Pattern Recognition 70 (2017) 104–111

Contents lists available at ScienceDirect

Pattern Recognition

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

Multiple kernel learning with hybrid kernel alignment maximization

Yueqing Wang[a,*], Xinwang Liu[a], Yong Dou[a], Qi Lv[a], Yao Lu[b]

[a] National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha, China
[b] College of Computer, National University of Defense Technology, Changsha, China

ARTICLE INFO

Article history:
Received 11 July 2016
Revised 20 March 2017
Accepted 7 May 2017
Available online 8 May 2017

MSC:
00-01
99-00

Keywords:
Multiple kernel learning
Local kernel alignment
Optimization

ABSTRACT

Two-stage multiple kernel learning (MKL) algorithms have been extensively researched in recent years due to their high efficiency and effectiveness. Previous works have attempted to optimize the combination coefficients by maximizing the centralized kernel alignment between the combined kernel and the ideal kernel. Though demonstrating previous promising performance, we observe that these algorithms may suffer from the approaching in calculating the alignment. In particular, we observe that the local information should be incorporated when computing the kernel alignment, which is beneficial to further improve the classification performance. To this end, we first define the local kernel alignment based on centralized kernel alignment. A new kernel alignment that combines the global and local information of base kernels is then developed. After that, we propose an alternative algorithm with proved convergence to identify the multiple kernel coefficients. Intensive experimental results show that the performance of the proposed algorithm is superior to those of existing MKL algorithms.

1. Introduction

In recent years, multiple kernel learning (MKL) has been extensively studied in machine learning community [1–9]. Different from traditional kernel methods where the kernel selection is left to users, MKL algorithms only require users to pre-specify a set of base kernels, and automatically learn the optimal coefficients for classification tasks. Existing MKL algorithms can roughly be grouped into two categories in terms of the training methodology. The first category is one-stage algorithms [10–13], which simultaneously learn the optimal kernel combination parameters and the structural parameters of a classifier. Differently, the second category is two-stage algorithms [1,14–18]. They first learn an optimal kernel $\mathbf{K}$ according to a certain criteria, and then apply the learned optimal kernel $\mathbf{K}$ into a standard kernel-based algorithm such as support vector machine (SVM). As a typical representation of the second category, kernel target alignment algorithm was first proposed in [19,20]. It learns the optimal kernel coefficients by maximizing centered kernel alignment between the combined kernel and the target kernel. Compared with one-stage algorithms, two-stage algorithms exhibit the following advantages: i) they consume less computational cost while achieving comparable or even better performance; ii) they are more flexible since the learned optimal kernel can be used directly in both classification and regression tasks.

Nevertheless, although the kernel alignment bears these good properties, we observe that it is calculated in a global way, which: i) rigidly forces closer and further sample pairs to be equally aligned to the same similarity, and neglects the intra-class variation of samples; and ii) is inconsistent with a well-established concept that the similarity evaluated for two further samples in a high dimensional space is less reliable due to the presence of underlying manifold structure. As a consequence, maximizing global kernel alignment could make these pre-specified kernels less effectively utilized, and in turn adversely affect the classification performance.

To address these issues, we propose the local kernel alignment and the corresponding algorithm in this paper. Local kernel alignment discards the kernel values of dissimilar samples, and thus, it avoids the above-mentioned disadvantages. In specific, we first choose the k-nearest neighbors for the ith sample where $i \in \{1, 2, \cdots, N\}$, then compute the local kernels $\mathbf{K}_i \in \mathbb{R}^{k\times k}$ and their corresponding target kernels $\mathbf{T}_i$. After that, we align the average kernel of all local kernels and the average target kernel. To further understand the local kernel alignment, we use Fig. 1 to illustrate the difference between our local kernel alignment and global kernel alignment.

By revisiting the local kernel alignment, we observe that the samples belonging to different classes while with similar features will influence the performance if we only maximize the local ker-

* Corresponding author.
E-mail address: yqwang2013@163.com (Y. Wang).

http://dx.doi.org/10.1016/j.patcog.2017.05.005
0031-3203/

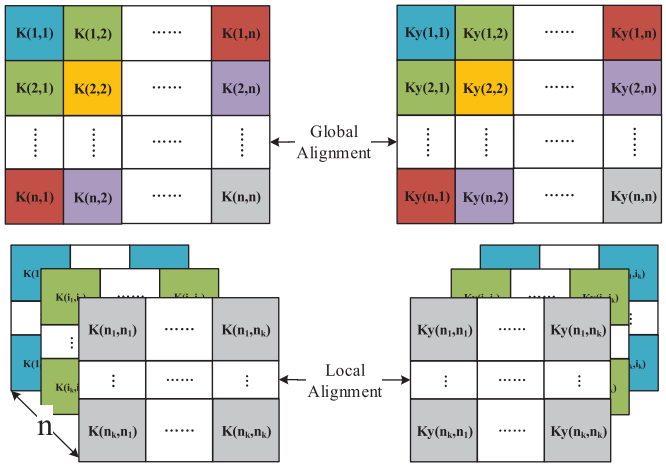

Fig. 1. Global kernel alignment and local kernel alignment. The $\mathbf{K}(i_j, i_k)$ in the local kernel indicates kernel value between the kth and jth neighbors of the ith sample.

nel alignment. In fact, both global and local kernel alignment have their shortcomings. Therefore, these two approaches are complementary, which indicates that the shortcomings will be overcome by combining these approaches. In this paper, we integrate the advantages of both the global and local kernel alignment, and maximize the hybrid kernel alignment to achieve higher performance. Extensive experiments have been conducted to compare the proposed algorithm with state-of-the-art MKL algorithms on 10 UCI and 6 MKL benchmark datasets. These results clearly validate the effectiveness of the proposed algorithm.

We end up this section by summarizing the contributions of our work as follows: i) We first propose the local kernel alignment, which compute and align the kernels in a local way; ii) We propose a novel kernel alignment approach that concurrently considers global and local kernel alignment; iii) We design an alternative algorithm with proved convergence to optimize the coefficient of the combined kernel. This algorithm is simple and easy-to-implement. The experimental results also show the effectiveness of our proposed algorithm.

The remainder of this paper is organized as follows. Section 2 briefly discusses related works. Section 3 presents the technical details of the proposed concepts and algorithm. Section 4 reports the results of the comparative experiments. Finally, Section 5 summarizes the paper and presents future research issues.

2. Related work

Let $\{\phi(x_i), y_i\}_{i=1}^n$ denote the training set, where $\phi(x_i) = [\phi_1^\top(x_i), \cdots, \phi_m^\top(x_i)]^\top$, $\{\phi_p(\cdot)\}_{p=1}^m$ are the feature mappings that corresponding to m pre-defined base kernels $\{K_p(\cdot,\cdot)\}_{p=1}^m$, and y_i is the class label of x_i. In the following sections, we also use $\{\mathbf{X}, \mathbf{Y}\}$ to represent the training set. We assume that the base kernels are positive semi-definite (PSD), and consider a kernel function K with the form $K = \sum_{p=1}^m \mu_p K_p$ where $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_m)^\top$ is selected from $\Delta_q = \boldsymbol{\mu} : \boldsymbol{\mu} \geq 0, \|\boldsymbol{\mu}\|_q = 1$.

2.1. Kernel alignment

Kernel alignment is defined as the similarity between two kernels [20]. The alignment of two kernel matrices is calculated as follows:

$$\rho(\mathbf{K}_1, \mathbf{K}_2) = \frac{\langle \mathbf{K}_1, \mathbf{K}_2 \rangle_F}{\sqrt{\langle \mathbf{K}_1, \mathbf{K}_1 \rangle_F \langle \mathbf{K}_2, \mathbf{K}_2 \rangle_F}} \tag{1}$$

The work in [1] proposes to optimize the kernel alignment by restricting the trace of the kernel matrix to be 1. Then it can be converted into an semi-definite programming (SDP) problem using arbitrary kernel weights in the combination. By restricting the kernel weights to be non-negative, the SDP formulation can be reduced to a quadratically constrained quadratic programming (QCQP) problem.

$$\begin{aligned} &\max \sum_{p=1}^m \mu_p \langle \mathbf{K}_p, \mathbf{Y}\mathbf{Y}^\top \rangle_F, \\ &s.t.\ \boldsymbol{\mu} \in \mathbb{R}_+^m,\ \sum_{p=1}^m \sum_{h=1}^m \mu_p \mu_h \langle \mathbf{K}_p, \mathbf{K}_h \rangle_F \leq 1. \end{aligned} \tag{2}$$

In [21], they propose maximizing the kernel alignment using gradient-based optimization. They compute the gradients with respect to the coefficients as:

$$\frac{\partial(\mathbf{K}_{\boldsymbol{\mu}}, \mathbf{Y}\mathbf{Y}^\top)}{\partial \mu_p} = \frac{\langle \frac{\partial \mathbf{K}_{\boldsymbol{\mu}}}{\partial \mu_p}, \mathbf{Y}\mathbf{Y}^\top \rangle_F \langle \mathbf{K}_{\boldsymbol{\mu}}, \mathbf{K}_{\boldsymbol{\mu}} \rangle_F - \langle \mathbf{K}_{\boldsymbol{\mu}}, \mathbf{Y}\mathbf{Y}^\top \rangle_F \langle \frac{\partial \mathbf{K}_{\boldsymbol{\mu}}}{\partial \mu_p}, \mathbf{K}_{\boldsymbol{\mu}} \rangle_F}{N\sqrt{\langle \mathbf{K}_{\boldsymbol{\mu}}, \mathbf{K}_{\boldsymbol{\mu}} \rangle_F^3}} \tag{3}$$

To prevent overfitting, the work [19] add a regularization term to the objective function. The resulting QP is very similar to the hard margin SVM optimization problem and can be solved in a similar way. To accelerate the optimizing procedure, the work in [22] chooses to optimize the distance between the combined kernel matrix and the ideal kernel, instead of optimizing the kernel alignment measure. The optimization problem is described in Eq. (4). By this way, the corresponding optimized problem can be quickly solved.

$$\min \langle \mathbf{K}_{\boldsymbol{\mu}} - \mathbf{Y}\mathbf{Y}^\top, \mathbf{K}_{\boldsymbol{\mu}} - \mathbf{Y}\mathbf{Y}^\top \rangle_F^2,\ s.t.\ \boldsymbol{\mu} \in \mathbb{R}_+^m,\ \sum_{p=1}^m \mu_p = 1. \tag{4}$$

2.2. Centered kernel alignment for two-stage MKL

Let K and K' be two kernel functions defined over $\chi \times \chi$ such that $0 < \mathbb{E}[K_c^2] < +\infty$ and $0 < \mathbb{E}[K_c'^2] < +\infty$. The center kernel alignment [15] between K and K' is defined by:

$$\rho_g(K, K') = \frac{\mathbb{E}[K_c K_c']}{\sqrt{\mathbb{E}[K_c]^2 \mathbb{E}[K_c']^2}}, \tag{5}$$

where K_c is the center kernel of K described in [15] as follows:

$$\begin{aligned} K_c(x, x') =& K(x, x') - \mathbb{E}_x[K(x, x')] \\ &- \mathbb{E}_{x'}[K(x, x')] + \mathbb{E}_{x,x'}[K(x, x')]. \end{aligned} \tag{6}$$

Let $\mathbf{K} \in \mathbb{R}^{n \times n}$ and $\mathbf{K}' \in \mathbb{R}^{n \times n}$ be kernel matrices associated with K and K', respectively. Then, the alignment between $\mathbf{K}$ and $\mathbf{K}'$ is defined by

$$\widehat{\rho}_g(\mathbf{K}, \mathbf{K}') = \frac{\langle \mathbf{K}_c, \mathbf{K}_c' \rangle_F}{\|\mathbf{K}_c\|_F \|\mathbf{K}_c'\|_F} \tag{7}$$

where $\mathbf{K}_c$ is the corresponding center kernel of K_c which is defined as follows:

$$\begin{aligned} (\mathbf{K}_c)_{ij} =& \mathbf{K}_{ij} - \frac{1}{n}\sum_{i=1}^n \mathbf{K}_{ij} \\ &- \frac{1}{n}\sum_{j=1}^n \mathbf{K}_{ij} + \frac{1}{n^2}\sum_{i,i=1}^n \mathbf{K}_{ij} \end{aligned} \tag{8}$$

Then, the kernel combination coefficients are optimized by solving the following problem,

$$\max_{\mu \in \Delta}\ \widehat{\rho}_g(\sum_{p=1}^m \mu_p \mathbf{K}_p, \mathbf{Y}\mathbf{Y}^\top). \tag{9}$$