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 $V=\{1,\ldots,n\}$ 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 $i\in S$