

# Multiple Kernel Learning for Support Vector Regression <sup>\*</sup>

Shibin Qiu<sup>†</sup>      Terran Lane<sup>‡</sup>

## Abstract

Kernel support vector (SV) regression has successfully been used for prediction of nonlinear and complicated data. However, like other kernel methods such as support vector machine (SVM) classification, the quality of SV regression depends on proper choice of kernel functions and their parameters. Kernel selection for model selection is conventionally performed through repeated cross validation over a range of kernels and their parameters. Multiple kernel learning arises when a range of kernel parameters need to be tuned within one training process, when different types of kernels are applied simultaneously, or when data are from heterogeneous sources and are characterized with different kernels. Multiple kernel learning can improve the efficiency of kernel selection by automatically evaluate the relative importance of the candidate kernels. Inspired by recent developments in kernel selection for SVM classification, we investigate multiple kernel learning for SV regression. Since more slack variables and constraints are involved in the optimization formulation of SV regression than SVM, we can only follow the general procedures used by SVM but cannot directly use the results derived from SVM. We transform the optimization problems of SV regressions using both  $\varepsilon$ -insensitive loss function and automatic error control into proper forms so that they can be formulated as semidefinite programming (SDP) problems. To avoid the high computational cost of SDP programming, we further formulate multiple kernel SV regression into quadratically constrained quadratic programming optimization problems. Experiments on public and simulated data sets demonstrate that multiple kernel SV regression improves prediction accuracy, reduces the number of support vectors, and helps characterize the properties of the data.

**Keywords:** Multiple kernel learning, support vector regression, model selection, kernel selection, QCQP, SDP.

---

<sup>\*</sup>This work is supported by NIH grant number 1P20RR18754 from the Institutional Development Award (IDeA) Program of the National Center for Research Resources.

<sup>†</sup>Computer Science Department, University of New Mexico, sqiu@unm.edu.

<sup>‡</sup>Computer Science Department, University of New Mexico, terran@cs.unm.edu.

# 1 Introduction

Kernel methods have been successfully applied for classification and prediction by establishing a linear relationship in a transformed feature space through a nonlinear kernel mapping. The success of the kernel trick depends on the proper choice of kernel functions. In most cases, traditional kernel selection tunes parameters for a single kernel and is performed through cross validation. Therefore kernel selection can be time consuming when the kernel space or the parameter space is large. In addition to using a single kernel function for a regression function, it is possible to use multiple kernels. In practice, we often tune kernels based on our initial guess and gradually find the ones that best describe our data. Alternatively, we might have a group of candidate kernel matrices and want to know the relative importance of each kernel. Another situation of practical importance is where data are obtained from heterogeneous sources and we compute different kernels on data from various sources. For example, a biological data set may contain a similarity matrix generated through sequence alignment, and a vectorial description from morphological characteristics of the species. In this case, we can compute a string kernel from the sequence data and a Gaussian kernel upon the vectorial data, and learn the relative significance of the two kernels via the setting of multiple kernel learning. By multiple kernel learning, the relative importance of the kernels can be evaluated together with the solution of the support vectors (SVs). Recently, multiple kernel learning has been automated for support vector machine (SVM) classification using semidefinite programming (SDP) in optimization theory [4]. However, the problem of multiple kernel learning on SV regression has not yet been examined. In this work, we formulate the SV regression problem as SDP and quadratically constrained quadratic programming (QCQP) optimization problems, so that kernel selection can be performed automatically for SV regression.

Since SV regression needs accuracy control on both sides of the regression function using different slack variables to form a error-margin tube, more variables are used to derive its optimization problem. Additionally, some SV regression formulations have more constraints. Consequently, the optimization problem for SV regression is more complicated than SVM. Even though we can follow the general procedures in SVM, as we have done for traditional SV regression, we cannot directly use the results derived from multiple kernel learning in SVM [4] for multiple kernel learning with SV regression. Therefore, we manipulate the variables in multiple kernel SV regressions so that they can be formulated as SDP problems. Although software packages solving SDP problems are available, they are computationally inefficient. To improve efficiency, we then focus on a simplified case of kernel selection, where the composed kernels are linear combinations of the component kernels with positive coefficients. This simplification yields QCQP optimization problems, which can be more efficiently solved than general SDP. Furthermore, nonnegative kernel coefficients also signify the relative importance of the component kernels through their relative magnitudes.

To demonstrate the effectiveness of multiple kernel learning in SV regression, we perform experiments on both publicly available data sets and simulated data. Results have indicated

that using multiple kernel SV regression improves prediction accuracy, reduces the number of support vectors, and helps characterize the properties and structures of the data.

This paper is organized as follows. Section 2 briefly reviews related work to SV regression. Section 3 formulates the optimization problems of multiple kernel learning as SDP problems for both SV regression with  $\varepsilon$ -insensitive loss function and SV regression with automatic accuracy control. In Section 4, we formulate the SV regressions into QCQP problems for kernels spanned by linear combination of component kernels using nonnegative coefficients. We show our empirical results in Section 5, and conclude the paper in Section 6. Some related formulations are given in the appendices.

## 2 Related Work

Support vector regression and the  $\varepsilon$ -insensitive loss function were introduced by Vapnik [18], where SV regression was formulated in a similar way to SVM classification. A comprehensive tutorial on SV regression was given, where the kernel trick, algorithms and variants of SV regression were discussed [12]. To facilitate tuning, SV regression with automatic accuracy control using a soft error-tube was introduced [11]. A soft-tube SV regression eliminates the necessity of specifying the error parameter  $\varepsilon$  in advance. SV regression with automatic accuracy control using linear programming was also examined, where quadratic programming was avoided [13]. To improve computational performance, online learning algorithms were proposed for SV regression [6]. Online learning improves efficiency and adapts the regressor to dynamic changes in the data. Some applications of SV regressions include financial forecasting, physiological prediction, and astronomical data mining [6, 15].

Kernel selection by choosing multiple parameters was studied for an SV classifier [3], where the parameters of a set of Gaussian kernels were tuned in one training process to find the best parameters. Learning multiple kernels for SVM classification was studied to find the best combination of kernels through semidefinite programming and QCQP [4]. We follow the treatment and use the general theoretical background of SDP in a similar way to Lanckriet et al. [4]. But the results for SVM are not directly applicable to SV regression due to different variables and formulations of the optimization problems. We investigate both SV regression with  $\varepsilon$ -insensitive loss function and automatic accuracy control proposed in [18, 11] for the purpose of kernel selection. Unlike the experiments conducted in [11, 13], where only simulated data were used, we test our algorithms on both simulated and public data sets. Multiple kernel learning for SVM in [4] was formulated in a transduction setting, but we simply divide the data into training set and test set. Furthermore, we formulate the optimization problem in standard forms for conic programming using matrix factorization methods.

### 3 SV Regressions as Semidefinite Programming

In this section, we formulate both SV regression with  $\varepsilon$ -insensitive loss function and SV regression with automatic accuracy control as semidefinite programming problems. To clarify notations, we summarize the formulations of each type of the SV regressions. We then manipulate the variables to rewrite the optimization problems in proper forms ready for SDP formulation. Finally, we present their SDP formulations.

#### 3.1 SV regression with $\varepsilon$ -insensitive loss function

Given a set of training samples

$$\{(x_1, z_1), (x_2, z_2), \dots, (x_l, z_l)\},$$

where  $x_i$  ( $1 \leq i \leq l$ ) is a vector in an input space  $\mathcal{X}$ ,  $z_i \in \mathbb{R}$  is its target label, and  $l$  is the number of training samples, we want to learn a regression function

$$f(x) = w^\top \phi(x) + b, \quad (1)$$

that can best predict unseen data  $x$  drawn from the same distribution as the training data. In  $f(x)$ ,  $w$  is the weight vector in the kernel feature space,  $\phi(x)$  is the kernel feature map of data point  $x$ , and  $b \in \mathbb{R}$  is a threshold constant.  $f(x)$  can be solved through the following optimization problem [18].

$$\begin{aligned} \min_{w, b, \xi, \xi^*} \quad & \frac{1}{2} w^\top w + C \sum_{i=1}^l \xi_i^+ + C \sum_{i=1}^l \xi_i^- \\ \text{s.t.} \quad & z_i - w^\top \phi(x_i) - b \leq \varepsilon + \xi_i^+ \\ & w^\top \phi(x_i) + b - z_i \leq \varepsilon + \xi_i^- \\ & \xi_i^+, \xi_i^- \geq 0, \quad i = 1, \dots, l \end{aligned} \quad (2)$$

where  $\varepsilon \geq 0$  is the parameter in the  $\varepsilon$ -insensitive loss function and controls the accuracy of the regressor. The parameter  $C$  adjusts the tradeoff between the regression error and the regularization on  $f$ .  $\xi^+ = \{\xi_1^+, \dots, \xi_l^+\} \in \mathbb{R}^l$  and  $\xi^- = \{\xi_1^-, \dots, \xi_l^-\} \in \mathbb{R}^l$  are slack variables allowing for errors around the regression function.

Introducing Lagrange multipliers  $\alpha_i^+$  on constraints corresponding to  $\xi_i^+$  and  $\alpha_i^-$  on con-

straints corresponding to  $\xi_i^-$ , the dual problem of (2) can be written as,

$$\begin{aligned} \max_{\alpha^+, \alpha^-} \quad & -\frac{1}{2}(\alpha^+ - \alpha^-)^\top K(\alpha^+ - \alpha^-) - \varepsilon \sum_{i=1}^l (\alpha_i^+ + \alpha_i^-) \\ & + \sum_{i=1}^l z_i (\alpha_i^+ - \alpha_i^-) \\ \text{s.t.} \quad & \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) = 0 \\ & \alpha_i^+, \alpha_i^- \in [0, C], \quad i = 1, \dots, l \end{aligned} \quad (3)$$

where  $\alpha^+ = \{\alpha_1^+, \dots, \alpha_l^+\} \in \mathbb{R}^l$  and  $\alpha^- = \{\alpha_1^-, \dots, \alpha_l^-\} \in \mathbb{R}^l$  are the dual variables, and  $K \in \mathbb{R}^{l \times l}$  is the kernel matrix evaluated from a kernel function  $k(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ ,  $K_{ij} = k(x_i, x_j)$  [12]. Solving  $\alpha^+$ ,  $\alpha^-$ , and  $b$  using KKT (Karush-Kuhn-Tucker) conditions in (3), the regression function of (1) becomes

$$f(x) = \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) k(x, x_i) + b, \quad (4)$$

where  $f(x)$  depends only on the training samples having nonzero coefficients (support vectors) through the representation of the kernel function  $k$ .

To formulate the dual problem of SV regression in (3) as an SDP, we define the following variables.

$$\alpha = \begin{pmatrix} \alpha^+ \\ \alpha^- \end{pmatrix} \in \mathbb{R}^{2l} \quad (5)$$

$$Q(K) = \begin{pmatrix} K & -K \\ -K & K \end{pmatrix} \in \mathbb{R}^{2l \times 2l} \quad (6)$$

$$z = (z_1, \dots, z_l)^\top \quad (7)$$

$$h = \begin{pmatrix} -\varepsilon e + z \\ -\varepsilon e - z \end{pmatrix} \in \mathbb{R}^{2l} \quad (8)$$

where  $e$  is a vector of all ones. We also define  $y \in \mathbb{R}^{2l}$  as  $y_i = 1$ , when  $i = 1, \dots, l$ , and  $y_i = -1$ , when  $i = l + 1, \dots, 2l$ . Using these variables, the dual problem of (3) can be written as,

$$\begin{aligned} \max_{\alpha} \quad & 2h^\top \alpha - \alpha^\top Q(K) \alpha \\ \text{subject to} \quad & y^\top \alpha = 0 \\ & \alpha_i \in [0, C], \quad i = 1, \dots, 2l. \end{aligned} \quad (9)$$

Through variable manipulation, the optimization problem in (9) becomes similar to the SVM formulation and is ready for an SDP formulation, which will be done in later sections.

### 3.2 SV regression with automatic accuracy control

The SV regression of (4) derived from (3) has demonstrated superior prediction accuracy and found successful applications. But the cost parameter  $\varepsilon$  need be specified in advance. To facilitate tuning, SV regression with automatic accuracy control (let's call it nSVR) was introduced, where  $\varepsilon$  was assembled into the objective function [11]. In nSVR, the error margin forms a soft-tube and changes dynamically with data. The primal problem of nSVR is formulated as

$$\begin{aligned}
\min_{w, b, \xi^+, \xi^-, \varepsilon} \quad & \frac{1}{2} w^\top w + C(\theta\varepsilon + \frac{1}{l} \sum_{i=1}^l (\xi_i^+ + \xi_i^-)) \\
\text{s.t.} \quad & z_i - w^\top \phi(x_i) - b \leq \varepsilon + \xi_i^+ \\
& w^\top \phi(x_i) + b - z_i \leq \varepsilon + \xi_i^- \\
& \varepsilon \geq 0 \\
& \xi_i^+, \xi_i^- \geq 0, \quad i = 1, \dots, l.
\end{aligned} \tag{10}$$

In (10), the error width  $\varepsilon$  is traded off against model complexity and slack variables through  $\theta \in (0, 1)$ . Its dual problem can be written as

$$\begin{aligned}
\max_{\alpha^+, \alpha^-} \quad & -\frac{1}{2} (\alpha^+ - \alpha^-)^\top K (\alpha^+ - \alpha^-) + \sum_{i=1}^l z_i (\alpha_i^+ - \alpha_i^-) \\
\text{s.t.} \quad & \sum_{i=1}^l (\alpha_i^+ - \alpha_i^-) = 0 \\
& \sum_{i=1}^l (\alpha_i^+ + \alpha_i^-) \leq Cl\theta \\
& \alpha_i^+, \alpha_i^- \in [0, C], \quad i = 1, \dots, l.
\end{aligned} \tag{11}$$

We define one more variable,

$$g = \begin{pmatrix} z \\ -z \end{pmatrix} \in \mathbb{R}^{2l}. \tag{12}$$

Then the dual problem in (11) can be formulated as,

$$\begin{aligned}
\max_{\alpha} \quad & 2g^\top \alpha - \alpha^\top Q(K)\alpha \\
\text{subject to} \quad & y^\top \alpha = 0 \\
& \alpha^\top e \leq Cl\theta \\
& \alpha_i \in [0, C], \quad i = 1, \dots, 2l.
\end{aligned} \tag{13}$$

Its regression function is the same as in (4).

Since the dual problem of nSVR in (13) has one more constraints compared with SVM, we go through its formulation in detail next.

### 3.3 SDP for SV regression

Solving for  $\alpha$  in the dual problem of nSVR in (13), we get an SV regressor of the form of (4) using a given kernel function  $k$ . In kernel selection, we want to learn the best kernels from a given set  $\mathcal{K}$  of positive semidefinite kernel matrices, in addition to learning the coefficients  $\alpha$ .

SDP solves optimization problems with convex objective functions over convex conic sets of symmetric and positive semidefinite matrices [8, 17]. Although it is possible to construct more general sets of  $\mathcal{K}$ , we restrict  $\mathcal{K}$  to linear combinations of kernel matrices. Since kernel matrices of strong diagonal dominance over fit data and decrease generalization capability [19], we impose a constraint on the trace of the kernel matrices. Therefore, our  $\mathcal{K}$  is the linear span of positive semidefinite kernel matrices with bounded traces,

$$\mathcal{K} = \left\{ \sum_{j=1}^m \mu_j K_j, K_j \succeq 0, \mu_j \in \mathbb{R}, \text{trace}(K) \leq \rho \right\}, \quad (14)$$

where  $K = \sum_{j=1}^m \mu_j K_j$ , and  $\succeq 0$  denotes positive semidefiniteness. In (14), we usually set  $\text{trace}(K) = \rho$ . Then, we can formulate our kernel selection problem for nSRV in (13) as

$$\begin{aligned} \min_{K \in \mathcal{K}} \max_{\alpha} \quad & 2g^\top \alpha - \alpha^\top Q(K) \alpha \\ \text{s.t.} \quad & y^\top \alpha = 0 \\ & \alpha^\top e \leq Cl\theta \\ & \alpha_i \in [0, C], \quad i = 1, \dots, 2l \\ & \text{trace}(K) = \rho. \end{aligned} \quad (15)$$

Since  $K \succeq 0$ , by Schur's complement lemma,  $Q(K) \succeq 0$  (Appendix B). Therefore,  $\psi(K) = \max_{\alpha} 2g^\top \alpha - \alpha^\top Q(K) \alpha$  is convex in  $K$ . The constraints in (15) are also convex, resulting in a convex optimization problem over positive semidefinite matrices. From the theory of SDP, (15) has strong duality [8, 17]. Therefore,  $\min_{K \in \mathcal{K}} \psi(K)$  has a global optima. We can write (15) as

$$\begin{aligned} \min_{K \in \mathcal{K}} \quad & t \\ \text{s.t.} \quad & t \geq \max_{\alpha} 2g^\top \alpha - \alpha^\top Q(K) \alpha \\ & y^\top \alpha = 0 \\ & \alpha^\top e \leq Cl\theta \\ & \alpha_i \in [0, C], \quad i = 1, \dots, 2l \\ & \text{trace}(K) = \rho. \end{aligned} \quad (16)$$

The Lagrangian of (13) is,

$$\begin{aligned} \mathcal{L}(\alpha, \nu, \delta, \lambda, \beta) = & 2g^\top \alpha - \alpha^\top Q(K) \alpha + 2\nu^\top \alpha \\ & + 2\lambda y^\top \alpha + 2\delta^\top (Ce - \alpha) + 2\beta(Cl\theta - \alpha^\top e), \end{aligned} \quad (17)$$

where  $\lambda \in \mathbb{R}$ ,  $\beta \geq 0$ , and  $\nu, \delta \in \mathbb{R}^{2l}$  are vectors whose components are nonnegative. Since the problem is convex and has strong duality, we have

$$\begin{aligned} & \min_{\nu, \delta, \lambda, \beta} \max_{\alpha} \mathcal{L}(\alpha, \nu, \delta, \lambda, \beta) \\ & = \max_{\alpha} \min_{\nu, \delta, \lambda, \beta} \mathcal{L}(\alpha, \nu, \delta, \lambda, \beta). \end{aligned} \quad (18)$$

By KKT condition, at optimality  $\frac{\partial \mathcal{L}}{\partial \alpha} = 0$ , which gives rise to  $\alpha$ ,

$$\alpha = Q^{-1}(g + \nu + \lambda y - \delta - \beta e). \quad (19)$$

Substituting  $\alpha$  above into (17), we eliminate the dual variables  $\alpha$  and obtain the dual problem of (13),

$$\begin{aligned} & \min_{\nu, \delta, \lambda, \beta} (g + \nu + \lambda y - \delta - \beta e)^{\top} Q^{-1}(g + \nu + \lambda y - \delta - \beta e) \\ & \quad + 2C\delta^{\top} e + 2\beta C l \theta \\ & \text{s.t. } \nu \geq 0 \\ & \quad \delta \geq 0 \\ & \quad \beta \geq 0. \end{aligned} \quad (20)$$

Let

$$\begin{aligned} \zeta(K) = & \min_{\nu, \delta, \lambda, \beta} (g + \nu + \lambda y - \delta - \beta e)^{\top} Q^{-1}(g + \nu + \lambda y - \delta - \beta e) \\ & + 2C\delta^{\top} e + 2\beta C l \theta. \end{aligned}$$

$\zeta(K)$  being the minimum implies that for  $t \geq 0$ ,  $\zeta(K) \leq t$  is true if and only if we can find feasible dual variables  $\nu, \delta, \beta, \lambda$ , such that

$$\begin{aligned} & t \geq \\ & (g + \nu + \lambda y - \delta - \beta e)^{\top} Q^{-1}(K)(g + \nu + \lambda y - \delta - \beta e) \\ & + 2C\delta^{\top} e + 2\beta C l \theta. \end{aligned} \quad (21)$$

Using Schur's complement lemma (Appendix A), the above condition can be written in matrix form as,

$$\begin{pmatrix} Q(K) & (g + \nu + \lambda y - \delta - \beta e) \\ (g + \nu + \lambda y - \delta - \beta e)^{\top} & t - 2C\delta^{\top} e - 2\beta C l \theta \end{pmatrix} \succeq 0 \quad (22)$$

Combining (16) and (22), the problem of kernel selection for SV regression becomes

$$\begin{aligned}
& \min_{K,t,\lambda,\nu,\delta,\beta} t & (23) \\
& \text{s.t. } \text{trace}(K) = \rho, \\
& K \in \mathcal{K} \\
& \begin{pmatrix} Q(K) & g + \nu + \lambda y - \delta - \beta e \\ (g + \nu + \lambda y - \delta - \beta e)^\top & t - 2C\delta^\top e - 2\beta C l \theta \end{pmatrix} \succeq 0, \\
& \nu \geq 0, \\
& \delta \geq 0, \\
& \beta \geq 0.
\end{aligned}$$

The non-negativity of  $\nu$  and  $\delta$  can be written in matrix form through the diagonal matrices  $\text{diag}(\nu) \succeq 0$  and  $\text{diag}(\delta) \succeq 0$ . Thus (23) is an SDP problem, which gives rise to solutions for both the optimal kernel matrices and the support vectors, generating a regression function of the following form.

$$f(x) = \sum_{i=1}^l \left( (\alpha_i^+ - \alpha_i^-) \sum_{j=1}^m \mu_j k_j(x_i, x) \right) + b. \quad (24)$$

Computing the solution of (23) can be done by software packages such as SeDuMi [14]. Its time complexity is  $O(m^2 l^{2.5})$ , which is generally high. We develop a more efficient solution next.

## 4 QCQP for SV Regressions

In the previous section, we formulated kernel selection for nSVR regression as an SDP problem in (23) over the set of kernels in (14). If we further restrict  $\mu_j \geq 0$  in (14), we derive a QCQP problem, whose solution is more computationally efficient than SDP. Our set of kernels for the QCQP formulation is  $\mathcal{K}_2$  defined below,

$$\mathcal{K}_2 = \left\{ \sum_{j=1}^m \mu_j K_j, K_j \succeq 0, \mu_j \geq 0, \text{trace}(K) = \rho \right\}. \quad (25)$$

In the following, we focus on SV regression with  $\varepsilon$ -insensitive loss function in (3). The nSVR can be similarly formulated. Let

$$\begin{aligned}
\gamma &= \alpha^+ - \alpha^- \\
\eta &= \alpha^+ + \alpha^-
\end{aligned}$$

The problem of kernel selection with the dual of  $\varepsilon$ -insensitive SV regression in (3) can be written as

$$\begin{aligned}
\min_K \max_{\gamma, \eta} \quad & 2z^\top \gamma - 2\epsilon e^\top \eta - \gamma^\top K \gamma & (26) \\
\text{s.t.} \quad & e^\top \gamma = 0 \\
& -C \leq \gamma \leq C \\
& 0 \leq \eta \leq 2C \\
& \text{trace}(K) = \rho \\
& K = \sum_{j=1}^m \mu_j K_j \\
& K_j \succeq 0 \\
& \mu_j \geq 0,
\end{aligned}$$

and is equivalent to

$$\begin{aligned}
\min_{\mu} \max_{\gamma, \eta} \quad & 2z^\top \gamma - 2\epsilon e^\top \eta - \gamma^\top K \gamma & (27) \\
\text{s.t.} \quad & e^\top \gamma = 0 \\
& -C \leq \gamma \leq C \\
& 0 \leq \eta \leq 2C \\
& \sum_{j=1}^m \mu_j \text{trace}(K_j) = \rho \\
& \mu_j \geq 0, \quad j = 1, \dots, m.
\end{aligned}$$

By strong duality and the fact  $\inf(-u) = -\sup(u)$ , and following a treatment similar to [4], the objective in (27) can be computed as

$$\begin{aligned}
\min_{\mu} \max_{\gamma, \eta} \quad & 2z^\top \gamma - 2\epsilon e^\top \eta - \gamma^\top \sum_{j=1}^m \mu_j K_j \gamma \\
= \max_{\gamma, \eta} \min_{\mu} \quad & 2z^\top \gamma - 2\epsilon e^\top \eta - \sum_{j=1}^m \mu_j \gamma^\top K_j \gamma \\
= \max_{\gamma, \eta} \quad & 2z^\top \gamma - 2\epsilon e^\top \eta + \min_{\mu} \left( - \sum_{j=1}^m \mu_j \gamma^\top K_j \gamma \right) \\
= \max_{\gamma, \eta} \quad & 2z^\top \gamma - 2\epsilon e^\top \eta - \max_{\mu} \sum_{j=1}^m \mu_j \gamma^\top K_j \gamma \\
= \max_{\gamma, \eta} \quad & 2z^\top \gamma - 2\epsilon e^\top \eta - \max_j \left( \frac{\rho}{\text{trace}(K_j)} \gamma^\top K_j \gamma \right). & (28)
\end{aligned}$$

Integrating the constraints in (26), (27) into the objective in (28), the kernel selection problem is equivalent to

$$\max_{\gamma, \eta} 2z^\top \gamma - 2\varepsilon e^\top \eta - \rho t \quad (29)$$

$$\text{s.t. } t \geq \frac{1}{\text{trace}(K_j)} \gamma^\top K_j \gamma, \quad j = 1, \dots, m \quad (30)$$

$$e^\top \gamma = 0$$

$$-C \leq \gamma \leq C$$

$$0 \leq \eta \leq 2C.$$

Since software packages for QCQP solver (such as MOSEK [1]) requires objective function to be linear and either linear or conic constraints, the  $m$  constraints in (30) can be transformed into standard conic forms as in (31), through the following factorization and variable changes.

$$\begin{aligned} K_j &= L_j L_j^\top \\ \beta_j &= L_j^\top \gamma \\ \gamma^\top K_j \gamma &= (L_j^\top \gamma)^\top L_j^\top \gamma = \beta_j^\top \beta_j \\ \text{trace}(K_j)t &= \tau^2 \\ \tau &\geq \sqrt{\beta_j^\top \beta_j} \end{aligned} \quad (31)$$

Since  $K_j \succeq 0$ , we can always factorize  $K_j$  as  $K_j = L_j L_j^\top$  using lower triangular  $L_j$ . This can be done using, for example, Cholesky factorization [5].

Using the above manipulation, we further transform the problem of (29) to the following standard form of conic programming.

$$\min_{\gamma, \eta, \beta_j, v, w, s, t_j} v - 2z^\top \gamma + 2\varepsilon e^\top \eta \quad (32)$$

$$\text{s.t. } w = 1$$

$$at_1 - s = 0$$

$$e^\top \gamma = 0$$

$$-C \leq \gamma \leq C$$

$$0 \leq \eta \leq 2C$$

$$t_j - t_1 = 0, \quad 2 \leq j \leq m$$

$$2vw \geq s^2$$

$$L_j^\top \gamma - \beta_j = 0, \quad j = 1, \dots, m$$

$$t_j \geq \sqrt{\beta_j^\top \beta_j}, \quad j = 1, \dots, m$$

where  $2vw \geq s^2$  is a rotated conic constraint, and  $a = \sqrt{2\rho/\text{trace}(K_j)}$ , when each  $K_j$  is normalized and has trace  $l$ .

Table 1: Data sets used for experiments.  $L$  denotes the number of samples;  $d$  is dimension of the data. ‘‘Ref’’ describes the source of the data set.

Name	Abbreviation	$L$	$d$	Ref
Boston housing	BH	506	13	[16]
Mileage/gallon	MPG	396	8	[16]
Air pollution	AP	60	15	[7]
Mixture	MIX	400	15	[16]

The optimization problem in (32) is a QCQP problem, a special case of SDP. Its complexity is  $O(l^3)$ , an improvement over SDP, especially for large sets of kernel matrices  $\mathcal{K}_2$ .  $\mu_j$  can be recovered from the solutions of the dual variables corresponding to the conic constraints. Kernels corresponding to large  $\mu_j$  are important in describing the data and are significant in characterizing the properties and structures of the data, whereas those with small  $\mu_j$  are insignificant.

## 5 Experiments

We perform experiments on the data sets listed in Table 1. The first three data sets in the table are publicly available. The MIX data set was generated by a polynomial function in a 15 dimensional space equipped with a two-Gaussian mixture with  $\sigma_1 = 0.1$  and  $\sigma_2 = 1$ . We normalize the kernels used for all data sets, where normalization was done as  $K_{ij} = k(x_i, x_j)/\sqrt{k(x_i, x_i)k(x_j, x_j)}$ . For a data set of size  $L$ , we split it into  $l$  training samples and  $N$  test samples, i.e.  $L = l + N$ , the traces of the kernels being  $l$ . We used MOSEK [1] to solve the QCQP problems in (32) and derived SV regressors in the form of (24). To test the prediction accuracy, we conducted 10-fold cross validation on each data set.

We found that using three kernels on the data sets generated better prediction accuracy than using a single kernel. We used a Gaussian kernel with  $\sigma_1 = 5.0$ , an exponential kernel  $k(x_i, x_j) = \exp(-\mu||x_i - x_j||)$ , with  $\mu = 1/(2\sigma_1^2)$ , and a rectangle kernel of support width  $W = 4\sigma_1$  on the BH data set. Details of the rectangle kernel function are described in Appendix C. Using only one Gaussian kernel with  $\sigma = 5.0$  yielded an relative mean squared error (MSE) of  $0.137 \pm 0.122$  based on 10-fold cross validation, as shown in Table 2. Using these three kernels yielded an MSE of  $0.099 \pm 0.120$ , an improvement over single kernel SV regression. The three coefficients in the kernel combination derived from the solution for BH were  $\mu_1 = 10.9$ ,  $\mu_2 = 49.5$ , and  $\mu_3 = 28$ , indicating  $K_2$  was more significant. Table 2 lists the prediction accuracies of multiple kernel SV regression and the best accuracies of single kernel SV regression. Table 2 also indicates that results from multiple kernel regression on the MIX data set were consistent with the parameters used to generate the data, since

Table 2: Performance of single kernel and multiple kernel SV regression.  $MSE_M$  represents the relative mean squared error for multiple kernel SV regression.  $MSE_S$  represents the relative mean squared error for single kernel SV regression. The kernels used on BH are Gaussian ( $\sigma = 5.0$ ), exponential  $\mu = 1/(2 \times \sigma^2)$ , and rectangle kernel of support width  $W = 5\sigma$ . The kernels used on MPG are Gaussian ( $\sigma = 10.1$ ), exponential  $\mu = 1/(2 \times 50^2)$ , and rectangle kernel of support width  $W = 1.5\sigma$ . The kernels used on AP are Gaussian ( $\sigma = 0.05$ ), exponential  $\mu = 1/(2 \times \sigma^2)$ , and a polynomial kernel of  $k(x_i, x_j) = (1 + x_i^\top x_j)^2$ . The kernels used on MIX are Gaussian with  $\sigma=0.1, 1.0,$  and  $5.0$ .

Data set	$\mu_1$	$\mu_2$	$\mu_3$	$MSE_M$	$MSE_S$
BH	10.9	49.5	28.0	$0.099 \pm 0.022$	$0.137 \pm 0.122$
MPG	1.31	1.52	1.30	$0.099 \pm 0.036$	$0.131 \pm 0.083$
AP	5.26	4.12	5.28	$0.00447 \pm 0.0017$	$0.0049 \pm 0.0017$
MIX	12.5	9.6	1.3	$0.07 \pm 0.022$	$0.08 \pm 0.027$

$\mu_1 = 12.5$  (corresponding to  $\sigma_1 = 0.1$ ) and  $\mu_2 = 9.6$  (corresponding to  $\sigma_2 = 1.0$ ) were both significantly larger than  $\mu_3 = 1.3$ .

Table 3: Support vectors used by single kernel and multiple kernel SV regression.  $MSE_S$  denotes the relative mean squared error maintained,  $SV_S$  and  $SV_M$  are the proportion of support vectors needed by the single kernel and multiple kernel SV regression, respectively. The type of kernels and their parameters are listed in Table 2.

Data	$MSE_S$	$SV_S$	$SV_M$
BH	0.137	45%	35%
MPG	0.131	20%	3%
AP	0.0049	15%	5%
MIX	0.08	40%	10%
Average		30%	13.25%

In addition to improvement in prediction accuracy, we found using multiple kernels reduced the number of support vectors in the SV regression predictions. Table 3 shows the number of support vectors needed by each data set to maintain the prediction accuracy achieved by the respective single kernel regressor. As Table 3 suggests, the number of support vectors were reduced by 56% on average. This SV reduction indicates that using more kernels can describe the data better than using a single kernel. Figure 1 shows the support vector ratios in BH for single and multiple kernel regression, demonstrating that multiple kernel SV regression needs fewer support vectors. As shown in the figure, SV reduction is especially significant when the requirement for prediction accuracy is high. Since a complicated data set usually requires more support vectors [18, 6], SV reduction by multiple kernel learning equivalently transforms a difficult problem into an easier one that needs fewer SVs. Therefore multiple kernel regression has a better ability to fit complex data and are adaptive

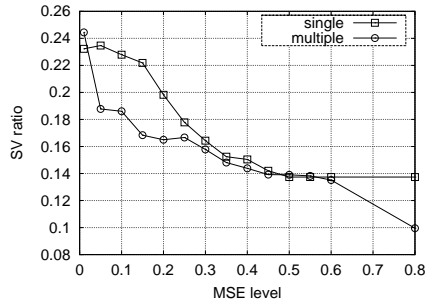


Figure 1: Support vectors required to maintain accuracy levels by single kernel SV regression (“single” in the plot) and multiple kernel SV regression (“multiple” in the plot) for the BH data set. The Y-axis is the required number of support vectors relative to training set size. The X-axis is the required relative MSE level.

to more difficult problems.

We also observed in our experiments that when prediction error was relatively large due to the wrong kernel parameter,  $\mu_j$ s are more different from each other. For example, we used a Gaussian kernel with  $\sigma = 20$  as  $K_1$  and the other two kernels are the same as described in Table 2 for the BH data set. We obtained  $\mu_1 = 3.1E - 5$ ,  $\mu_2 = 0.15$ ,  $\mu_3 = 681$ . Since  $\mu_1$  was very small and the differences were so large, the results were consistent with the inappropriateness of  $K_1$ . When the prediction error was relatively small, the differences between the  $\mu_j$ s were smaller.

## 6 Conclusions

To better use the kernel methods and perform automatic kernel selection, we investigated the problem of multiple kernel learning for SV regression. We transformed the formulation for SV regression with  $\varepsilon$ -insensitive loss function in a higher dimensional variable space and derived a similar formulation as multiple kernel SVM. We then presented it as an SDP problem. We also manipulated the formulation of the SV regression with automatic accuracy control (nSVR), which used variables of higher dimensions and had more constraints. We went through the dual formulation in detail for nSVR and derived its SDP formulation. To avoid the high computational cost of SDP, we further formulated multiple kernel learning for SV regression as a QCQP problem, which required nonnegativity of the kernel combination coefficients and was more computationally efficient. We presented a QCQP formulation for multiple kernel learning on SV regression using  $\varepsilon$ -insensitive loss function, but the formulation is equally applicable to SV regression with automatic accuracy control. We simply divided the data into training set and test set, which is simpler than the transduction setting. We also factorized the quadratic forms in our formulation and derived standard forms for conic programming.

To demonstrate the performance of multiple kernel learning for SV regression, we ex-

perimented on three public data sets and one simulated data set. Results indicated that regression using 3 kernels improved prediction accuracy. The kernel coefficients in multiple kernel regression were able to characterize the properties of the data. Using multiple kernels also reduced the number of support vectors. This support vector reduction equivalently reduced a complicated data set into a simpler one and increased the adaptivity of SV regression, especially for complex data in difficult problems.

Multiple kernel learning for the two popular SV regressions have been formulated as SDP and QCQP problems. The SDP formulation in (23) has global optimal solution, since it has semidefinite matrix constraints and strong duality. The SDP formulation is ready for use by experiments. In this work, we have only tested the performance and behavior of multiple kernel SV regression for  $\mu \geq 0$  over  $\mathcal{K}_2$  in (25), which is a QCQP problem. We have yet to test the performance of SV regressors formulated as SDP problems in (23). Kernel selection over  $\mathcal{K}$  in (14) might generate negative kernel coefficients and we need to explore their implications.

Even though QCQP is more efficient to solve than general SDP, it still requires a commercial solver. For multiple kernel SV regression to be widely applicable, it is better to implement it in a way similar to sequential minimal optimization (SMO) [9] and make it freely available to the data mining community. This algorithmic improvement deserves more investigation. Another direction for the future is to design proper string kernels and apply multiple kernel SV regression to biological applications.

## Appendices

**A.** Schur's complement lemma [2]. Given a partitioned matrix  $D$  of the form

$$D = D^\top = \begin{pmatrix} A & B \\ B^\top & C \end{pmatrix}$$

where  $A$  and  $C$  are square matrices. If  $A^{-1}$  exists, Schur's complement is defined as  $R = C - B^\top A^{-1} B$ . And if  $A \succeq 0$ , then  $R \succeq 0 \Leftrightarrow D \succeq 0$ .

**B.**  $Q(K)$  in (6) is positive semidefinite.  $A = C = K$ ,  $B = B^\top = -K$ ,  $R = K - (-K)^\top K^{-1} (-K)^\top \succeq 0$ .

**C.** The rectangle kernel  $k_R(x_i, x_j)$  is designed to approximate a Gaussian kernel. Inside its support width  $W$ , it takes values of the Gaussian kernel at the middle points of the rectangles. It vanishes when  $r \geq W$ .  $k_R$  is described below and shown in Figure 2, where  $r = \|x_i - x_j\| = \sqrt{\sum_{m=1}^d (x_{im} - x_{jm})^2}$  is the Euclidian distance between  $x_i$  and  $x_j$ .  $k_R$  is positive definite under certain conditions and has computational advantages [10].

$$k_R(x_i, x_j) = \begin{cases} e^{-\frac{1}{2\sigma^2}(\frac{\tau}{2})^2}, & 0 \leq \|x_i - x_j\| \leq \tau; \\ e^{-\frac{1}{2\sigma^2}(\frac{3\tau}{2})^2}, & \tau < \|x_i - x_j\| \leq 2\tau; \\ \dots & \dots \\ 0, & \|x_i - x_j\| > W. \end{cases}$$

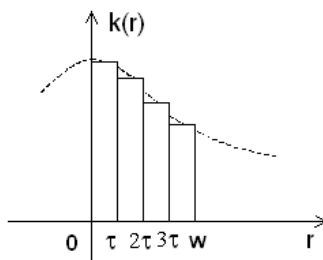


Figure 2: The rectangle kernel function. Dashed line represents the Gaussian kernel and the rectangles are used for approximation.

## References

- [1] E. D. Andersen and A. D. Andersen, *The MOSEK interior point optimizer for linear programming: An implementation of the homogeneous algorithm*, In H. Frenk, C. Roos, T. Terlaky and S. Zhang (editors), *High Performance Optimization*, pp. 197–232, Kluwer Academic Publishers, 2002.
- [2] D. Carlson and T. Markham, *Schur complements of diagonally dominant matrices*, *J. Czech. Math*, Vol. 29, pp. 246-251, 1979.
- [3] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, *Choosing multiple parameters for support vector machines*, *Machine Learning*, Vol. 46, pp. 131–159, 2002.
- [4] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, *Learning the kernel matrix with semidefinite programming*, *J. Machine Learning Research*, Vol.5, pp. 27–72, 2004.
- [5] C. F. Van Loan, *Introduction to scientific computing*, 2ed., Prentice Hall, 2000.
- [6] J. Ma, J. Theiler and S. Perkins, *Accurate online support vector regression*, *Neural Computation*, Vol. 15, pp. 2683–2703, 2003.
- [7] G. C. McDonald and R. C. Schwing, *Instabilities of regression estimates relating air pollution to mortality*, *Technometrics*, Vol.15, pp. 463–482, 1973.

- [8] Y. Nesterov and A. Nemirovsky, *Interior point polynomial methods in convex programming: theory and applications*, SIAM, 1994.
- [9] J. Platt, *Sequential minimal optimization: A fast algorithm for training support vector machines*, In B. Scholkopf, C. Burges, and A Smola (editors), *Advances in Kernel Methods—Support Vector Learning*, MIT Press, Cambridge, USA, 1998.
- [10] S. Qiu and T. Lane, *Parallel computation of RBF kernels for support vector classifiers*, Proc. 5th SIAM International Conference on Data Mining (SDM05), pp. 334–345, Newport Beach, CA, April, 2005.
- [11] B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson, *Support vector regression with automatic accuracy control*, In L. Niklasson, M. Boden, and T. Ziemke (editors), Proc. 8th Int'l Conf. Artificial Neural Networks, pp. 111–116, 1998.
- [12] A. Smola and B. Schölkopf, *A tutorial on support vector regression*, NeuroCOLT2 Tech Report, NC2-TR-1998-030, 1998.
- [13] A. Smola, B. Schölkopf and G. Rätsch, *Linear programs for automatic accuracy control in regression*, Proc. 9th ICANN, London, pp.575–580, 1999.
- [14] J. F. Sturm (1999), *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization Methods and Software, Vol. 11–12, Special Issue on Interior Point Methods, pp. 625–653, 1999.
- [15] T. B. Trafalis, *Support Vector Machine for Regression and Applications to Financial Forecasting*, International Joint Conference on Neural Networks, (IJCNN'00), Vol. 6, p. 6348, 2000.
- [16] UCI machine learning data datasets, <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
- [17] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Review, Vol. 38, pp. 49–95, 1996.
- [18] V. N. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, 1998.
- [19] J. Weston, B. Schölkopf, E. Eskin, C. Leslie and W. S. Noble, *A kernel approach for learning from almost orthogonal patterns*, Proc. 6th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'02), Helsinki, Finland, August, 2002, Springer LNCS 243.