather than the whole to calculate the centroids at the cost of little accuracy lost. In fact, the hierarchical k-means clustering for each image category can be seen as a preprocessing procedure, which is not a really important factor of efficiency. It is also important to note that the hierarchical clustering minimizes distortion only locally at each node rather than in a minimization of the total distortion.

Assuming two clustering levels are implemented for the descriptors of each image category, the numbers of centroids at different levels are set as K1, K2, respectively. The image-to-class distances can be calculated at different levels:

“Level 1”. For each descriptor extracted from the query image, compute the least distance to the centroids on the first level, the sum of the least distances is taken as the image-to-class distance. The time complexity is O(K1*N), where N is the number of descriptors of the query image.

“Level 2”. For each descriptor of the query image, find the nearest centroid on the first level; then compute the least distance to the centroids on the second level corresponding to the nearest first level centroid, the sum of the least distances is taken as the image-to-class distance. The time complexity is O((K1+K2)*N).

“Level 3”. For each descriptor of the query image, find the nearest first-level centroid and the corresponding nearest second-level centroid in sequence, then compute the least distance to descriptors around the nearest second-level centroid. The sum of the least distances is taken as the image-to-class distance. The complexity is O(N *(K1+K2+ K3), where K3 is the average number of descriptors associated with the second-level clustering centroids.

The three level image-to-class distance schemes mentioned result in a tradeoff between accuracy and efficiency for image classification. The first method (“level 1”) is the most efficient and takes up least memory space, but the large quantization error degrade greatly the search accuracy. The second method (“level 2”) decreases the quantization error to boost the accuracy performance at the cost of more time and space complexity. Unlike the two methods of computing descriptor-to-centriod distances, the third one (“level 3”) computes directly the descriptor-to- descriptor distances. The search scope is shrunk by the consequence two level filters, decreasing to the average of 1/1(K1*K2) of the total descriptors. Comparing to the first two methods, this method can bring more accuracy, but the time and space complexity are also greatly growing. Nevertheless, it is still much more efficiency than exact search.

It is important to note that the nearest neighbors are not definitely to be assigned one centroid. If the query descriptor and its nearest descriptor are assigned to different clustering, all the methods mentioned above cannot get the real least distance from the query descriptor to the feature space, which will inevitably bring errors to the image-to-class distance. Our solution is to replace the hard assignment (each descriptor is assigned to one centroid) with the soft assignment, in which each descriptor is assigned to several(M) centroids. The weights of the query descriptor to M classes are normalized by the distances from the query descriptor to the neighbor centroids of M classes. The search scope increases to M times approximately than that of Method 3.

C. *Power-law distance calculation*

For each test descriptor x_i , we compute the Euclidean distance d_i between x_i and its nearest neighbor $NN(x_i)$.

$$d_i = \| x_i - NN(x_i) \|) \tag{5}$$

For method 1 and 2, $NN(x_i)$ is a centroid, while for method 3 it is a descriptor. The Euclidean distance is normalized to [0 1] through power-law weighting:

$$d_i \Leftarrow 1 - \exp(-a * d_i) \tag{6}$$

Where a is the normalization parameter which is positive and class-independent. The normalized distance of query image to class *c*, d^c , is calculated as:

$$d^c = \sum_i d_i^c / N_q \tag{7}$$

where N_q is the number of descriptors extracted from the query image. The category label C of the query image is the class with the least normalized distance:

$$C = \min_c (d^c) \tag{8}$$

III. EXPERIMENT EVALUATION

A. *Experiment setup*

The proposed algorithm is evaluated on two image datasets, PI 100 and Caltech 101. The first dataset is exclusively collected for product image retrievel and classification in E-commerce, while the latter is popular for general image classification.

All of the experiments were performed on an Intel Pentium CPU 2.66GHz computer running Windows XP with 2GB RAM, implemented by mixing programming of MATLAB2010 and VC++6.0. In this experiment, 10% images of each category are randomly chosen as the query set. The hierarchical levels are set to 3 and the numbers of centroids at the first and the second level are set to 10, respectively. At the same set, compare the method with exhaust search, kdtree and LSH. Each experiment was repeated 5 times and we report the average results over all experiments.

TABLE I. THE ACCURACY AND TIME COST RESULTS OF FOUR SEARCH METHODS (PI100)

NBNN		Average Accuracy (%)		Classification time			
				Preprocessing(s/ class)		Test (s/image)	
		PI 100	Caltec h 101	PI 100	Caltech 101	PI 100	Caltech 101
Exhaust search		84.2	72.1	2.9	21.1	64.5	132.2
kdtree		84.2	72.1	6.7	25.7	73.2	152.4
LSH		81.1	68.5	10.5	82.3	25.5	50.3
Hierar-chical cluster ing	1	77.6	63.8	3.7	13.4	2.4	5.6
	2	80.9	68.3	5.1	22.4	6.6	13.5
	3	84.0	71.5	5.9	22.4	8.7	19.5

B. *Result &Discussion*

1) Compare hierarchical clustering based NBNN with exhaust search, kdtree, and LSH

Table 1 and Table 2 illustrate the classification results of NBNN on PI 100 and Caltech 101, respectively. Four search methods (exhaust search, kdtree, LSH and hierarchical clustering with power law normalization distance) are separately employed to find the nearest neighbor for NBNN classification. For LSH, the number of hash bucket and hash table are set to 12, 10, respectively.

Some observations from Table 1 and analyses are briefly summarized as below:

(1) Both exhaust and kd-tree are exact search methods, while LSH and hierarchical clustering are implemented for approximate searching. The accuracies of the latter two are slightly worse than the exact search methods but perform is much more efficiency.

(2) The proposed hierarchical clustering based NBNN could get approximate accuracy, but it could arrive at 2.4~8.7s/image (PI 100), which are much efficiency than LSH and kd-tree, the latter even performs less efficiency than the simple exhaust search.

(3) The approach provides a trade-off between the accuracy and the cost. At level 1, only the first-level cluster centers are considered, the image-to-class distance is the sum of the distances between each inquiry descriptor and the nearest clustering centers at the first level. At level 2, the image-to-class distance is the sum of the distances between each inquiry descriptor and the nearest clustering centers at level 2, which is derived from the nearest centroid at level 1. At the third level, the image-to-class distance is the sum of the distances between each inquiry descriptor and the nearest descriptor associated with nearest second level centriod. The lost of accuracy at level 1 and 2 is largely brought by the quantization (employing descriptor-to-centroid distance), which is significantly reduced for level 3 (employing descriptor-to-descriptor distance). As the nearest descriptors are not always falling into the same cluster heap, the accuracy of level 3 is a little inferior to the exact search methods. “Soft assign” scheme can be employed to computer several nearest cluster to make the result more accuracy at the cost of lower efficiency.

2) *Category cutting.*

In addition, the so-called “category cutting” scheme is proposed and implemented to improve efficiency. The basic idea of “category cutting” is to narrow down the category scope as searching down the hierarchy levels. Some dissimilar categories are cut down based on the descriptor-to-centroid distances of the former level. Suppose the number of total categories is N, c% categories are cut down for each level, then at the first level, the distances of each inquiry descriptor should be calculated to N categories; while at the second level, the search scope becomes N(1-c%), and at the third level, the search scope further shrunk to N(1-c%)2. From Table 2, it can be observed that as c rises from 0 to 50%, the accuracy (PI 100) remain unchanged while the test time decrease from 8.4s/image to 5.4s/image. As c increases to 80%, the accuracy only fell by 2.1 percent, while the average test time dropped to 2.8s/image, decreased by about 66.7% percent.

IV. Conclusion

In this paper, the author combines hierarchical clustering with Naive-Bayes nearest-neighbor for image classification and analyze the tradeoff between accuracy and efficiency. Furthermore, this paper proposes the scheme of category cutting, which could improve the test speed at the cost of little accuracy dropping, which is particularly efficient for large scale image classification. Experiments showed that the scheme is the most efficiency among LSH, kd-tree, and linear search algorithms.

TABLE II. THE ACCURACY AND TIME COST RESULTS WITH DIFFERENT PERCENTAGE OF CLASSES(PI 100)

Percentage of category cutting (%)	Average Accuracy(%)		Test time(s/image)	
	PI 100	Caltech 101	PI 100	Caltech 101
0	84.0	71.5	8.4	19.5
10	84.0	71.5	7.9	18.9
20	84.0	71.5	7.1	17.4
30	84.0	71.2	6.3	15.4
40	84.0	70.8	6.0	13.1
50	84.0	70.1	5.4	12.9
60	83.4	68,3	4.6	11.7
70	83.0	63.5	3.7	10.9
80	81.9	61.0	3.6	8.0
90	78.6	60.5	2.8	5.5

However, the better performance is achieved on the balanced categories, in which the numbers and the resolutions of images are equilibrium distribution in all the categories, that is, assuming similar densities in feature space for all classes, such that the same kernel bandwidth can be used for all of them. In practice, this assumption is often violated, resulting in a strong bias towards one or a few object classes. The future study should focus on the large scale image classification for unbalanced categories.

Acknowledgments

The research work was supported by National Natural Science Foundation of China under Grant No. F010408 and Science Project of Liaoning Education Bureau under Grant No. L2014174.

References

[1] Perronnin F, J Sanchez, and T Mensink. Improving the fisher kernel for large-scale image classification[C]. Computer Vision (ECCV).2010: 143-156.

[2] Siddiquie, B., S. Vitaladevuni, and L. Davis, Combining Multiple Kernels for Efficient Image Classification[C]. Workshop on Applications of Computer Vision (WACV), 2009:1-8.

[3] Chapelle, O., P. Haffner, and V. Vapnik, SVMs for histogram-based image classification[J]. IEEE transactions on Neural Networks, 1999,10(5): 1055-1064.

[4] Zhang H, Berg A C, Maire M, et al. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition(CVPR) ,2006,2: 2126-2136.

[5] Amato, G. and F. Falchi. kNN based image classification relying on local feature similarity[C]. SISAP '10 Proceedings of the Third International Conference on SImilarity Search and APplications .NY,USA,2010,101-108.

[6] Boiman O., E. Shechtman, and M. Irani. In Defense of Nearest-Neighbor Based Image Classification[C]. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2008:1-8.

[7] Rodriguez F, G.Sapiro and M U MINNEAPOLIS. Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries Technical report, University of Minnesota, December 2007. IMA Preprint, www.ima.umn.edu.

[8] Yang J, Kai Y, Yihong G and Huang T. Linear spatial pyramid matching using sparse coding for image classification[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2009:1794-1801.

[9] Zhou X, Na C, Zhen L, et al.Hierarchical gaussianization for image classification[C]. 12th IEEE International Conference on Computer Vision(ICCV) . 2009:1971-1977.

[10] Sivic J. and Zisserman A. Video Google: A Text Retrieval Approach to Object Matching in Videos[C]. Ninth IEEE International Conference onComputer Vision(ICCV). 2003,2:1470-1477.

[11] Lowe D. Distinctive Image Features from Scale-Invariant Keypoints[J]. Int. J. Comput. Vision, 2004, 60(2):91-110.

[12] Weber R, Schek H J and Blott S. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces[J]. INSTITUTE OF ELECTRICAL & ELECTRONICS ENGINEERS. 1998:1-12.

[13] Moosmann F, Triggs W, and Jurie F. Randomized clustering forests for building fast and discriminative visual vocabularies. Advances in Neural Information Processing Systems[C].2006.

[14] Sebe N. Machine learning in compter vision[M]. 2005.

[15] Behmo R, Paul M, Arnak D et al. Towards optimal naive Bayes nearest neighbor[C]. Computer Vision(ECCV),2010:171-184.

[16] Tuytelaars T, Fritz M, Saenko K, et al.The NBNN kernel[C]. IEEE International Conference on Computer Vision(ICCV).2011:1824-1831.

[17] X Xie, L Lu, M Jia, et al. Mobile search with multimodal queries[J]. Proceedings of the IEEE, 2008，96(4):589-601.

[18] Fei-Fei L, Fergus R and Perona P. Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories[J].Computer Vision and Image Understanding, 2007, 106(1): 59-70.