Engineering Applications of Artificial Intelligence 64 (2017) 391–400

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Multiple kernel learning using composite kernel functions

Shiju S.S., Asif Salim, Sumitra S. *

Department of Mathematics, Indian Institute of Space Science and Technology, India

ARTICLE INFO

Keywords:
Multiple kernel learning
Classification
Reproducing kernel
Support vector machine
Composite kernel functions

ABSTRACT

Multiple Kernel Learning (MKL) algorithms deals with learning the optimal kernel from training data along with learning the function that generates the data. Generally in MKL, the optimal kernel is defined as a combination of kernels under consideration (base kernels). In this paper, we formulated MKL using composite kernel functions (MKLCKF), in which the optimal kernel is represented as the linear combination of composite kernel functions. Corresponding to each data point a composite kernel function is designed whose domain is constructed as the direct product of the range space of base kernels, so that the composite kernels make use of the information of all the base kernels for finding their image. Thus MKLCKF has three layers in which the first layer consists of base kernels, the second layer consists of composite kernels and third layer is the optimal kernel which is a linear combination of the composite kernels. For making the algorithm more computationally effective, we formulated one more variation of the algorithm in which the coefficients of the linear combination are replaced with a similarity function that captures the local properties of the input data. We applied the proposed approach on a number of artificial intelligence applications and compared its performance with that of the other state-of-the-art techniques. Data compression techniques had been used for applying the models on large data, that is, for large scale classification, dictionary learning while for large scale regression pre-clustering approach had been applied. On the basis of the performance, rank was assigned to each model we used for analysis, The proposed models scored higher rank than the other models we used for comparison. We analyzed the performance of the MKLCKF model by incorporating with kernelized locally sensitive hashing (KLSH) also and the results were found to be promising.

1. Introduction

The kernel methods are applied in various class of problems like classification (Boser et al., 1992), regression (Pozdnoukhov, 2002), dimensionality reduction (Schölkopf et al., 1998) etc. The performance of the kernel algorithm depends on the selection of reproducing kernel. Hence the development of efficient methods for finding the best kernel is very much essential as far as the area of kernel methods are concerned. The current available tools for kernel selection include techniques like cross validation and multiple kernel learning (MKL), of which the later is a data driven approach. The advantage of MKL is that it automatically finds the best combination of kernels from a pool of available kernels (base kernels).

One of the initial approaches used in MKL algorithms is to represent the optimal kernel as a linear combination of a set of kernels (Lanckriet et al., 2004). By representing the function that generates the data as a linear combination of optimal kernel, the MKL problem finds the function as well as the optimal kernel in a simultaneous manner. Thus in MKL learning paradigm, there are two sets of parameters, where one set of parameters corresponds to unknowns of the function to be learned and the other corresponds to the optimal kernel. MKL techniques use either one stage (Lanckriet et al., 2004) in which both set of parameters are solved in same iteration or two stage optimization technique in which the functional parameters are updated in first stage and kernel parameters are updated in second stage, such that, these two stages are repeated until convergence (Rakotomamonjy et al., 2008).

There exist different approaches for finding the parameters of MKL. Fixed weight approach is used in Pavlidis et al. (2001) in which the kernel parameters are fixed constants, while heuristically calculated weights are assigned for the kernels in de Diego et al. (2010). Other major approaches are regularized MKL (Varma and Babu, 2009b)and localized MKL using two stage (Gonen and Alpaydn, 2013). Bayesian techniques are also used for finding the combination of kernels by defining some priors. Boosting (Bennett et al., 2002), semi-supervised (Wang et al., 2012) and unsupervised (Hsu and Lee, 2011) algorithms are also

* Corresponding author.
E-mail address: sumitra@iist.ac.in (Sumitra S).

http://dx.doi.org/10.1016/j.engappai.2017.06.026
Received 2 February 2017; Received in revised form 9 May 2017; Accepted 29 June 2017
Available online 27 July 2017
0952-1976/

adapted for finding the parameters associated with the representation of optimum kernel.

Classification based approaches have also been applied in MKL (Kumar et al., 2012). MKL for large data (Sonnenburg et al., 2006) and non linear combination of kernels (Cortes et al., 2009) are other versions of MKL learning. MKL theory has been applied in areas such as feature selection (Dileep and Sekhar, 2009), feature fusion (Yeh et al., 2012) etc. Pairwise classification (Kreßel, 1999) is another well researched area in case of multi class classification where pairwise kernels are defined.

The main contribution of the paper is the formulation of MKL using composite kernel functions for finding the best combination of kernels from a given P base kernels for machine learning problems such as classification and regression. With reference to each data point we designed a composite kernel function such that it make use of the information of all the given P base kernels for finding the image at each of the points in its domain. We are proposing two variants of this formulation. In the first variant, the optimal kernel is represented as a linear combination of newly designed kernels. As each composite kernel function is built upon a data point, we introduced a second variant in which the coefficients of the linear combination are replaced with a neighborhood function of the reference data point. This representation makes the algorithm more computationally efficient. We verified the efficiency of the proposed models using real world datasets and compared its performance with existing techniques. The proposed methods showed excellent performance. Of the two variants of the approach, the performance of the second variant was found to be better.

As the data increases the number of training points as well as the number of terms in the proposed kernel increases. Thus the overall complexity of the problem is increased. In order to tackle this problem, subset selection approach developed by Nair and Dodd (2015) is followed for regression and dictionary learning approach (Jiang et al., 2013) for classification. We did experimental analysis by incorporating these two in the proposed method and the results were found to be promising.

The rest of the paper is organized as follows. The Section 2 describes the background theory and state-of-the-art algorithms. The Section 3 details the theory behind the proposed weighted kernel approach and its applications while Section 3.4 details localized approach and its applications. The experimental analysis is given in Section 4.

2. Background and state-of-the-art methods

Kernel methods search the function to be approximated in Reproducing Kernel Hilbert Space (RKHS). Corresponding to each RKHS there exists a unique reproducing kernel and vice versa.

Let $\{(x_1, y_1), (x_2, y_2) \dots (x_N, y_N)\}, x_i \in \mathcal{X} \subset \mathbb{R}^n$ be the training points and $y_i \in \mathbb{R}, i = 1, 2 \dots N$ be the corresponding labels for the training points. In kernel algorithms, the function $f \in$ RKHS $\mathcal{F}$ to be learned is found by minimizing the regularized cost function

$$\sum_{i=1}^{N} l(f(x_i), y_i) + \frac{\lambda}{2}\|f\|^2 \tag{1}$$

where $l()$ is a loss function and $\lambda > 0$ is the regularization parameter. By representer theorem, Kimeldorf and Wahba (1971) and Schlkopf et al. (2001), the function that minimizes the above cost function can be represented as

$$f = \sum_{i=1}^{N} \alpha_i k_{x_i} \tag{2}$$

where $\alpha_i \in \mathbb{R}$ and $(k_{x_i}), i = 1, 2, \dots, N$ are the representer evaluators of input training points. Using the reproducing kernel k over $\mathcal{F}$ (2) becomes

$$f(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x). \tag{3}$$

2.1. Combination of kernels

The first work in the domain of MKL is (Lanckriet et al., 2004), in which the optimal kernel is represented as the linear combination of the base kernels and the parameters are learned from the data using semi-definite programming. Its theory can be briefly described as follows. Consider P base kernels $\{k_1, k_2, \dots, k_p\}$ from which the optimal kernel has to be learned. By applying the theory from Lanckriet et al. (2004), (3) becomes

$$f(x) = \sum_{i=1}^{N} \alpha_i \sum_{j=1}^{P} d_j k_j(x_i, x) \tag{4}$$

where $d_j \geq 0, j = 1, 2, \dots, P$.

The other major works in this domain are Simple MKL, Generalized MKL etc. Simple MKL (Rakotomamonjy et al., 2008) uses the above formulation but solves the MKL much faster using two step optimization. In the first step, the function parameters (α) gets optimized by fixing the kernel parameters (d) and in second step, the kernel weights gets updated using gradient descent approach by fixing function parameters. Generalized MKL (Varma and Babu, 2009b) incorporates a regularization term in its formulation. (Jain et al., 2012) extends the generalized MKL for handling million kernels. However the application of MKL over large data is yet to be solved efficiently in terms of space since the million kernels over a million data points require lots of memory. In localized MKL (Gonen and Alpaydn, 2013), selection of kernels is done in a local manner. The localized MKL may not work well with large number of kernels as well as large number of data points.

2.2. Large data algorithm approaches

The main disadvantage of kernel methods is their computational complexity which scales as $O(N^3)$ where N is the number of training points. Nair and Dodd (2015) developed a supervised pre-clustering approach for scaling kernel based regression by making use of the concepts of uniform continuity and compactness. In our work we used this compression technique for large scale regression problems and dictionary based learning for classification. The description of these compression techniques are given below.

2.2.1. Supervised pre-clustering

In the pre-clustering approach developed by Nair and Dodd (2015), the function f to be learned is uniformly continuous, by assuming that it lies in a continuous RKHS $\mathcal{F}$, having the domain of its members a compact set $\mathcal{X}$. The idea of uniform continuity is used to define a similarity measure on the function to be estimated, As the function, f, is uniformly continuous, corresponding to similarity measure ϵ, there exists a radius, δ, independent of $x \in \mathcal{X}$, such that

$$\hat{d}(f(x), f(x')) < \epsilon \quad \forall\, x' \in B(x, \delta) \tag{5}$$

where $B(x, \delta)$, is an open ball of radius δ in input space and $\hat{d}$ is a suitable metric on $\mathbb{R}$. An open ball $B(x, \delta)$ in the input space is called a *cluster* if all points associated with it satisfy (5). The basic idea of pre-clustering is that any data points which satisfy (5) can be considered to be "similar" and therefore form pre-clusters. The centers of the clusters are then used as a sparse dataset for the function estimation. Output information has also been used to form clusters and hence it is a supervised clustering.

The working procedure of the algorithm is as follows. Corresponding to the given similarity measure ϵ, the algorithm finds the radius δ in an iterative manner. In an iteration, an open ball $B(x, \delta)$ is formed in a greedy manner and all those non center points that satisfy (5) get eliminated. Thus in each iteration, the training points consists of the center of the open balls and those points that do not satisfy (5). The algorithm gets terminated if all the training points under consideration satisfy (5). Otherwise δ gets updated using the formula $\delta := \delta - h$, where h is the step length and moves to the next iteration.