Formulation of Two Stage Multiple Kernel Learning Using Regression Framework

S.S. Shiju, Asif Salim, and S. Sumitra(✉)

Department of Mathematics, Indian Institute of Space Science and Technology,
Thiruvananthapuram, India
{shijusnair.13,sumitra}@iist.ac.in, asifsalim.16@res.iist.ac.in

Abstract. Multiple kernel learning (MKL) is an approach to find the optimal kernel for kernel methods. We formulated MKL as a regression problem for analyzing the regression data and hence the data modeling problem involves the computation of two functions, namely, the optimal kernel function which is related with MKL and the optimal regression function which generates the data. As such a formulation demands more space requirements supervised pre-clustering technique has been used for selecting the vital data points. We used two stage optimization for finding the models, in which, the optimal kernel function is found in the first stage and the optimal regression function in the second stage. Using kernel ridge regression the proposed method had been applied on real world problems and the experimental results were found to be promising.

Keywords: Multiple kernel learning · Regression · Kernel ridge regression

1 Introduction

Kernel algorithms have been successfully applied to various machine learning applications. Compared to other machine learning approaches, kernel algorithms have a strong theoretical foundation and become a popular tool because of their guaranteed convergence and good generalization capacity. Support Vector Machine [3], Kernal Principal Component Analysis [16], Kernel Ridge Regression [14] etc. are examples of kernel algorithms.

Kernel methods represent the solution f of the learning problem in the form

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

where $x_i \in \mathbb{R}^n, i = 1, \ldots N$, are the given inputs, k is the reproducing kernel corresponding to the reproducing kernel Hilbert space in which f lies and $\alpha_i \in \mathbb{R}, i = 1, 2, \ldots N$.

The performance of a kernel algorithm depends on the selection of reproducing kernel. The selection of suitable kernel can be automated using multiple

B.U. Shankar et al. (Eds.): PReMI 2017, LNCS 10597, pp. 61–68, 2017.
https://doi.org/10.1007/978-3-319-69900-4_8

kernel learning (MKL) algorithms, that is, these algorithms select the most suitable reproducing kernel from a pool of kernels by itself. Many formulations of MKL are proposed for learning the kernels which are extensively surveyed in [12].

Generally, in multiple kernel learning algorithms, the reproducing kernel is defined as a linear combination of a set of kernels. Using this concept, (1) can be written as

$$f(x) = \sum_{i=1}^{N} \alpha_i \sum_{l=1}^{P} d_l k_l(x_i, x), d_l \geq 0 \tag{2}$$

where k_l are the reproducing kernels under consideration. The parameters in (2) can be optimized either by using two-step optimization [15] or one-step optimization [11]. In one-step method, all the parameters are updated in each iteration of optimization algorithm. In two step method, the learning parameters (α_i) are optimized in first step by fixing kernel weights and kernel weights (d_l) are updated in next step (fixing learning parameters) and this process continues until convergence. One step method mostly uses an alignment measure [5] which is defined between the kernels. [7,9,19] are extensions of one step optimization technique in which the objective is to minimize the alignment between ideal kernel and combination of kernels by applying techniques like semi-definite programming, advanced gradient based methods *etc*. The works, [6,18] use two stage optimization technique for solving the MKL. The faster optimization of parameters for adapting to large scale data set is detailed in [2,17]. The non linear combination of kernels have been used in [4].

[10] used binary classification approach for finding the optimal kernel associated with binary classification problems. That is in this approach the optimal kernel is a function $f^* : \mathcal{X}^* \subset \mathbb{R}^P \rightarrow \mathbb{R}$ such that

$$f^*(z) = d^T z \tag{3}$$

where $\mathcal{X}^* = Range(k_1(.,.)) \times Range(k_2(.,.)) \times ... \times Range(k_P(.,.))$ and $d = \{d_1, d_2, \ldots d_P\}^T \in \mathbb{R}^P$ is as given in (2). From (3) it is clear that f^* is a hyperplane defined on $\mathcal{X}^*$. Using this approach (2) is represented as

$$f(x) = \sum_{i=1}^{N} \alpha_i f^*(\tilde{K}(x, x_i)) \tag{4}$$

where $\tilde{K}(x, x_i) = [k_1(x, x_i) \;\; k_2(x, x_i) \;\; \ldots \;\; k_p(x, x_i)]^T$.

f^* is found out using the N^2 data points $\{(\tilde{K}(x_i, x_j), y_i y_j), i, j = 1, 2, \ldots N\}$. The output for f^* is generated using the ideal kernel, that is, $f^*(\tilde{K}(x_i, x_j)) = k(x_i, x_j) = y_i * y_j$ where x_i and x_j are input data points and y_i and y_j are corresponding labels.

The main contribution of this paper is the formulation of MKL as a regression problem for solving regression data sets. For that the methodology used by [10] is adopted. We proved that the ideal kernel for this formulation is same that of [10]. The main challenge in that approach is that, for training f^*, N^2 training points has to be stored in memory. [10] used a fast optimization algorithm using all N^2

points for training f^*. On the other hand we used data compression approach, namely, supervised pre-clustering approach for finding the vital points. Kernel Ridge regression was used for finding the models.

The rest of the paper can be summarized as follows. The details of the model we proposed is given in Sect. 2: we proved that ideal kernel concept used in classification MKL algorithms is valid for MKL Regression formulation also. Its description is given in Sect. 2.1; the concept of supervised pre-clustering is explained in Sect. 2.2, while the details of optimization we followed is discussed in Sect. 2.3. In Sect. 3 the experimental results and their analysis are given.

2 Regression Frame Work for MKL

We adopted the techniques used in [10] for developing the regression framework for MKL. This section explains the different components of the model we developed.

For developing f^* using regression, input and output data is needed. As the objective of MKL algorithms is to find the best possible kernel, it could be assumed that the output of f^* is the same as the output of the best available kernel (ideal kernel). We have proved that the ideal kernel for regression is $k(x_i, x_j) = y_i.y_j$ using kernel ridge regression framework. The description is given below.

2.1 Ideal Kernel Over Regression Data

The cost function corresponding to kernel ridge regression can be stated as

$$\min_{\alpha \in \mathbb{R}^n} \ \frac{1}{2}\|K\alpha - y\|^2 + \frac{\lambda}{2}\alpha^T K\alpha$$

where K is the kernel matrix, y is the training output vector, $\lambda > 0$ is the regularization parameter and α is the solution vector. The representation for optimal α is

$$\alpha = (K + \lambda I)^{-1} y \tag{5}$$

Let v be the actual output value for a data point x then its predicted output label v_{pred} can be written as

$$\tilde{k}^T \ \alpha = v_{pred} \tag{6}$$

where $\tilde{k} = [k(x_1, x) \ k(x_2, x) \dots k(x_N, x)]^T$,

If the ij^{th} element of the kernel matrix is $k(x_i, x_j) = y_i * y_j$ then (5) can be written as below

$$\alpha = (yy^T + \lambda I)^{-1} y \tag{7}$$

where $y = [y_1, y_2, \dots y_N]^T$

Now $\tilde{k} = yv$ and hence (6) becomes

$$v_{pred} = vy^T \ \alpha$$

Using Eq. (7)

$$v_{pred} = vy^T \ (yy^T + \lambda I)^{-1} y \tag{8}$$

Using Sherman-Morrison Theorem inverse associated with (8) can be found. If A is an invertible square matrix and u, v are column vectors, then Sherman-Morrison formula states that

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1}u} \tag{9}$$

If we consider $A = \lambda I$ and $u = v = y$ then

$$(\lambda I + yy^T)^{-1} =(\lambda I)^{-1} - \frac{(\lambda I)^{-1}yy^T(\lambda I)^{-1}}{1 + y^T(\lambda I)^{-1}y} = \frac{I}{\lambda} - \frac{\frac{yy^T}{\lambda^2}}{1 + \frac{y^T y}{\lambda}} \tag{10}$$

Now

$$\begin{aligned} y^T(yy^T + \lambda I)^{-1}y =& y^T\left(\frac{I}{\lambda} - \frac{\frac{yy^T}{\lambda^2}}{1 + \frac{y^T y}{\lambda}}\right)y = \frac{y^T y}{\lambda} - \frac{\frac{y^T yy^T y}{\lambda^2}}{1 + \frac{y^T y}{\lambda}} \\ =& \frac{\frac{y^T y}{\lambda}}{1 + \frac{y^T y}{\lambda}} \end{aligned} \tag{11}$$

Therefore

$$y^T\ (yy^T + \lambda I)^{-1}y \to 1, \text{when } \lambda \to 0 \tag{12}$$

Substituting Eq. (12) in Eq. (8) we get

$$v_{pred} = vy^T\ (yy^T + \lambda I)^{-1}y \sim v \times 1 \sim v \tag{13}$$

This means that $k(x_i, x_j) = y_i y_j$ is an ideal kernel for regression problems.

2.2 Data Compression

As discussed earlier the data points corresponding to f^* scales as $O(N^2)$. We used supervised pre-clustering approach for compressing the data in an efficient manner.

[13] developed a supervised pre-clustering approach for scaling kernel based regression by making use of the concepts of uniform continuity and compactness. In the pre-clustering approach developed by [13], 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 $\mathcal{X}$. i.e., for the function f, corresponding to similarity measure ϵ, there exists a radius, δ, independent of $x \in \mathcal{X}$, such that

$$\hat{d}(f(x), f(x')) < \epsilon\ \forall\ x' \in B(x, \delta) \tag{14}$$

The basic idea of pre-clustering is that any data points which satisfy (14) can be considered to be "similar" and therefore form pre-clusters. The centers of the clusters are then used as a sparse data set for the function estimation.

If $M << N$ are the data points after compression then f^* can be found using the $M^2 << N^2$ data points $\left\{\left(\tilde{K}(x_i, x_j), y_i y_j\right), i, j = 1, 2, \dots M\right\}$.

2.3 Two Stage Approach

We used two stage optimization for finding f and f^*, that is f^* is first solved and then f is found out using the new f^*. Kernel ridge regression approach is used to find f and f^*.

M^2 data points find out using pre-clustering approach is used to train f^*, that is the input data is $\left\{\left(\tilde{K}(x_i, x_j), y_i y_j\right), i, j = 1, 2, \ldots M\right\}$. The corresponding outputs are generated using the ideal kernel. As f^* is in the form of a hyperplane it is assumed that it lies in a RKHS whose reproducing kernel is the linear kernel.

Let $\tilde{K}$ be the kernel matrix associated with f. Then its ij^{th} element $\tilde{k}_{ij} = f^*(\hat{K}(x_i, x_j))$. The optimal α associated with f is found out by minimizing

$$\frac{1}{2}\|\tilde{K}\alpha - y\|^2 + \frac{\lambda}{2}\alpha^T\alpha$$

On solving this equation, we get α as

$$\alpha = (\tilde{K} + \lambda I)^{-1} y \tag{15}$$

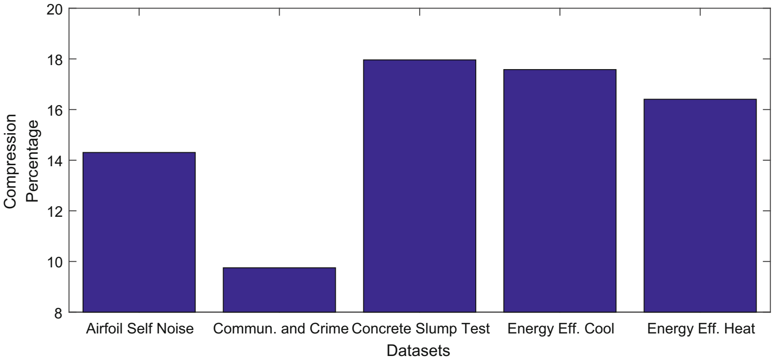

Fig. 1. Compression rate

3 Experiments

The algorithm we developed is named as Two stage Multiple kernel learning approach for regression (TSMKLR). The experimental results are given below.

3.1 Setup

We implemented the proposed algorithms in matlab. The performance of TSMKLR was compared with that of SimpleMKL [15] and SPG-MKL [8] (a modified version of GMKL [18]). The codes for SimpleMKL [15] and SPG-MKL [8] are

Table 1. TSMKL results table

Dataset	KRR	SimpleMKL	SPG-GMKL	TSMKL
Airfoil self noise	4.22529 ± 0.17282 (4)	3.83287 ± 0.20978 (3)	3.40593 ± 0.32411 (2)	3.13291 ± 0.29307 (1)
Commun. and crime	5.82782 ± 0.33076 (2)	5.79657 ± 0.29028 (2)	5.86840 ± 0.29237 (2)	5.00437 ± 0.31056 (1)
Concrete slump test	7.53245 ± 0.51391 (4)	6.48337 ± 0.45852 (3)	6.09983 ± 0.52536 (2)	5.46865 ± 0.38802 (1)
Energy eff. cool	1.85125 ± 0.12772 (3)	1.33792 ± 0.10755 (2)	1.23957 ± 0.10164 (2)	1.15763 ± 0.10176 (1)
Energy eff. heat	2.68947 ± 0.18045 (4)	2.40471 ± 0.21294 (3)	1.40673 ± 0.03337 (2)	1.04312 ± 0.14548 (1)
Average	3.4	2.6	2	1

taken from the author web pages. All the experiments were conducted on the same machine throughout under similar conditions.

Using different hyper parameters in reproducing kernel functions such as Laplacian Kernel, Gaussian Kernel and Polnomial Kernel, 42 base kernels were generated. The σ of both Laplace and Gaussian kernel are assigned with values from $[2^{-9}, 2^{-8}, ..., 2^{9}]$. The polynomial kernel of degree 1,2,3 and 4 were used. The performance for the proposed model were assessed using root mean square (RMSE). Datasets are collected from UCI repository [1].

3.2 Results and Analysis

Using pre-clustering approach data was compressed. The ratio of compression for the datasets are shown in Fig. 1. The compressed data are used to compute the training points for f^*. Using f^*, f was computed. The experimental results are shown in Table 1. It shows that TSMKLR produced superior results in comparison with other models. The difference between the results of TSMKLR and that of other models were statistically significant.

The t-test was performed over the 30 times hold out results for verifying the statistical significance of the results (significance level $\alpha = 0.1$). Based on the statistical significance measure, the models were ranked for their performance on each data. For example: let M_1 and M_2 are two models; let P_1 and P_2 are the values of a performance measure P for a given data set D. Then we say that M_1 is better than M_2 on the basis of P on D if $P_1 > P_2$ and their difference is statistically significant.

4 Conclusion

We have extended the two stage MKL algorithm binary classification framework to regression domain. For that we proved that the ideal kernel for regression is

$k(x_i, x_j) = y_i y_j$. The supervised pre-clustering approach was used to select the vital points. The experiment results clearly proved that the proposed framework is a suitable approach in finding the optimal kernel as far regression data is concerned.

References

1. Asuncion, A., Newman, D.: UCI machine learning repository (2007). http://www.ics.uci.edu/~mlearn/MLRepository.html
2. Bach, F.R., Lanckriet, G.R.G., Jordan, M.I.: Multiple kernel learning, conic duality, and the SMO algorithm. In: Proceedings of the Twenty-First International Conference on Machine Learning, ICML 2004, p. 6. ACM (2004)
3. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT 1992, pp. 144–152. ACM, New York (1992). http://doi.acm.org/10.1145/130385.130401
4. Cortes, C., Mohri, M., Rostamizadeh, A.: Learning non-linear combinations of kernels. In: Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., Culotta, A. (eds.) Advances in Neural Information Processing Systems, vol. 22, pp. 396–404 (2009)
5. Cristianini, N., Kandola, J., Elisseeff, A., Shawe-Taylor, J.: On kernel-target alignment. In: Advances in Neural Information Processing Systems, vol. 14, pp. 367–373. MIT Press (2002)
6. Gonen, M., Alpaydn, E.: Localized algorithms for multiple kernel learning. Pattern Recogn. **46**(3), 795–807 (2013)
7. Igel, C., Glasmachers, T., Mersch, B., Pfeifer, N., Meinicke, P.: Gradient-based optimization of kernel-target alignment for sequence kernels applied to bacterial gene start detection. IEEE/ACM Trans. Comput. Biol. Bioinform. **4**(2), 216–226 (2007)
8. Jain, A., Vishwanathan, S.V.N., Varma, M.: SPG-GMKL: generalized multiple kernel learning with a million kernels. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, August 2012
9. Kandola, J., Shawe-Taylor, J., Cristianini, N.: Optimizing kernel alignment over combinations of kernels. Technical report 121, Department of Computer Science, Royal Holloway, University of London, UK (2002)
10. Kumar, A., Niculescu-Mizil, A., Kavukcuoglu, K., Daume III., H.: A Binary Classification Framework for Two-Stage Multiple Kernel Learning. ArXiv e-prints, June 2012
11. Lanckriet, G.R.G., Cristianini, N., Bartlett, P., Ghaoui, L.E., Jordan, M.I.: Learning the kernel matrix with semi-definite programming. J. Mach. Learn. Res. **5**, 27–72 (2004)
12. Mehmet, G., Ethem, A.: Multiple kernel learning algorithms. J. Mach. Learn. Res. **12**, 2211–2268 (2011)
13. Nair, S.S., Dodd, T.J.: Supervised pre-clustering for sparse regression. Int. J. Syst. Sci. **46**(7), 1161–1171 (2015)
14. Pozdnoukhov, A.: The analysis of kernel ridge regression learning algorithm. Idiap-RR Idiap-RR-54-2002, IDIAP, Martigny, Switzerland (2002)
15. Rakotomamonjy, A., Bach, F.R., Canu, S., Grandvalet, Y.: Simple MKL. J. Mach. Learn. Res. **9**, 2491–2521 (2008)
16. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. **10**(5), 1299–1319 (1998). doi:10.1162/089976698300017467

17. Sonnenburg, S., Ratsch, G., Schafer, C., Scholkopf, B.: Large scale multiple kernel learning. J. Mach. Learn. Res. **7**, 1531–1565 (2006)
18. Varma, M., Babu, B.: More generality in efficient multiple kernel learning. In: Proceedings of the International Conference on Machine Learning, pp. 1065–1072, June 2009
19. Yu, S., Tranchevent, L.C., Moor, B.D., Moreau, Y.: Kernel-Based Data Fusion for Machine Learning, vol. 345. Springer, Heidelberg (2011)