Localized Multiple Kernel Regression

Mehmet Gönen
Department of Computer Engineering
Boğaziçi University, 34342, İstanbul, Turkey
gonen@boun.edu.tr

Ethem Alpaydın
Department of Computer Engineering
Boğaziçi University, 34342, İstanbul, Turkey
alpaydin@boun.edu.tr

Abstract

*Multiple kernel learning (*MKL*) uses a weighted combination of kernels where the weight of each kernel is optimized during training. However,* MKL *assigns the same weight to a kernel over the whole input space. Our main objective is the formulation of the localized multiple kernel learning (*LMKL*) framework that allows kernels to be combined with different weights in different regions of the input space by using a gating model. In this paper, we apply the* LMKL *framework to regression estimation and derive a learning algorithm for this extension. Canonical support vector regression may overfit unless the kernel parameters are selected appropriately; we see that even if provide more kernels than necessary,* LMKL *uses only as many as needed and does not overfit due to its inherent regularization.*

1. Introduction

Recently, methods have been proposed for combining multiple kernels instead of selecting a single one. The simplest approach is to use an unweighted sum of kernels that gives equal importance to each kernel [5]. Using a weighted sum (e.g., convex combination) is more reasonable, and the estimated weights also correspond to the overall importance of kernels. The multiple kernel learning (MKL) framework uses an unweighted summation of discriminant values in different feature spaces [1], which corresponds to a weighted summation of kernel values:

$$f(\boldsymbol{x}) = \sum_{m=1}^{P} \langle \boldsymbol{w}_m, \Phi_m(\boldsymbol{x}) \rangle + b$$

where m indexes feature spaces, $\{\boldsymbol{w}_m\}_{m=1}^{P}$ are the weight coefficients, $\Phi_m(\cdot)$ is the mapping function for feature space m, and b is the bias term. After eliminating $\{\boldsymbol{w}_m\}_{m=1}^{P}$ from the model by using the duality conditions, the discriminant function uses a convex combination of kernels obtained from different feature spaces.

Using a global combination rule (unweighted or weighted) has the disadvantage of assigning the same weight to a kernel over the whole input space. If kernel weights can be assigned in a data-dependent way by considering the underlying localities in training data, a better learner may be produced. This paper is an extension to the localized multiple kernel learning (LMKL) framework [2], where the idea is to divide the input space into regions by using a parametric gating model that assigns higher combination weights to kernels which are suitable for each region.

In Section 2, we give a brief overview of the LMKL framework and then generalize it for regression estimation. We describe an algorithm with a two-step alternating optimization method for regression problems using the localized kernel idea. In Section 3, we describe our experimental procedure and list our empirical results. We summarize and conclude in Section 4.

2. Localized multiple kernel regression

The LMKL framework divides the input space into regions and assigns combination weights to kernels in a data-dependent way. The decision function for binary classification is rewritten as

$$f_{\mathcal{C}}(\boldsymbol{x}) = \sum_{m=1}^{P} \eta_m(\boldsymbol{x}|\mathbf{V}) \langle \boldsymbol{w}_m, \Phi_m(\boldsymbol{x}) \rangle + b$$

where $\eta_m(\boldsymbol{x}|\mathbf{V})$ is a parametric gating model that assigns a weight to feature space m as a function of the input $\boldsymbol{x}$ and $\mathbf{V}$ is the vector of gating model parameters. A similar architecture has been previously proposed under the name "mixture of experts" in the neural network literature [3]. Note that unlike in MKL, in LMKL it is not obligatory to use different feature spaces; we can also use multiple copies of the same feature space in different regions of the input space in order to obtain

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.352

IEEE computer society

a more complex discriminant function. LMKL learns both the support vector coefficients and gating model parameters in a coupled manner using a two-step alternating optimization method [2].

2.1. Regression estimation

In this paper, our objective is to generalize the discriminative LMKL model [2] to regression estimation. The decision function for regression estimation is

$$f_{\mathcal{R}}(\boldsymbol{x}) = \sum_{m=1}^{P} \eta_m(\boldsymbol{x}|\mathbf{V}) \langle \boldsymbol{w}_m, \Phi_m(\boldsymbol{x}) \rangle + b$$

and the optimization problem becomes

$$\begin{aligned}
\text{min.} \quad & \frac{1}{2} \sum_{m=1}^{P} \|\boldsymbol{w}_m\|_2^2 + C \sum_{i=1}^{N} (\xi_i^+ + \xi_i^-) \\
\text{w.r.t.} \quad & \boldsymbol{w}_m, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-, \mathbf{V} \\
\text{s.t.} \quad & \epsilon + \xi_i^+ \geq y_i - f_{\mathcal{R}}(\boldsymbol{x}_i) \qquad \forall i \\
& \epsilon + \xi_i^- \geq f_{\mathcal{R}}(\boldsymbol{x}_i) - y_i \qquad \forall i \\
& \xi_i^+ \geq 0 \qquad \forall i \\
& \xi_i^- \geq 0 \qquad \forall i
\end{aligned} \tag{1}$$

where C is the regularization parameter, $\{\boldsymbol{\xi}^+, \boldsymbol{\xi}^-\}$ are the vectors of slack variables, and ϵ is the tube width. The optimization problem in (1) is not convex due to the nonlinearity formed by using the gating model outputs in the constraints. For a given $\mathbf{V}$, (1) becomes a convex optimization problem and we can obtain the dual formulation as

$$\begin{aligned}
\text{max.} \quad & J(\mathbf{V}) = \sum_{i=1}^{N} y_i(\alpha_i^+ - \alpha_i^-) - \epsilon \sum_{i=1}^{N} (\alpha_i^+ + \alpha_i^-) \\
& \quad - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-) k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j) \\
\text{w.r.t.} \quad & \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^- \\
\text{s.t.} \quad & \sum_{i=1}^{N} (\alpha_i^+ - \alpha_i^-) = 0 \\
& C \geq \alpha_i^+ \geq 0 \qquad \forall i \\
& C \geq \alpha_i^- \geq 0 \qquad \forall i
\end{aligned} \tag{2}$$

where the *locally combined kernel function* is

$$k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{m=1}^{P} \eta_m(\boldsymbol{x}_i|\mathbf{V}) k_m(\boldsymbol{x}_i, \boldsymbol{x}_j) \eta_m(\boldsymbol{x}_j|\mathbf{V})$$

and the resulting decision function is

$$f_{\mathcal{R}}(\boldsymbol{x}) = \sum_{i=1}^{N} (\alpha_i^+ - \alpha_i^-) k_\eta(\boldsymbol{x}_i, \boldsymbol{x}) + b.$$

2.2. Gating models

We can use different gating models. Generally, we want to obtain sparse gating outputs for each data instance; that is, the number of kernels whose corresponding gating outputs are nonzero should be small in order to reduce computational complexity and to provide extra regularization. This is usually achieved by using the softmax function at the output

$$\eta_m(\boldsymbol{x}|\mathbf{V}) = \frac{((\langle \boldsymbol{v}_m, \boldsymbol{x}^{\mathcal{G}} \rangle + v_{m0})}{\sum\limits_{h=1}^{P} \exp(\langle \boldsymbol{v}_h, \boldsymbol{x}^{\mathcal{G}} \rangle + v_{h0})} \tag{3}$$

where $\mathbf{V} = \{\boldsymbol{v}_m, v_{m0}\}_{m=1}^{P}$ and there are $P(D_{\mathcal{G}} + 1)$ parameters where $D_{\mathcal{G}}$ is the dimensionality of the gating feature space.

Using $\boldsymbol{x}^{\mathcal{G}} \equiv \boldsymbol{x}$ in the gating model corresponds to a linear gating model that divides the input space into regions with linear boundaries. If the linear gating model is not adequate, we can use a more complex gating model by extracting nonlinear features from the original features.

2.3. Training with alternating optimization

We can not perform the joint-optimization of the support vector coefficients and gating model parameters in (1) efficiently because of non-convexity. We use a two-step alternating optimization procedure in order to solve (1), as also used for obtaining η_m parameters of MKL in a previous study [6]. There are two steps: (a) solving the model with a fixed gating model, and, (b) updating the gating model parameters with the gradients calculated from the current solution.

Due to convexity, for a given $\mathbf{V}$, the gradients of the objective function $J(\mathbf{V})$ in (2) are equal to the gradients of the objective function in (1). These gradients are used to update the gating model parameters at each step.

2.4. Regularization

The main advantage of LMKL over canonical multiple kernel machines is the inherent regularization effect of the gating model. Canonical methods learn sparse models as a result of regularization on the weight vector but the underlying complexity of the kernel function is the main factor for determining the model complexity. MKL can combine only different kernel functions and more complex kernels are favored over the simpler ones in order to get better performance. However, LMKL can also combine multiple copies of the same kernel and it can dynamically construct a more complex locally combined kernel by using the kernels in a data-dependent way. LMKL eliminates some of the kernels by assigning zero weights to the corresponding gating outputs in order to get a more regularized solution.

3. Experiments

We implement the algorithm in MATLAB and solve the optimization problem (2) with MOSEK optimization software [4]. The linear kernel (k_L), the polynomial kernel (k_P), and the Gaussian kernel (k_G) are used in the experiments:

$$\begin{aligned} k_L(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \\ k_P(\boldsymbol{x}_i, \boldsymbol{x}_j) &= (\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + 1)^q \qquad q \in \mathbb{N} \\ k_G(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 / s^2) \qquad s \in \mathbb{R}_{++}. \end{aligned}$$

We compare support vector regression (SVR) and LMKL on the MOTORCYCLE data set discussed in [7]. The data set has 133 instances and we use 10-fold cross validation to create 10 different training sets. For SVR and LMKL, we take the tube width, ϵ, as 16 and the regularization parameter, C, as 1000. For SVR, k_P with $q = 1, 2, \dots, 20$ and k_G with $s = 0.05, 0.10, \dots, 1.00$ are used. For LMKL, we combine multiple copies of k_L with $P = 1, 2, \dots, 20$.

Figure 1 illustrates the idea behind LMKL for regression problems. We learn a piecewise linear fit through three local models that are obtained using linear kernels in each region and we combine them by using the softmax gating model (shown by dashed lines, which are multiplied by 50 for visual clarity). The softmax model divides the input space between kernels and generally selects a single model to use; we need to evaluate the kernel function between a test instance and only the support vectors in this region. The softmax gating also ensures a smooth transition between local fits.

Figure 2 shows the average of 10 fitted curves with changing parameters on the MOTORCYCLE data set. SVR with k_P ($q = 5$ or 7) overshoot at the boundaries and overfit. SVR with k_G obtains a good fit if the radius, s, is chosen appropriately. LMKL with k_L obtains very similar fits for $P = 6, 13$, and 20 due to its inherent regularization property.

Figure 3 shows the average mean square error values on the test data and the support vector percentages. SVR with k_P overfits the training data with increasing model complexity and performs very badly in terms of mean square error. We see that LMKL with three or more linear kernels is enough to learn the MOTORCYCLE data set. LMKL obtains nearly the same mean square error with three or more linear kernels and is not prone to overfitting. The number of stored support vectors does not increase significantly if we increase the number of kernels combined.

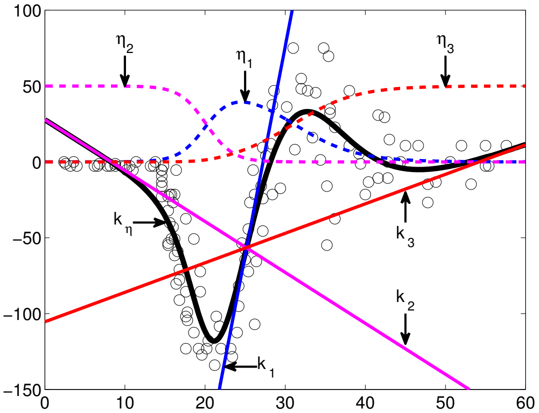


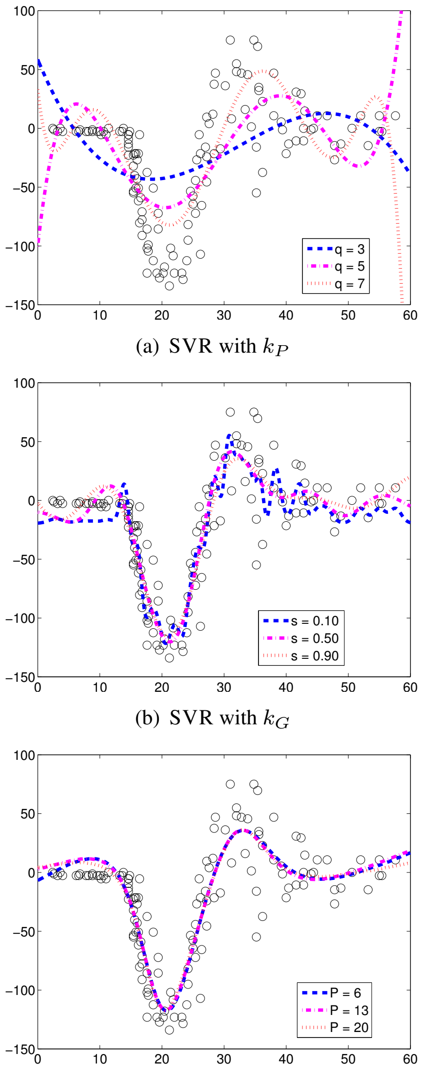
Figure 1. LMKL with three linear kernels.

We also perform experiments on ABALONE, CONCRETE, HOUSING, REDWINE and WHITEWINE data sets from the UCI repository. For each data set, a random one-third is reserved as the test set and the remaining two-thirds is resampled using 5×2 cross-validation to generate ten training and validation sets. The validation sets of all folds are used to optimize C by trying values 0.01, 0.1, 1, 10, and 100. ϵ is selected from $\{0.08, 0.16, 0.32, 0.64, 1.28\}$ for the REDWINE and WHITEWINE data sets, and from $\{1, 2, 4, 8, 16\}$ for the other data sets. The best configuration over the validation sets is used to train the final regressors on the training folds and their performance is measured over the test set. So, for each data set, we have ten test set results and we report their averages. For SVR, k_L and k_P with $q = 2, 3, 4, 5$ are used. For LMKL, we combine multiple copies of k_L with $P = 5$.

Table 1 lists the average mean square errors on the test data and the support vector percentages for the UCI data sets. We see that LMKL obtains statistically comparable mean square error values by using significantly fewer support vectors on all data sets.

Table 1. Results on the UCI data sets.

	SVR		LMKL	
Data Set	MSE	SV	MSE	SV
ABALONE	4.5564	50.89	4.7399	16.03
CONCRETE	58.3216	73.16	51.8043	58.76
HOUSING	14.5893	53.23	16.4055	32.82
REDWINE	0.4376	62.40	0.4569	29.14
WHITEWINE	0.5160	66.27	0.5274	48.90



(a) SVR with k_P

(b) SVR with k_G

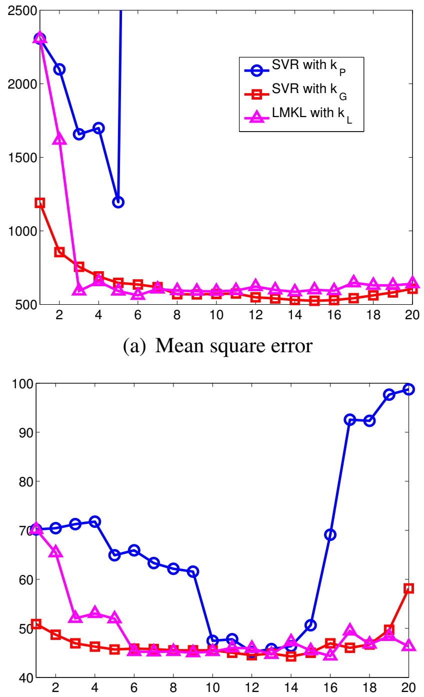
(c) LMKL with k_L

Figure 2. Fits on the MOTORCYLE data set.

4. Conclusions

We generalize the LMKL framework to regression estimation and test it on six benchmark regression data sets. We see that LMKL uses enough number of kernels to match the complexity of the underlying problem. Even if we provide more kernels than necessary, the proposed method uses only as many as required and does not overfit, unlike canonical SVR that may overfit if the kernel parameters are not selected appropriately.

Acknowledgments This work was supported by the Turkish Academy of Sciences in the framework of the Young Scientist Award Program under EA-TÜBA-GEBİP/2001-1-1, the Boğaziçi University Scientific Research Project 07HA101, and the Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grant EEEAG 107E222. The work of M. Gönen was supported by the Ph.D. scholarship (2211) from TÜBİTAK.



(a) Mean square error

(b) Support vector percentage

Figure 3. Results on the MOTORCYLE data set.

References

[1] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML*, 2004.

[2] M. Gönen and E. Alpaydın. Localized multiple kernel learning. In *ICML*, 2008.

[3] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

[4] Mosek. *The MOSEK Optimization Tools Manual Version 6.0 (Revision 66)*. MOSEK ApS, Denmark, 2010.

[5] P. Pavlidis, J. Weston, J. Cai, and W. N. Grundy. Gene functional classification from heterogeneous data. In *RECOMB*, 2001.

[6] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *JMLR*, 9:2491–2521, 2008.

[7] B. W. Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *JRSS: Series B*, 47:1–52, 1985.