A GRAPH-THEORETIC APPROACH TO CLASSIFIER COMBINATION

Jian Hou[*]

College of Information Science and Technology
Bohai University, China

Zhan-Shen Feng, Bo-Ping Zhang

School of Computer Science
Xuchang University, China

ABSTRACT

Classifier combination can be used to combine multiple classification decisions to improve object classification performance, and weighted average is a popular method for this purpose. In this paper we propose to use a graph-theoretic clustering method to define the weights for SVM classifier decisions. Specifically, we use the dominant set clustering to evaluate the difficulty of a kernel matrix for a SVM classifier. This degree of difficulty is found to be related to the SVM classification performance and thus used to define the weight of this classifier. Though simple and intuitive, the method is shown to be as powerful as more sophisticated methods in extensive experiments with several datasets of diverse object types.

Index Terms— classifier combination, weight, graph-theoretic, object classification

1. INTRODUCTION

Classifier combination is used to combine the strength of multiple classifiers and produce better performance than individual classifiers. Based on the level at which they operate, classifier combination can be categorized into two types. The first one combines all features into one final feature, which is then used in classification. The second type, denoted by classifier fusion in this paper, fuses the decisions or scores of all classifiers and produce one final decision or score. Classifier fusion is attractive in that different types of classifiers, e.g., SVM and k-NN, can be combined together. We focus on classifier fusion in this paper.

Majority voting is the most simple method for combining the decisions of individual classifiers into one final decision. This approach counts the number of each label and selects the label with largest number as the final decision. Majority voting uses only the class labeling information and discards the probability information of the label. Another approach is to use in combination the posterior probability of each training class, i.e., the soft label. Some popular methods for this approach include weighted sum, logistic regression [1], Dempster-Shafer rules [2] and neural networks [3].

[*]This work is supported by NSFC project 61171189 and NSF of The Education Department of Henan Province under contract No. 12A520038.

In this paper we present a weighting scheme for SVM classifier combination based on a graph-theoretic concept. Specifically, we use dominant set clustering [4] to evaluate how difficult a kernel matrix is for a SVM classifier to classify. This degree of difficulty is related to the prediction accuracy of this classifier and thus the reliability of the soft labels. Therefore we use the degree of difficulty to define kernel accuracy and then use kernel accuracy as the weight of this kernel matrix in combination. These kernel accuracies are defined separately, i.e., given a kernel matrix, our method outputs its accuracy. The approach is intuitive and simple, but shown to be effective in comparison with other combination methods in experiments.

In Section 2 we present a brief introduction of the dominant set clustering[4]. Section 3 details our method to compute the weight of a classifier with dominant set clustering. We report the experimental results in Section 4 with comparison with other combination methods and literature. Section 5 concludes the paper.

2. DOMINANT SET

There is no formal definition of clustering. However, it's usually agreed that a cluster should satisfy the constraints of internal coherency and external incoherency. In other words, the clustering of a given dataset is totally determined by the pairwise similarity distribution of data. The popular k-means clustering method, however, requires users to input the number of clusters k.

In [4] the authors present dominant set as a graph-theoretic concept of clustering. Representing the data for clustering as an undirected edge-weighted graph, [4] define a dominant set as a locally maximal subset of vertices that are internally coherent. After we extract a dominant set, we remove its included vertices from the graph and extract another dominant set from remaining graph. Repeating this procedure until all vertices are included in dominant sets, we are faced with a partition of all data where each partition is a dominant set. It's evident that these dominant sets are internally coherent and externally incoherent and this, in turn, means that each dominant set can be regarded as a cluster.

Dominant set based clustering naturally incorporates the internal coherency and external incoherency properties and

determines the number of clusters by itself. This property poses it as an attractive clustering method and explains why we choose dominant set clustering over other clustering methods. Given the pairwise similarity matrix of the data to be clustered, we can easily extract a dominant set with a game dynamics, e.g., the Replicator Dynamics or the infection and immunization dynamics [5]. In our implementation we use the latter for efficiency reason. Due to limited space, we refer the readers to [4] for details of dominant set extraction.

3. DOMINANT SET CLUSTERING BASED WEIGHT

With a kernel matrix as input, a SVM classifier tries to partition training images of different classes. If the training images have high intra-class and low inter-class similarities, it's easy for a SVM to partition different classes with a large margin and produce a high recognition rate. In other words, the possibility of a kernel matrix producing a high recognition rate is decided by to which degree it satisfies the high intra-class and low inter-class similarities constraint. In this paper we call such a measure of a kernel matrix as its accuracy. Intuitively an accurate kernel matrix means a powerful kernel and reliable outputs, and the corresponding classification results should be given a big weight in combination.

By training labels we obtain a partition of training examples where each part corresponds to one class. This is the ideal partition we expect a SVM classifier to achieve. By dominant set clustering in the kernel matrix (also a similarity matrix) we obtain another partition of the training examples, where each part corresponds to a dominant set (cluster). Note that in dominant set clustering, the clusters satisfy the constraint of high intra-cluster and low inter-cluster similarity, and that the partition by a SVM classifier satisfy the same constraints. We see that the partition by dominant set clustering is the one a SVM is likely to achieve with a large margin. Now it's natural to conclude that the accuracy of a kernel matrix can be defined by to which degree the real partition by dominant set clustering is close to the expected partition by training labels. We don't adopt the k-means-like clustering here based on the fact that different k yields different partitions and it's not clear which one to choose.

Ideally the two partitions coincidence with each other. In this case the kernel matrix strictly satisfies the constraint of high intra-class and low inter-class similarity and we define the kernel accuracy to be 1. This is not to say that the kernel matrix will produce a 100% recognition rate, but that in our framework the potential of similarity distribution of the kernel matrix has been fully explored to obtain an accurate classification.

Obviously the above ideal case is unlikely to exist in practice. In fact, the number of dominant sets is almost always much larger than the number of classes. The single-class dominant sets contain subsets of one classes and multi-class dominant sets contain subsets of multiple classes. As a result, the partitions by dominant set and by training labels have much overlaps and intersections. In the following we calculate of the closeness of two partitions.

From the perspective of training labels, one class may be occupied by one single-class dominant or one multi-class dominant set, or shared by some single-class dominant sets or some multi-class dominant sets. We will analyze the four cases one by one and calculate an accuracy for each class. Finally we use the average of the accuracies of all classes as the kernel accuracy.

One class occupied by one single-class dominant set is actually the ideal case. All training examples of the same class have high similarity with each other and low similarity with examples of other classes. This is easy for a SVM to classify and we define the accuracy of such a class to be 1.

One class occupied by one multi-class dominant set means that while all training examples of the same class are highly similar to each other, they are also similar to other classes. Intuitively, a larger share of the class in the multi-class dominant set means fewer examples of other classes are involved, and thus a bigger accuracy for the class. This observation can be expressed as

$$r^{'}_{share} = \frac{N_{dset\_in\_class}}{N_{dset}} \tag{1}$$

where N_{dset} is the number of examples in the dominant set, and $N_{dset\_in\_class}$ is the number of examples in the overlap of the class and the dominant set. It's easy to see that for this case and the first case, the kernel accuracy can be expressed as $r^{'}_{share}$.

If one class is shared by some single-class dominant sets, no examples in the class are very similar to other classes. However, more than one single-class dominant sets also imply that some examples in the class are not very similar to others in the same class. Obviously this will impact on the classification performance. If one of these dominant sets is very large and all others are very small, this case looks similar to the first one and the negative effect is fairly small. On the other hand, if all dominant sets are of roughly the same size, the negative effect will be rather notable. We express this negative effect as a factor

$$r_{class} = \sqrt{\frac{\sum_{i=1}^{M}(N^{(i)}_{dset})^2}{(\sum_{i=1}^{M} N^{(i)}_{dset})^2}} \tag{2}$$

where M is the number of dominant sets in the class, and $N^{(i)}_{dset}$ is the number of examples in the i-th dominant set. In this case the kernel accuracy can be represented by r_{class}. Note that the first case can also be accommodated by this expression.

In case of one class shared by some multi-class dominant sets, we must take into account all the factors involved in the above three cases, i.e. $r^{'}_{share}$ and r_{class}. Note that the expression of $r^{'}_{share}$ in (1) is only for the case of one dominant set.

We extend it to be

$$r_{share} = \frac{\sum_{i=1}^{M} N_{dset\_in\_class}^{(i)}}{\sum_{i=1}^{M} N_{dset}^{(i)}} \quad (3)$$

Then a simple expression of one class's accuracy can be selected as

$$P_{class} = r_{share} r_{class} \quad (4)$$

It is easy to verify that this expression is applicable to all four cases.

Now we can define the weight of kernel based on its accuracy. In implementation we use $w_{label} = \frac{\sum P_{class}}{N_c}$ to compute the weight for each kernel, where N_c is the number of classes. After calculating the accuracy of each kernel matrix, we can use it as the weight in kernel matrix combination. In implementation we use w_{label}^4 instead of w_{label} as the weight to highlight the difference between different kernels.

4. EXPERIMENTS

We test our weighting scheme in classifier combination with SVM classification experiments. In all experiments the regulation parameter C is fixed to be 1000. The multi-class SVM is trained in a one-versus-all mode. When distances are used to build kernels, the transformation is in the form of $k(x, y) = exp(-d_0^{-1} d(x, y))$ where d is the pairwise distances and d_0 is the mean of pairwise distances. The experimental setups and accuracy measures are selected to be same as the literatures used for comparison. The experiments are repeated 10 times with different training-testing splits and the average of recognition rates are reported. The following 3 diverse datasets are adopted in experiments.

The Oxford Flower-17 dataset [6] is composed of flower images of 17 categories with 80 images in each category. For ease of comparison, we use the distance matrices and the 3 predefined training-testing splits provided by the authors for combination. The 7 kernels are from [6] and [7]. We report the overall accuracy and comparison with literature in Table 1.

The Event-8 dataset [8] consists of images from 8 sports events categories with 130 to 250 images in each category. Following the setup in [8], we randomly select 70 images per class as training and another 60 images as testing, and report the 8-class overall recognition rate. The Scene-15 dataset [9] contains images from 15 categories with 200 to 400 images in each category. We follow the experimental setup in [9], i.e., randomly select 100 images per class as training, with all the others as testing, and report the mean recognition rate per class.

For Event-8 and Scene-15, we use the following features to build kernel matrices.

PHOG Shape Descriptor. Oriented (20 bins) and unoriented (40 bins) PHOG [10] are constructed from level 0 to 3. Different from the implementation in [10], in this paper the descriptor of level L is just composed of its 2^L windows, with no addition from lower levels.

Table 1. Flower-17 recognition rates and comparison.

method	accuracy
best single	70.6 ± 1.6
average	85.8 ± 2.7
this paper	85.7 ± 2.4
[7]	88.3 ± 0.3
[16]	85.5 ± 3.0
[17]	82.6 ± 0.3

Table 2. Event-8 and Scene-15 recognition rates.

Event-8		Scene-15	
method	accuracy	method	accuracy
best single	84.4 ± 1.7	best single	79.6 ± 0.4
average	85.1 ± 1.1	average	81.9 ± 0.6
this paper	88.9 ± 0.9	this paper	84.8 ± 0.4
[18]	84.2 ± 1.0	[19]	86.7 ± 0.4
[8]	73.4	[18]	84.1 ± 0.5
		[9]	81.4 ± 0.5

Bag of Visual Words. We use SIFT descriptors[11] on 16×16 patches with spacing of 8 pixels to build a 500-bin vocabulary. The descriptors are extracted in gray (128d), HSV (384d) and CIE-Lab (384d) spaces for the Event and in gray space for the Scene. The visual words histograms are built in a pyramid from level 0 to 1.

Locally Binary Patterns. The basic locally binary patterns (LBP) [12] are extracted and clustered to create a descriptor for one image. The descriptor length is 256 and we built it from level 0 to 2.

Gray Value Histogram. We also use the 64-bin gray value histograms from level 0 to 3.

Gist Descriptor. The gist descriptor [13] are extracted in a pyramid from level 0 to 1.

Self-similarity Descriptor. Self-similarity descriptors [14] of 30 dimensions (10 orientations and 3 radial bins) are extracted and quantized into a vocabulary of 500 bins. The histogram is built from level 0 to 1.

Gabor and RFS filters. We use two texture features: Gabor and RFS filters [15] to build histograms (500 bins) from level 0 to 1.

The selected distance measures are χ^2 distance and Earth Mover's Distance (EMD). Altogether we use 58 kernels for Event-8 and 50 kernels for Scene-15 in combination. We report the average results and comparison in Table 2.

On Oxford flower dataset, our method produces better results than LP-β method in [16] and the MKL methods in [17] and [6]. These comparisons indicate that with proper definition, our simple, intuitive kernel accuracy weighting can be

as powerful as the other sophisticated optimization methods. From Table 2 we observed that while average combination is better than the best single feature, our weighted combination further improve the results considerably for both datasets. These results imply that with relatively simple features and distance measures, our combination method produces a significant improvement on classification performance.

5. CONCLUSION

We proposed a weighting scheme for classifier combination in object classification based on dominant set clustering. We partition the training images by enumerating all the dominant sets in a kernel matrix. The partition is used to evaluate how difficult the kernel matrix is for a SVM classifier, and thus the possibility of producing a high recognition rate. Based on this evaluation, we define the accuracy of a kernel matrix and use it as the weight in classifier combination. We tested the method in experiments with several datasets of diverse object types and observed considerable improvement over benchmark combination methods. The results are also comparable to the state of the art obtained with more sophisticated methods.

6. REFERENCES

[1] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Transaction on Pattern Analsis and Machine Intelligence*, vol. 16, no. 1, pp. 66–75, 1994.

[2] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods for combining multiple classifiers and their applications to handwriting recognition," *IEEE Transaction on System, Man, and Cybernetics*, vol. 23, no. 3, pp. 418–435, 1992.

[3] D. S. Lee, "Theory of classifier combination: the neural network approach," Ph.d thesis, SUNY at Buffalo, 1995.

[4] M. Pavan and M. Pelillo, "Dominant sets and pairwise clustering," *IEEE Transaction on Pattern Analsis and Machine Intelligence*, vol. 29, no. 1, pp. 167–172, 2007.

[5] S. R. Bulo, "A game-theoretic framework for similarity-based data clustering," Phd thesis, University Ca' Foscari, 2009.

[6] M. E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *IEEE International Conference on Computer Vision*, 2006, pp. 1447–1454.

[7] M. E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008, pp. 722–729.

[8] L. L. Jia and L. Fei-Fei, "What, where and who? classifying event by scene and object recognition," in *IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[9] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006, vol. 2, pp. 2169–2178.

[10] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *ACM International Conference on Image and video retrieval*, 2007, pp. 401–408.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[12] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transaction on Pattern Analsis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[13] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[14] E. Shechtman and M. Irani, "Matching local self-similarities across imagesn and videos," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[15] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *Image and Vision Computing*, vol. 62, no. 1-2, pp. 61–81, 2005.

[16] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *IEEE International Conference on Computer Vision*, 2009, pp. 221–228.

[17] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[18] J. X. Wu and J. M. Rehg, "Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel," in *IEEE International Conference on Computer Vision*, 2009, pp. 630–637.

[19] L. Bo, X. Ren, and D. Fox, "Kernel descriptors for visual recognition," in *Advances in Neural Information Processing Systems*, 2010.