Pattern Recognition 46 (2013) 3129–3139

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A simple feature combination method based on dominant sets

Jian Hou [a], Marcello Pelillo [b,*]

[a] School of Information Science and Technology, Bohai University, Jinzhou, China
[b] DAIS, Università Ca' Foscari di Venezia, Via Torino 155, 30172 Venezia Mestre, Italy

ARTICLE INFO

Article history:
Received 6 March 2012
Received in revised form
23 February 2013
Accepted 3 April 2013
Available online 22 April 2013

Keywords:
Feature combination
Object classification
Dominant sets
Kernel accuracy

ABSTRACT

Feature combination is a popular method for improving object classification performances. In this paper we present a simple and effective weighting scheme for feature combination based on the dominant-set notion of a cluster. Specifically, we use dominant sets clustering to evaluate how accurate a kernel matrix is expected to be for a SVM classifier. This expected kernel accuracy reflects the discriminative power of the kernel matrix and thus used in weighting the kernel matrix in feature combination. Our method is simple, intuitive, memory and computation efficient, and performs comparably to the popular and sophisticated optimization based methods. We conduct experiments with several datasets of diverse object types and validate the effectiveness of the proposed method. In fact, in five out of the six datasets used in our experiments, we obtained the best results until now in our knowledge.

1. Introduction

In order to design an effective object classification system, feature combination is usually adopted in an attempt to combine the strengths of multiple complementary features and produce better performance than any individual feature. Feature combination methods can be categorized into two types according to the level at which they operate [32]. The first one uses features of all individual classifiers to form a joint feature vector, which is then used in later classification. In the case of support vector machine (SVM) classification, for example, feature combination translates to combining a set of kernel functions into one final kernel function. The second type operates at the decision or the score level, namely, the outputs of all individual classifiers are used in combination. This approach is attractive as different types of classifiers, e.g., SVM and k-NN, can be combined together. In this paper we focus on kernel combination with applications to SVM classification.

Usually the kernel combination problem refers to the process to find the best final kernel from the weighted sum of given kernels, i.e., $k^*(x,y)=\sum_{i=1}^{n} w_i k_i(x,y)$, where the weights $w_i, i=1,\ldots,n$ are what we need. Average combination is the simplest combination method and widely used as the benchmark for comparison with other combination methods. In average combination, all participating kernels are given equal weights, regardless of how they perform in practice. Intuitively this is not an optimal solution as we tend to believe that kernels with larger discriminate power should be given larger weights in order to obtain the best combination performance. Based on this intuition, a straightforward approach is to estimate the discriminative power of kernels with cross-validation and then define the weights of kernels in combination. In this paper, however, we propose another approach to make use of the intuition from a difference perspective. Unlike the cross-validation method doing classification inside training examples, our method is based on the correlation between the SVM classification mechanism and dominant sets clustering [24,25,30]. In other words, no classification procedures are involved in our method. For ease of expression, in this paper we call the estimated discriminative power of a kernel as the kernel's *accuracy*. Intuitively, a kernel with a larger accuracy should be given a larger weight in combination and vice versa.

In our method, the kernel accuracy measured by dominant sets clustering reflects how accurate a kernel is for SVM classification, and thus is used to weight the kernel in combination. Unlike MKL [16] or LPBoost [11] calculating the weights from optimization with all kernels, our method computes the weights of kernels separately, i.e., given a kernel, our method outputs its weight in combination. This implies that in the case that a very large number of kernels are used in combination, e.g., [1], our method requires much smaller memory than optimization based methods. While the cross-validation method is also memory efficient, experiments in Section 5 indicate that our approach produces better performance with smaller computation consumption. Our approach is intuitive and simple, but is shown to be effective in comparison with other combination methods on a variety of datasets.

The paper is organized as follows. In Section 2 we briefly review some of the major research advances in kernel combination, and show how they inspire our work in this paper. Section 3 introduces

[*] Corresponding author. Tel.: +39 041 2348 440
E-mail addresses: dr.houjian@gmail.com (J. Hou), pelillo@dsi.unive.it (M. Pelillo).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.04.005

the dominant set concept of a cluster, which is used to determine the kernel weights in Section 4. In Section 4 we detail the method to compute the weights of kernels in combination based on dominant sets clustering. The experimental results are reported in Section 5 with comparison with other combination methods and the literature. In Section 6 we discuss the experimental results and future plans to enhance the method. Finally, Section 7 concludes the paper.

2. Related works

Average combination and product combination are the two simplest kernel combination methods. They define the final kernel function as $k^*(x,y)=\frac{1}{n}\sum_{i=1}^{n}k_i(x,y)$ and $k^*(x,y)=(\prod_{i=1}^{n}k_i(x,y))^{1/n}$ respectively. A more sophisticated idea is multiple kernel learning (MKL) [16,15,18,35], which seeks to jointly optimize the weights w_i of all kernels in $k^*(x,y)=\sum_{i=1}^{n}w_ik_i(x,y)$ and the SVM parameters. In recent advances in MKL, [40] proposed a non-linear kernel combination method, i.e., learning different combinations for different data point clusters, and obtained very encouraging performance improvement. Other works on non-linear kernel learning include [6,34]. Another promising direction is to use a very large number of kernels in combination [1]. To efficiently solve the MKL problem, [37] showed that p-norm MKL can be trained using sequential minimal optimization (SMO) algorithm, and thus greatly improves the training speed for large kernel space and large data space. In contrast with MKL, [11] presented LPBoost to train the weights of kernels and SVM parameters in two steps. First the SVMs are trained separately on each kernel. Then the weights of all kernels are optimized in a second step. Experiments on the Caltech datasets validated the effectiveness of this method.

While various works on feature combination have been published in the past decades, there are still many important problems left unsolved in this domain. On one hand, existing combination methods are often computation and memory expensive. The popular MKL-like methods determine the weights of kernels based on the optimization among all participating kernels, and this usually means enormous computation and memory consumption, especially when a large dataset or a very large number of kernels are involved, e.g., the case in Bach [1]. On the other hand, the real effectiveness of the sophisticated, optimization based methods in practical applications has been called in question. In [11] Gehler and Nowozin observed that if all participated features are carefully designed to be powerful, the sophisticated optimization based methods, e.g., MKL, do not show evident advantage over the baseline average combination. Only when both strong and weak features are combined, the optimization based methods reduce the effect of weak features and perform better than average combination. In the supplement to [11] it is also mentioned that the predictive power of learning mixing coefficients seems to be overestimated because of missing comparison with the simple (yet powerful) average combination. Moreover, the supplement claimed that there seems to be an agreement that MKL almost never improves performance. In other words, the sophisticated optimization operations, and also the large amount of computation and memory consumption involved in MKL, seem not necessary at all and the demonstrated performance of MKL in literature might also be obtained by the simple average combination.

Compared with average combination, the popular MKL-like methods obtain tiny, if any, performance gain at the cost of enormous computation and memory consumption. This observation prompts us to reassess the average-like simple combination methods, whose credits are often under-estimated or even ignored just because of their simpleness. With this consideration in mind, and observing the correlation between the SVM classification mechanism and dominant sets clustering, we propose to use dominant sets clustering to evaluate the discriminative power and determine the weights of kernels in combination.

3. Dominant sets and their properties

Dominant set is a graph-theoretic concept of a cluster and dominant sets clustering algorithms have many advantages over classical, e.g., spectral and graph-based, techniques. In particular, they do not require *a priori* knowledge on the number of clusters and make no assumption on the structure of the affinity matrix, being able to work with asymmetric and even negative similarity functions alike [30]. Further, they allow extracting overlapping clusters and generalize naturally to high-order relations [26]. In Section 4 we will see that these nice properties make dominant sets clustering particularly attractive for our purpose of determining kernel accuracy by clustering. Since their introduction in Pavan and Pelillo [24], dominant sets have found a variety of successful applications in such diverse domains as bioinformatics [10], content-based image retrieval [38], human activity analysis [12] and object detection [41], etc.

Unlike traditional approaches to data clustering, which insist on the idea of determining a partition of the input data, dominant sets attempt to provide a formal answer to the question of what is a cluster. Although motivated from purely graph-theoretical concepts, being a generalization of the notion of a maximal clique to edge-weighted graphs, dominant sets turn out to have non-trivial connections to optimization theory and game theory. In this section we provide the basic definitions and properties of dominant sets, which are necessary to understand the proposed method in this paper. The interested reader can find more details in [24,25,30].

We represent the data to be clustered as an undirected edge-weighted graph with no self-loops $G=(V,E,w)$, where $V=\{1,\dots,n\}$ is the vertex set, $E\subseteq V\times V$ is the edge set, and $w: E\to R_+^*$ is the (positive) weight function. Vertices in G correspond to data points, edges represent neighborhood relationships, and edge-weights reflect similarity between pairs of linked vertices. As customary, we represent the graph G with the corresponding weighted adjacency (or similarity) matrix, which is the $n\times n$ nonnegative, symmetric matrix $A=(a_{ij})$ defined as $a_{ij}=w(i,j)$ if $(i,j)\in E$, and $a_{ij}=0$ otherwise. Since in G there are no self-loops, we note that all entries on the main diagonal of A are zero.

Intuitively, a "cluster" can be informally defined as a maximally coherent set of vertices, i.e., as a subset $S\subseteq V$ which satisfies both an *internal* criterion (all elements belonging to S should be highly similar to each other) and an *external* one (no larger clusters should contain S as a proper subset). In other words, a cluster should have high internal homogeneity and there should be high inhomogeneity between its elements and those outside. This amounts to saying informally that the weights on the edges within a cluster should be large, and those on the edges connecting the cluster nodes to the external ones should be small.

Now, in an attempt to formally capture this notion, we need some notations and definitions. For a non-empty subset $S\subseteq V$, $i\in S$, and $j\notin S$, we define

$$\phi_S(i,j)=a_{ij}-\frac{1}{|S|}\sum_{k\in S}a_{ik}. \tag{1}$$

where $\|S\|$ denotes the cardinality of S. This quantity measures the (relative) similarity between nodes i and j, with respect to the average similarity between node i and its neighbors in S. Note that $\phi_S(i,j)$ can be either positive or negative. Next, to each vertex $i\in S$

we assign a weight defined (recursively) as follows:

$$w_S(i) = \begin{cases} 1 & \text{if } |S| = 1, \\ \sum_{j\in S\setminus\{i\}} \phi_{S\setminus\{i\}}(j,i) w_{S\setminus\{i\}}(j) & \text{otherwise.} \end{cases} \tag{2}$$

Intuitively, $w_S(i)$ gives us a measure of the overall similarity between vertex i and its neighbors in S, i.e., the vertices of $S\setminus\{i\}$, with respect to the overall similarity among the vertices in $S\setminus\{i\}$. Therefore, a positive $w_S(i)$ indicates that adding i into its neighbors in S will increase the internal coherence of the set, whereas in the presence of a negative value we expect the overall coherence to be decreased. Finally, the total weight of S can be simply defined as

$$W(S) = \sum_{i\in S} w_S(i). \tag{3}$$

A non-empty subset of vertices $S \subseteq V$ such that $W(T) > 0$ for any non-empty $T \subseteq S$, is said to be a *dominant set* if:

1. $w_S(i) > 0$ for all $i\in S$.
2. $w_{S\cup\{i\}}(i) < 0$ for all $i\notin S$.

It is evident from the definition that a dominant set satisfies the two basic properties of a cluster, namely internal coherence and external incoherence. Condition 1 indicates that a dominant set is internally coherent, whereas condition 2 implies that this coherence will be destroyed by the addition of any vertex from outside. In other words, a dominant set is a maximally coherent dataset.

Starting from the perspective of a dominant set as a maximally coherent dataset, we derive a method to extract a dominant set in the following. Naturally, the internal coherency of a cluster can be represented by $x^T Ax$. The clustering problem is then transformed into the following linearly constrained quadratic optimization problem:

$$\max \quad x^T Ax \quad \text{s.t.} \quad x\in\Delta \tag{4}$$

where $\Delta = \{x\in R^n : \sum_i x_i = 1, \text{ and } x_i \geq 0 \text{ for all } i = 1,\ldots,n\}$ is the standard simplex of R^n.

In [24,25] Pavan and Pelillo established a connection between dominant sets and the local solutions of (4). In particular, they showed that if S is a dominant set then its "weighted characteristic vector" x^S, which is the vector of Δ defined as

$$x_i^S = \begin{cases} \dfrac{w_S(i)}{W(s)} & \text{if } i\in S, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

is a strict local solution of (4). Conversely, under mild conditions, it turns out that if x is a strict local solution of (4) then its "support" $S = \{i\in V : x_i > 0\}$ is a dominant set. By virtue of this result, we can find a dominant set by first localizing a solution of (4) with an appropriate continuous optimization technique, and then selecting the support set of that solution. In this sense, we indirectly perform combinatorial optimization via continuous optimization.

A simple and effective optimization algorithm to extract a dominant set is given by the so-called *replicator dynamics* developed and studied in evolutionary game theory

$$x_i^{(t+1)} = x_i^{(t)} \frac{(Ax^{(t)})_i}{x^{(t)'} Ax^{(t)}} \tag{6}$$

for $i = 1,\ldots,n$. In our implementation, however, we used a more efficient algorithm proposed recently in Rota Bulò [27], which has a computational complexity per step that grows linearly in the number of vertices.

After extracting a dominant set, we remove its vertices from the graph and repeat the process until all elements are clustered. Using this "peeling-off" strategy, the number of clusters is automatically determined and the resulted clusters satisfy the constraint of high intra-cluster and low inter-cluster similarity. In other words, with dominant sets clustering, the number of clusters is totally determined by the similarity distribution within the pairwise similarity matrix, instead of defined by users. This property makes dominant set a flexible clustering notion, thereby making it especially attractive for our kernel combination problem, as will be shown in Section 4.

4. Using dominant sets to determine a kernel's accuracy

The SVM is a popular classifier and widely used in object classification. With a kernel matrix and corresponding training labels as input, a SVM classifier partitions the training examples of different classes with as large a margin as possible. From this mechanism, we see that if the training examples of the same class are highly similar to each other and those of different classes are dissimilar, it is likely that the SVM classifier separates training examples of different classes with a large margin and produces a high recognition rate. In other words, the chance of a kernel matrix to produce a high recognition rate is measured by the extent to which it satisfies the high intra-class and low inter-class similarity constraint. This measure reflects the discriminative power of the kernel and will be used in this paper to define the kernel's accuracy. Intuitively, a kernel with a large accuracy has a good chance to produce a high recognition rate and therefore should be assigned a large weight in combination. In the following we analyze how to define the accuracy of a kernel matrix based on dominant sets clustering.

By feeding the dominant sets clustering algorithm with a given kernel matrix (in fact a similarity matrix) we obtain a partition of the training data, where each part corresponds to a cluster. From Section 3 we know that in this partition the number of parts is determined automatically and the parts satisfy the constraint of high intra-part similarity and low inter-part similarity. Noticing that the labels of the training data trivially determine another partition of the training examples where each part corresponds to a single class, we see here that the resemblance between the two partitions determines to which extent a kernel matrix satisfies the constraint of high intra-class and low inter-class similarity, and this, in turn, determines the chance of a kernel to classify accurately, i.e., the kernel's accuracy.

Ideally the two partitions coincide and each class corresponds to a dominant set (see an example illustration in Fig. 1(a)). In this case the kernel matrix strictly satisfies the constraint of high intra-class and low inter-class similarity and has a good chance to produce a high recognition rate. We define the kernel accuracy in such cases to be 1. Note that this is not to say that the kernel matrix will produce a 100% recognition rate, but that the potential of similarity distribution has been fully explored to obtain an accurate classification in the given framework.

Obviously the ideal case rarely occurs in practice. In fact, the number of dominant sets is usually larger than the number of classes. Some dominant sets may contain subsets of only one class and other dominant sets may contain subsets of multiple classes. As a result, the dominant-set-based and the training-label-based partitions might have substantial intersection, as illustrated in Fig. 1(b). If all dominant sets are of single-class, the constraint of low inter-class similarity is still satisfied. Although the high intra-class similarity constraint is not strictly satisfied in this case, we note that the low inter-class similarity is still easy for a SVM to classify correctly. Therefore we still define the accuracy of such a kernel matrix to be 1.

The existence of multi-class dominant sets implies that some training examples of different classes are very similar to each other and this presents difficulty for a SVM classifier. It is natural to see that the percentage of examples in all multi-class dominant sets should be inversely proportional to the kernel's accuracy. Within

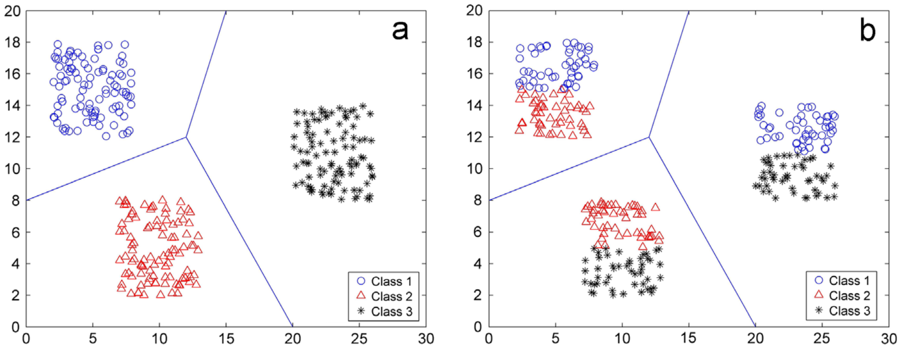

Fig. 1. An illustration of the relationship between the partition by training labels and the one by dominant sets clustering. In the figures the blue lines indicate the partition by dominant sets clustering, and the partition by training labels are denoted by different colors and symbols. (a) Ideal case. (b) Practical case. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

a multi-class dominant set, the shares of different classes also affect the accuracy. If the share of one class is very large and the shares of other classes are very small, the multi-class dominant set looks similar to a single-class one, and its negative effect on the kernel accuracy is relatively small. On the other hand, if all classes are distributed uniformly in a dominant set, all classes are equally involved in the difficult partition and none of them can be ignored. This is expected to lead to a notable decrease in kernel accuracy.

From the above analysis we see that a kernel's accuracy should be inversely related to the amount of entropy within dominant sets. Therefore we define the kernel accuracy in the following way. Suppose that we have found N dominant sets, and we have C classes. We define the entropy of each dominant set i ($i=1,\ldots,N$) as follows:

$$H_i = -\sum_{j=1}^{C} \frac{n_{ij}}{N_i} \log \frac{n_{ij}}{N_i} \tag{7}$$

where n_{ij} is the number of elements in dominant set i which belong to class j, and N_i is the overall number of elements in dominant set i. H_i will be 0 if and only if within dominant set i all items are assigned a single label, and it attains its maximum value (namely, $\log C$) when all classes get equal share. Hence, by dividing H_i by $\log C$ we get a measure between 0 and 1, where 0 corresponds to the ideal case where dominant set i gets a unique label, and 1 indicates the opposite extreme, i.e., uniform distribution of all classes within dominant set i.

Finally, we define an overall accuracy measure of a kernel K in the following way:

$$w_{dset}(K) = \frac{1}{N} \sum_{i=1}^{N} \left(1 - \frac{H_i}{\log C}\right). \tag{8}$$

Obviously w_{dset} equals 1 in the ideal case where all dominant sets are of single-class, and becomes 0 when all dominant sets are shared equally by all classes. For each kernel, we calculate w_{dset} and use it to determine the kernel's weight in combination. We tested different weighting methods, including $w_{dset}^k, k=1,\ldots,6$ and $\exp(w_{dset})$, and finally selected w_{dset}^3 as the weight as it produced the best overall performance.

To sum up, our method to compute the weight of a kernel in combination involves the following steps:

1. Do dominant sets clustering with the kernel matrix as input and obtain N dominant sets.
2. Calculate $H_i, i=1,\ldots,N$ of each dominant set with (7).
3. Calculate w_{dset} of the kernel with (8).
4. Use w_{dset}^3 as the kernel weight in combination.

Our feature combination method is intuitive as it assigns a meaningful weight to each feature in combination, and simple in that the weights of features are computed separately. In the case of a large set of features are combined, the latter property implies a much smaller memory consumption than optimization based methods. While our method involves only simple equations, it does provide a novel approach to estimate the discriminative power of a kernel and is shown to be effective in experiments.

In SVM classification a multi-class classifier must be extended from the basic two-class one by means of one-versus-one or one-versus-all training. Whereas in our feature combination method the weights of features are determined totally by the resemblance between the partition by training labels and the one by dominant sets clustering, and the weights calculation has nothing to do with the number of classes. This means that the processing steps of both two-class and multi-class cases are exactly the same.

In our method we compare the partition by training labels with the one by dominant sets clustering to estimate to which extent the kernel matrix satisfies the constraint of high intra-class and low inter-class similarity, and then the kernel accuracy. The key requirement of the clustering algorithm used here is that the clusters obtained satisfy the constraint of high intra-cluster and low inter-class similarity. While any clustering algorithm can be said to meet this requirement, our method requires the number of clusters to be determined appropriately, i.e., the degree of "high" and "low" in the constraint of high intra-cluster and low inter-cluster similarity to be decided automatically. Otherwise, an inappropriate selection of the number of clusters may make our method totally useless. If the number of clusters is too large, the examples that are very similar to each other may be divided into different clusters and make most of clusters be of single-class. At the other extreme, the number of clusters is very small and the kernel accuracies are determined mostly by the proportion of different classes in the training dataset. Obviously both cases are not what we expect. Dominant sets clustering extracts clusters with high intra-cluster similarity sequentially and determines the number of clusters automatically, and this is why we choose this clustering method in our feature combination method.

5. Experimental results

We tested our kernel weighting method in feature combination with SVM classification experiments on six diverse datasets. In all experiments the SVM penalty parameters were fixed to be 1000. In the case of multi-class datasets, the SVMs were trained in the one-versus-all mode. When distances were used to build kernels, the transformation used was 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. With all the six datasets, the experimental setups and accuracy measures were selected to be same as in the literature used for comparison. Unless otherwise stated, the experiments were repeated 10 times with different training-testing splits and the average recognition rates are reported. Since cross-validation (CV) is usually used to evaluate the powerfulness of a kernel, in comparisons we also included this weighting method together with the popular MKL method, where we use the Simple MKL toolbox as the solver for MKL.

The whole experimental procedures are demonstrated in Fig. 2.

5.1. Recognition rates

1. Event-8 and Scene-15 datasets

The Event-8 dataset [14] consists of images from eight sport event categories: badminton, bocce, croquet, polo, rock climbing, rowing, sailing and snowboarding. Each category has 130–250 images. Besides classifying events from static images, the dataset presents some other challenges for classification, including cluttered and diverse backgrounds, and various poses, sizes and views of foreground objects. See Fig. 3 for sample images and a brief description. Following the setup in Jia and Fei-Fei [14], we randomly selected 70 images per class as training and another 60 images as testing, and report the 8-class overall recognition rate.

The Scene-15 dataset [23,9,17] contains images from 15 categories with 200–400 images in each category (see Fig. 4 for some example images). We followed the experimental setup in Lazebnik et al. [17], i.e., 100 randomly selected images per class as training and all the others as testing, and report the mean recognition rate per class.

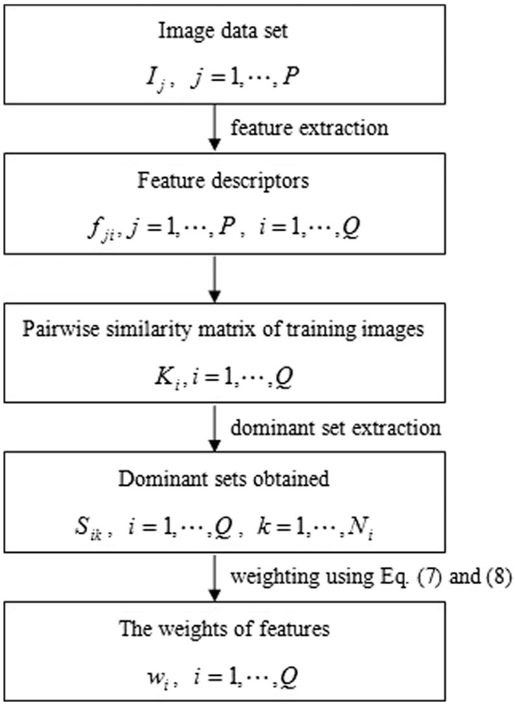

Fig. 2. The flowchart of feature weighting in our method. In the denotations, P is the number of images in the dataset, Q is the number of kernels used in combination, and N_i is the number of dominant sets extracted from the kernel matrix K_i.

For both datasets, we used the following features to build the kernels.

PHOG Shape Descriptor. Oriented (20 bins) and unoriented (40 bins) PHOG descriptors [7,4] were constructed from level 0 to 3. Different from the implementation in Bosch et al. [4], in this paper the descriptor of level L was just composed of the descriptors of its 2^L windows, with no addition of those from lower levels. We denote the two kinds of features as *hog*180-*L* and *hog*360-*L* respectively.

Bag of Visual Words. We used SIFT descriptors [19] on 16×16 patches with spacing of eight pixels to build a 500-bin vocabulary. The descriptors were extracted in gray (128-bin), HSV (384-bin) and CIE-Lab (384-bin) spaces. The visual word histograms were built in a pyramid from level 0 to 2. The three kinds of features are denoted by *gvw-L*, *hvw-L* and *lvw-L* respectively.

Locally Binary Patterns. The basic locally binary patterns (LBP) [22] were extracted and clustered to create a descriptor for one image. The descriptor length is 256 and we built the descriptors of level 0 to 2 (*lbp*-0 to *lbp*-2).

Gray Value Histogram. We also used the 64-bin gray value histograms from level 0 to 3 (*hoi*-0 to *hoi*-3).

Gist Descriptor. The gist descriptors [23] were extracted in a pyramid from level 0 to 1 (*gist*-0 to *gist*-1).

Self-similarity Descriptor. Self-similarity descriptors [29] of 30 dimensions (10 orientations and three radial bins) were extracted and quantized into a vocabulary of 500 bins. The histograms were built from level 0 to 2 (*ssm*-0 to *ssm*-2).

Gabor and RFS filters. We used two texture features: Gabor and RFS filters [36] to build histograms (500 bins) from level 0 to 2. The two kinds of features are denoted as *gab-L* and *txn-L* respectively.

We used these features to build kernels with χ^2 distance and obtained 35 kernels for Event-8. Note that Scene-15 only contains gray-level images and we only extracted bag of visual word features in gray space, therefore getting only 29 kernels. Here the selection of χ^2 distance in building kernels is based on our previous work in Hou et al. [13], where χ^2 based kernel was shown to outperform some other kernels including linear, Gaussian, histogram intersection kernel [2], Euclidean and $l1$ distance based kernels. Besides, our selection is also supported by Gehler and Nowozin [11]. The classification results and comparison with the literature are shown in Table 1. To demonstrate the effects of individual features on combination performance, we also show the performance comparison of individual features with combination in Figs. 5 and 6.

From Table 1 we observe that while average combination performs better than the best individual feature, our weighted combination further improves the results considerably for both datasets. Our recognition rate for Event-8 is 89.8%, which is substantially better than 73.4% in Jia and Fei-Fei [14] and 84.2% in Wu and Rehg [39]. For Scene-15, our best result is 87.0%, outperforming the state-of-the-art result of 86.7% reported in Bo et al. [3] and other results. In both cases our weighting method performs better than CV and MKL weighting method, and this further confirms the effectiveness of our method. These results

Fig. 3. Sample images of the Event-8 dataset. Two images per category are displayed with four categories in one row. From left to right and top to bottom, the categories are badminton, bocce, croquet, polo, rock climbing, rowing, sailing and snowboarding.

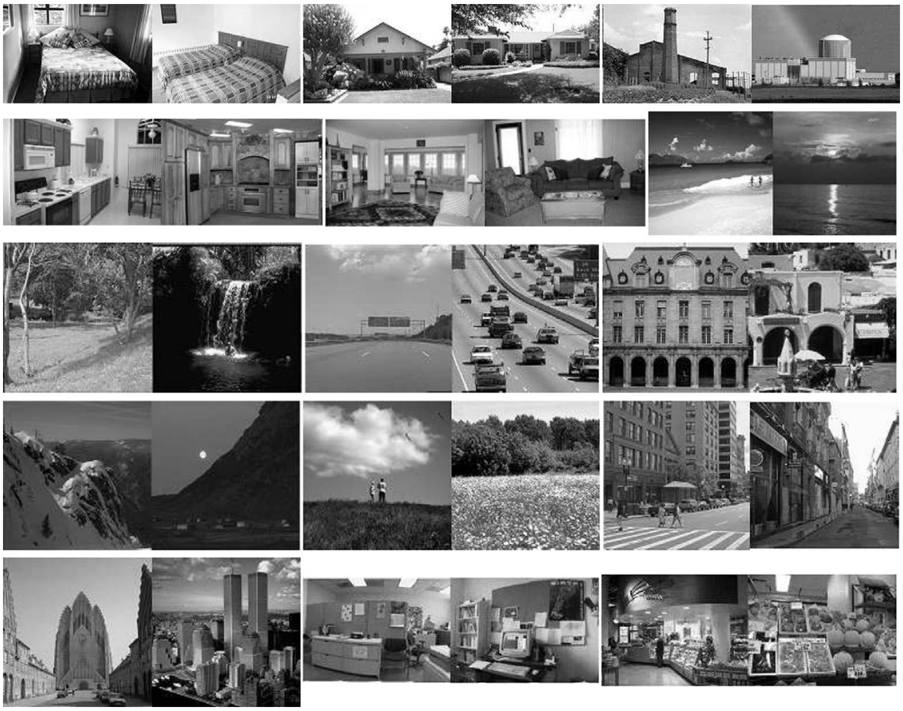

Fig. 4. Sample images of the Scene-15 dataset. Two images per category are displayed with three categories in one row. From left to right and top to bottom, the categories are bedroom, suburb, industrial, kitchen, living room, coast, forest, highway, inside city, mountain, open country, street, tall building, office and store.

Table 1
Event-8 and Scene-15 recognition rates and comparison.

Event-8		Scene-15	
Method	Accuracy	Method	Accuracy
Best single	85.0 ± 0.9	Best single	79.6 ± 0.7
Average	86.0 ± 1.5	Average	86.1 ± 0.5
CV weight	87.6 ± 1.3	CV weight	86.4 ± 0.6
MKL	76.9 ± 1.9	MKL	76.5 ± 0.6
This paper	**89.8 ± 0.9**	This paper	**87.0 ± 0.2**
[39]	84.2 ± 1.0	[3]	86.7 ± 0.4
[14]	73.4	[39]	84.1 ± 0.5
		[17]	81.4 ± 0.5
		[5]	73.4 ± 1.0

imply that with relatively simple features and kernel functions, our combination method produces a significant improvement in classification performance.

It is evident from Figs. 5 and 6 that when the performance variance of individual features is small, the advantage of our weighting method over average combination is a little small. This is consistent with the trend observed in Gehler and Nowozin [11], i.e., if all participating kernels are of similar discriminative power, current learning methods show little advantage over average combination.

2. Flower-17 and Caltech-101 datasets

Oxford Flower-17 dataset [20] is composed of flower images of 17 categories with 80 images in each category. See Fig. 7 for example images. Nilsback and Zisserman [20,21] provided seven carefully designed kernels for this dataset. Besides these seven kernels, we added the same 35 kernels as with Event-8 into combination in experiments. We report the overall recognition rate and comparison with the literature in Table 2, and the performance comparison of individual features with combination in Fig. 8.

On the Flower-17 dataset, our method produced a 90.4% recognition rate, which, to our knowledge, is the best result achieved on this dataset to date, better than the results obtained with MKL in Nilsback and Zisserman [21] and Varma and Ray [35], and with LP-β method in Gehler and Nowozin [11]. This indicates that with a relatively small number of kernels, our simple weighting scheme can be used to produce superior classification performance.

Comparing Fig. 8 with Figs. 5 and 6, we found that when the performance of individual features varies significantly (in the case of Flower-17), our weighting scheme produces a significant performance improvement. In fact, the standard deviation of recognition rates of individual classifiers is 16.33 for Flower-17, compared to the 12.03 and 13.37 for Event-8 and Scene-15, respectively. This means that our method, just like MKL and LP-β, has the property of suppressing the negative effects of weak features in combination.

With the Caltech-101 dataset [8], we followed the widely adopted experimental setup. Specifically, for each of the 102 classes, we randomly selected 5, 10, 15, 20, 25, 30 images for training and up to 50 images in the remaining for testing. For comparison, we adopted the same features as in Gehler and Nowozin [11] to build kernel matrices and used mean recognition rate per class as accuracy measure. In total 39 kernels from various features were used in combination. The experimental results and comparisons are shown in Fig. 9. Since we used exactly the same data and experimental setup as in Gehler and Nowozin [11], we report the results of MKL, LP-average and LP-β from [11] in comparison. In this experiment our weighting scheme is outperformed by the LP-β algorithm, but it compares favorably with all the other combination methods, including LP-average and MKL.

These comparisons indicate that with proper definition, our simple and intuitive kernel accuracy weighting scheme can be as powerful as more sophisticated optimization methods. Noticing its good performance and advantage in computation complexity and memory consumption, we think this kind of methods should also be used as a benchmark combination method, just as the average combination.

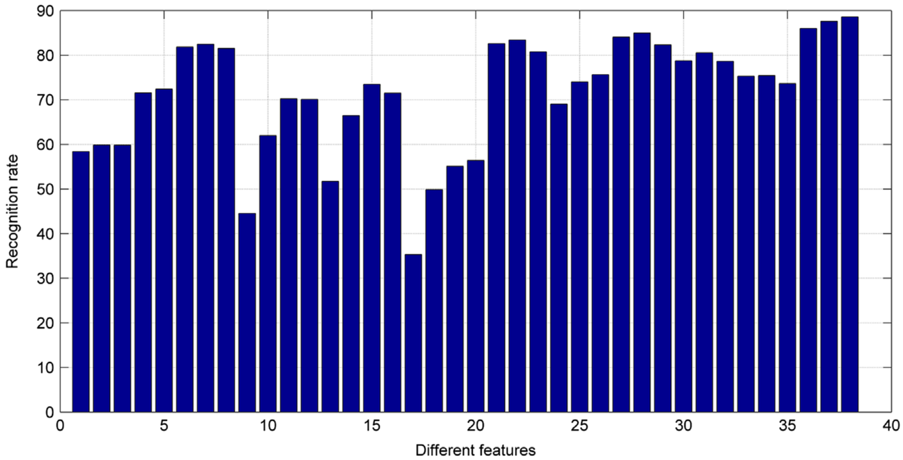

Fig. 5. Performance comparison of individual features and combination with Event-8 dataset. The left 35 bars are of individual features, in the order of *gab*-0, *gab*-1, *gab*-2, *gist*-0, *gist*-1, *gvw*-0, *gvw*-1, *gvw*-2, *hog*180-0, *hog*180-1, *hog*180-2, *hog*180-3, *hog*360-0, *hog*360-1, *hog*360-2, *hog*360-3, *hoi*-0, *hoi*-1, *hoi*-2, *hoi*-3, *hvw*-0, *hvw*-1, *hvw*-2, *lbp*-0, *lbp*-1, *lbp*-2, *lvw*-0, *lvw*-1, *lvw*-2, *ssm*-0, *ssm*-1, *ssm*-2, *txn*-0, *txn*-1 and *txn*-2, and the rightmost three bars are of average, CV weight and this paper's method respectively.

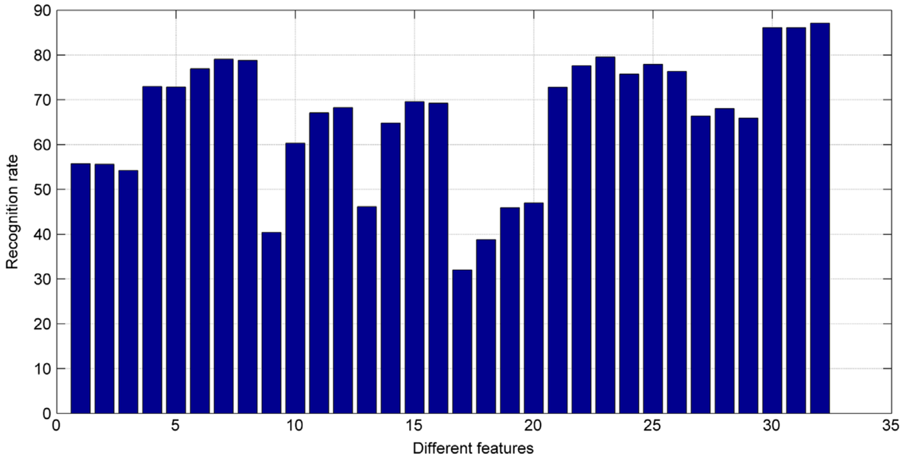

Fig. 6. Performance comparison of individual features and combination with Scene-15 dataset. The left 29 bars are of individual features, in the same order as Event-8 excluding *hvw*-0, *hvw*-1, *hvw*-2, *lvw*-0, *lvw*-1 and *lvw*-2, and the rightmost three bars are of average, CV weight and this paper's method respectively.

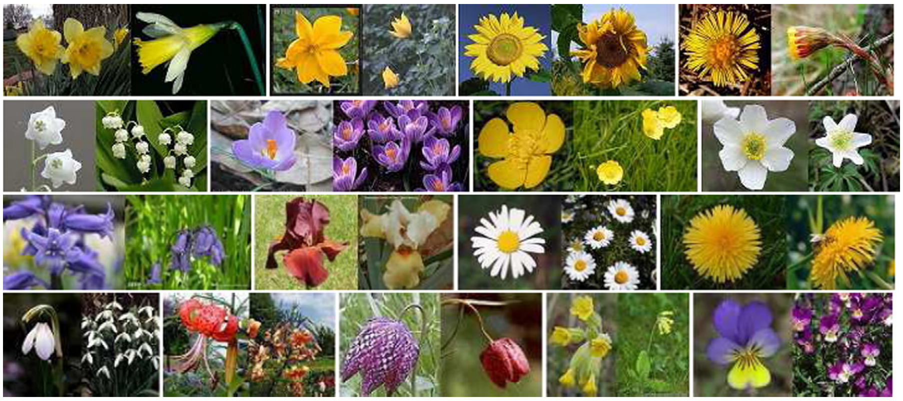

Fig. 7. Sample images of the Flower-17 dataset. Two images per category are displayed with four categories per row in the top three rows and five categories in bottom row.

3. TMA and MRI datasets

Finally, we tested our method on two medical image datasets. The first one consists of 1272 tissue micro arrays (TMA) images of renal cell carcinoma. All the images are of size 80×80 pixels centered at labeled cell nuclei, with 890 images labeled as benign and 382 as malignant. The details of the dataset can be found in Schuffler et al. [28] and some example images are shown in Fig. 10. Here we used the same features as the ones employed with the Scene-15 dataset to build kernels and the reported results are expressed in terms of recognition rate with 10-fold cross-validation.

The second dataset contains brain magnetic resonance images (MRIs) of 64 schizophrenia patients and 60 healthy controls. For each subject, 14 regions of interest (ROIs) were manually segmented from MRI and then cut into a number of slices (Fig. 10). With normalized gray value histograms from these ROIs and their pdf's as features, 13 distance measures were obtained to build 182 kernels in total. See [33] for details of this dataset. The classification was conducted with the leave-one-out mode and the results are reported as overall recognition rate in Table 3.

With both datasets, our combination method shows its superiority over best single feature and average methods. On TMA dataset, [28] built kernels based on the segmentation of nucleus from surrounding tissue and obtained a best recognition rate of 83%. While in our experiments we applied the combination to the original unsegmented images and achieved 87.1%. On MRI dataset, our method also outperformed the dissimilarity matrices combination in Ulas et al. [33], where the reciprocal of the average dissimilarity value was used as the weight.

5.2. Computation efficiency

Besides classification accuracy, the computation cost is another important measure used to evaluate feature combination methods, especially when a large number of features are involved. Since in experiments the difference of various feature combination methods lies only in the kernel weighting step, we report the running time in the kernel weighting steps of different methods with Caltech-101 in Table 4 and with other datasets in Table 5. All experiments were run on a computer equipped with 2 AMD Athlon 3 GHz CPUs and 8 GB RAM.

From Tables 4 and 5 we observe that in computing the weights of kernels, our method is much faster than CV and MKL. In our method, the majority of computation is devoted to the dominant sets clustering in each participating kernel matrix. With the traditional game dynamics, e.g., replicator dynamics, this computation may be quite time consuming as these game dynamics are usually designed for sceneries with a small number of players and computationally inefficient. However, in our implementation we use the infection and immunization dynamics [27] and are able to finish the clustering efficiently. With CV based weighting method, classification in the cross-validation mode must be conducted to evaluate the discriminative power of kernels. This means that the computation load is determined by the classifier adopted, kernel matrix size and the number of folds in cross-validation, etc. In the case that a large kernel matrix is involved, the cross-validation classification process may be rather computationally expensive. As to MKL, the joint optimization of kernel weights and SVM classifier parameters is always time consuming, especially for large kernels, as shown by the case of Caltech-101.

6. Discussion

In all the six sets of experiments of Section 5, our method consistently performed better than the best single classifiers. This confirms that our method is effective in exploring the potential of the combination of multiple complementary features to produce better performance. While average combination also produces better results than the best single classifier, our method evaluates the powerfulness of participating features and assigns different weights to different kernels, and thus outperforms average combination. In comparison with the other two trained weighting

Table 2
Flower-17 recognition rates and comparison.

Method	Accuracy
Best single	76.8 ± 2.0
Average	85.3 ± 2.1
CV weight	87.5 ± 1.9
MKL	84.6 ± 1.5
This paper	**90.4** ± **1.1**
[21]	88.3 ± 0.3
[11]	85.5 ± 3.0
[35]	82.6 ± 0.3

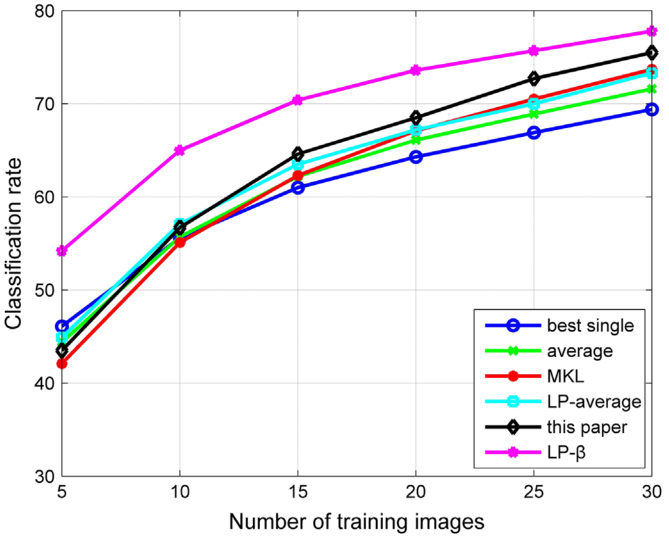

Fig. 9. Caltech-101 recognition rates and comparison.

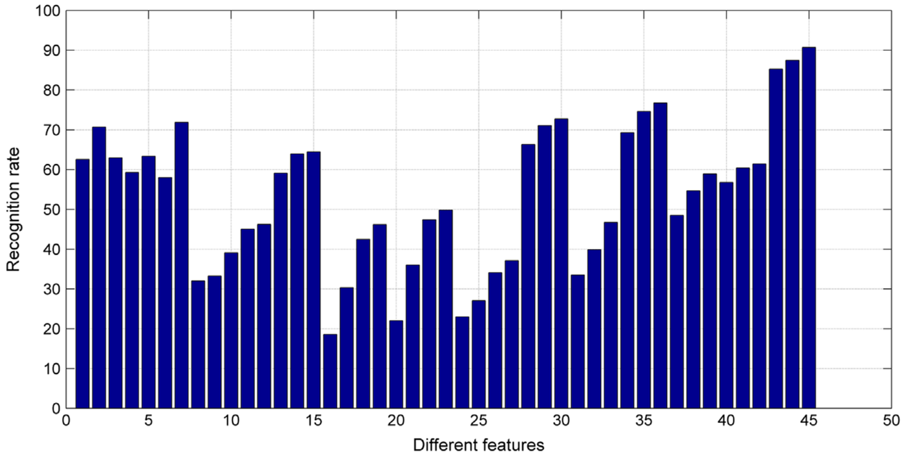

Fig. 8. Performance comparison of individual features and combination with Flower-17 dataset. The leftmost seven bars are of the seven kernel matrices from [20,21], and the middle 35 are in the same order as Event-8, and the rightmost three are of average, CV weight and this paper's method respectively.

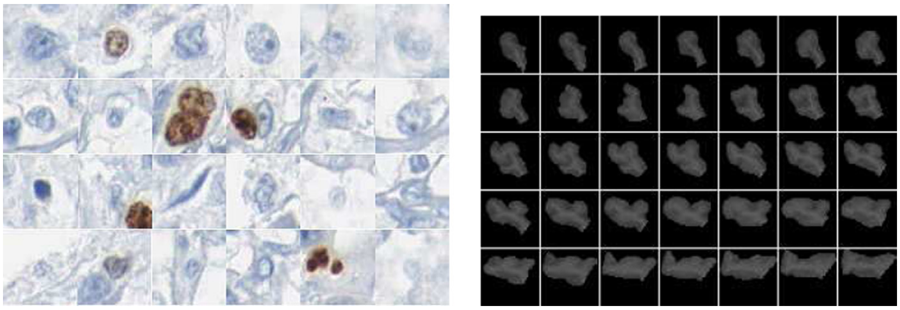

Fig. 10. Sample images of the TMA (left) and Brain MRI (right) dataset. In TMA dataset, the top two rows are of malignant and the bottom two rows are of benign. In Brain MRI dataset is 35 slices of one ROI.

Table 3
TMA and brain MRI datasets recognition rates and comparison.

TMA		Brain MRI	
Method	Accuracy	Method	Accuracy
Best single	82.1 ± 0.3	Best single	70.1
Average	85.4 ± 0.7	Average	76.6
CV	86.5 ± 0.3	CV	75.0
MKL	69.9 ± 0.0	MKL	63.7
This paper	**87.1 ± 0.4**	This paper	**79.8**
[28]	83.0	[33]	79.0

Table 4
The running time comparison of different combination methods on Caltech-101 with varying number of training examples.

Method	5	10	15	20	25	30
CV	105.0	419.4	883.3	1460.4	722.6	3259.5
MKL	1848.8	7421.9	8353.4	11,022.3	21,650.8	187,800.5
This paper	13.9	61.9	163.7	298.4	606.9	917.0

Table 5
The running time comparison of different combination methods on five other datasets.

Method	Event-8	Scene-15	Flower-17	TMA	MRI
CV	22.3	107.9	71.7	1356.3	757.4
MKL	172.3	1007.7	1377.8	1398.0	166,328.3
This paper	16.5	70.1	34.1	386.0	615.4

methods, namely CV and MKL, our method produces better performance with much smaller computation consumption. The reason that our method performs better than the CV method lies probably in the mechanism of cross-validation, namely one part of the data as training and the other one as testing. This kind of powerfulness evaluation method seems always not able to utilize the training kernel matrix as a whole, and this probably harms the estimation precision of kernel accuracy. As to MKL, it rarely outperforms average combination, and performs much worse than even the best single in some cases. This behavior may be due to the sophisticated and time-consuming optimization which causes some kind of over-fit in the training model. To sum up, in experiments our method outperformed the best single classifier, average combination, CV weighting, MKL and the literature by several or more percent in recognition rate in most of the cases. Considering that the datasets used in our experiments are mostly well-known and widely investigated, we think this improvement is impressive enough.

While our proposed method has been demonstrated to be effective in experiments on six datasets, it also leaves some space for further improvement. The key idea of our method is to compare the partitions by training labels and by clustering within a kernel matrix to evaluate the kernel's accuracy. In order for the method to be effective, the partition by clustering must satisfy the constraint of high intra-part and low inter-part similarity. In this paper we select dominant sets clustering to partition the kernel matrix due to its ability to determine the number of clusters automatically and produce the clusters we need. Although the superior performance of dominant sets clustering has been validated in our experiments, we also note some limitations in the clustering procedures. First, current dominant sets clustering algorithms adopt a “peeling-off” strategy and different clusters are actually extracted from different similarity matrices. This results in the problem that the degrees of high intra-cluster similarity of these clusters may be different. In fact, we observed that those clusters extracted later from smaller similarity matrices tend to have a relatively low intra-cluster similarity compared with those from larger similarity matrices. This behavior is obviously not what we expect and will influence the performance of our method. In the next step we will work on new dominant sets clustering strategies, possibly based on the soft clustering method proposed in Torsello et al. [31]. Second, in our experiments, we found that dominant sets clustering tends to generate more clusters than expected. In other words, this clustering method sets a somewhat too strict requirement of the high intra-cluster similarity. While this property may be good for other applications, it departs a little from our expectation as required by our method. Therefore we plan to explore the possibility to relax the intra-cluster similarity requirement in dominant sets clustering. Besides the dominant sets clustering method, the kernel accuracy definition from the comparison criterion of two partitions also affects the combination performance. In this paper we use the concept of entropy within each dominant set to evaluate the discriminative power of kernels. It is possible to design a more effective kernel accuracy measure based on the analysis of SVM classification and kernel combination mechanism.

7. Conclusion

In this paper we have proposed a simple yet effective kernel weighting method for feature combination in object classification based on dominant sets clustering. Starting from the intuition that better-performing kernels should be given larger weights in combination, we analyzed the correlation between the SVM classification mechanism and dominant sets clustering. As a result, we proposed a novel method to evaluate the discriminative power of kernels. Specifically, we partition the training examples by dominant sets clustering and by their training labels. The resemblance of these two partitions is found to reflect the possibility of obtaining a high recognition rate with this kernel, and thus used to determine the weight of the kernel in combination. We tested the proposed method with extensive experiments on several datasets of diverse object types and reported systematic improvement over benchmark combination methods. While our method is simple, it performs comparably to more sophisticated, state-of-the-art methods with much smaller memory and computation consumption.

Conflict of interest statement

None declared.

Acknowledgements

This work was done while the first author was working in the Computer Vision and Pattern Recognition group at the University of Venice as a postdoctoral fellow. We acknowledge the financial support of the Future and Emerging Technology (FET) Programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open project SIMBAD Grant no. 213250.

References

[1] F.R. Bach, Exploring large feature spaces with hierarchical multiple kernel learning, in: Advances in Neural Information Processing Systems, 2008, pp. 105–112.

[2] A. Barla, F. Odone, A. Verri, Histogram intersection kernel for image classification, in: International Conference on Image Processing, 2003, pp. 513–516.

[3] L. Bo, X. Ren, D. Fox, Kernel descriptors for visual recognition, in: Advances in Neural Information Processing Systems, 2003, pp. 244–252.

[4] A. Bosch, A. Zisserman, X. Munoz, Representing shape with a spatial pyramid kernel, in: ACM International Conference on Image and Video Retrieval, 2007, pp. 401–408.

[5] H. Cai, F. Yan, K. Mikolajczyk, Learning weights for codebook in image classification and retrieval, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2010, pp. 2320–2327.

[6] C. Cortes, M. Mohri, A. Rostamizadeh, Learning non-linear combinations of kernels, in: Advances in Neural Information Processing Systems, 2009, pp. 396–404.

[7] N. Dalal, B. Triggs, Histogram of oriented gradients for human detection, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2005, pp. 886–893.

[8] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, in: CVPR Workshop on Generative-Model Based Vision, 2004, p. 178.

[9] L. Fei-Fei, P. Perona, A Bayesian hierarchical model for learning natural scene categories, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2005, pp. 524–531.

[10] F. Frommlet, Tag SNP selection based on clustering according to dominant sets found using replicator dynamics, Advances in Data Analysis and Classification 4 (2010) 65–83.

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

[12] R. Hamid, S. Maddi, A.Y. Johnson, A.F. Bobick, I.A. Essa, C. Isbell, A novel sequence representation for unsupervised analysis of human activities, Artificial Intelligence 173 (2009) 1221–1244.

[13] J. Hou, B.P. Zhang, N.M. Qi, Y. Yang, Evaluating feature combination in object classification, in: International Symposium on Visual Computing, 2011, pp. 597–606.

[14] L.L. Jia, L. Fei-Fei, What, where and who? Classifying events by scene and object recognition, in: IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[15] A. Kumar, C. Sminchisescu, Support kernel machines for object recognition, in: IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[16] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, M. Jordan, Learning the kernel matrix with semidefinite programming, Journal of Machine Learning Research 5 (2004) 27–72.

[17] S. Lazebnik, C. Schmid, 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, pp. 2169–2178.

[18] Y.Y. Lin, T.L. Liu, C.S. Fuh, Local ensemble kernel learning for object category recognition, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[19] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–110.

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

[21] M.E. Nilsback, 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.

[22] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transaction on Pattern Analsis and Machine Intelligence 24 (2002) 971–987.

[23] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, International Journal of Computer Vision 42 (2001) 145–175.

[24] M. Pavan, M. Pelillo, A graph-theoretic approach to clustering and segmentation, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2003, pp. 145–152.

[25] M. Pavan, M. Pelillo, Dominant sets and pairwise clustering, IEEE Transaction on Pattern Analsis and Machine Intelligence 29 (2007) 167–172.

[26] S. Rota Bulò, M. Pelillo, A game-theoretic approach to hypergraph clustering, in: Advances in Neural Information Processing Systems, 2009, pp. 1571–1579.

[27] S. Rota Bulò, M. Pelillo, I.M. Bomze, Graph-based quadratic optimization: a fast evolutionary approach, Computer Vision and Image Understanding 115 (2011) 984–995.

[28] P. Schuffler, T. Fuchs, C. Ong, V. Roth, J. Buhmann, Computational TMA analysis and cell nucleus classification of renal cell carcinoma, in: 32 Annual Symposium of the German Pattern Recognition Society, 2010, pp. 202–211.

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

[30] A. Torsello, S. Rota Bulò, M. Pelillo, Grouping with asymmetric affinities: a game-theoretic perspective, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2006, pp. 292–299.

[31] A. Torsello, S. Rota Bulò, M. Pelillo, Beyond partitions: allowing overlapping groups in pairwise clustering, in: International Conference on Pattern Recognition, 2008, pp. 1–4.

[32] S. Tulyakov, S. Jaeger, V. Govindaraju, D. Doermann, Review of classifier combination methods, in: S. Marinai, H. Fujisawa (Eds.), Machine Learning in Document Analysis and Recognition: Studies in Computational Intelligence, vol. 90, Springer, Berlin, 2008, pp. 361–386.

[33] A. Ulas, R. Duin, U. Castellani, M. Loog, M. Bicego, V. Murino, M. Bellani, S. Cerruti, M. Tansella, P. Brambilla, Dissimilarity-based detection of schizophrenia, in: ICPR Workshop on Pattern Recognition Challenges in fMRI Neuroimaging, 2010, pp. 32–35.

[34] M. Varma, B.R. Babu, More generality in efficient multiple kernel learning, in: International Conference on Machine Learning, 2009, pp. 1065–1072.

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

[36] M. Varma, A. Zisserman, A statistical approach to texture classification from single images, Image and Vision Computing 62 (2005) 61–81.

[37] S.V.N. Vishwanathan, Z. Sun, N.T. Ampornpunt, M. Varma, Multiple kernel learning and the SMO algorithm, in: Advances in Neural Information Processing Systems, 2010, pp. 2361–2369.

[38] M. Wang, Z.L. Ye, Y. Wang, S.X. Wang, Dominant sets clustering for image retrieval, Signal Processing 88 (2008) 2843–2849.

[39] J.X. Wu, 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.

[40] J.J. Yang, Y.N. Li, Y.H. Tian, L.Y. Duan, W. Gao, Group-sensitive multiple kernel learning for object categorization, in: IEEE International Conference on Computer Vision, 2009, pp. 436–443.

[41] X. Yang, H. Liu, L.J. Latecki, Contour-based object detection as dominant set computation, Pattern Recognition 45 (2012) 1927–1936.

Jian Hou received his PhD degree in 2007 from Harbin Institute of Technology, China. From 2007 to 2010, he worked as a postdoctoral researcher in National University of Singapore, Singapore and University of Venice, Italy. In 2011, he was with Ningbo Institute of Materials Technology and Engineering of Chinese Academy of Sciences and Xuchang University, China. He joined the School of Information Science and Technology at Bohai University, China in 2012, where he is currently an associate professor. His research interests include computer vision, pattern recognition and machine learning. He is a member of IEEE.

Marcello Pelillo joined the faculty of the University of Bari, Italy, as an assistant professor of computer science in 1991. Since 1995, he has been with the University of Venice, Italy, where he is currently a Full Professor of Computer Science. He leads the Computer Vision and Pattern Recognition Group and has served from 2004 to 2010 as the Chair

of the board of studies of the Computer Science School. He held visiting research positions at Yale University, the University College London, McGill University, the University of Vienna, York University (UK), and the National ICT Australia (NICTA). He has published more than 130 technical papers in refereed journals, handbooks, and conference proceedings in the areas of computer vision, pattern recognition and neural computation. He serves (or has served) on the editorial board for the journals *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *Pattern Recognition*, and is regularly on the program committees of the major international conferences and workshops of his fields. In 1997, he co-established a new series of international conferences devoted to energy minimization methods in computer vision and pattern recognition (EMMCVPR) which has now reached the eighth edition. He is (or has been) scientific coordinator of several research projects, including SIMBAD, an EU-FP7 project devoted to similarity-based pattern analysis and recognition. Prof. Pelillo is a Fellow of the IAPR and a Fellow of the IEEE.