Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2012, Article ID 601296, 10 pages
doi:10.1155/2012/601296

Research Article

ℓ_p-Norm Multikernel Learning Approach for Stock Market Price Forecasting

Xigao Shao,[1,2] Kun Wu,[1] and Bifeng Liao[3]

[1] *School of Mathematics and Statistics, Central South University, Changsha, Hunan 410075, China*
[2] *Wengjing College, Yantai University, Yantai, Shandong 264005, China*
[3] *School of Mathematics and Information Science, Yantai University, Yantai, Shandong 264005, China*

Correspondence should be addressed to Kun Wu, 1027330663@qq.com

Received 10 August 2012; Revised 18 November 2012; Accepted 27 November 2012

Academic Editor: Daoqiang Zhang

Linear multiple kernel learning model has been used for predicting financial time series. However, ℓ_1-norm multiple support vector regression is rarely observed to outperform trivial baselines in practical applications. To allow for robust kernel mixtures that generalize well, we adopt ℓ_p-norm multiple kernel support vector regression ($1 \leq p < \infty$) as a stock price prediction model. The optimization problem is decomposed into smaller subproblems, and the interleaved optimization strategy is employed to solve the regression model. The model is evaluated on forecasting the daily stock closing prices of Shanghai Stock Index in China. Experimental results show that our proposed model performs better than ℓ_1-norm multiple support vector regression model.

1. Introduction

Forecasting the future values of financial time series is an appealing yet difficult activity in the modern business world. As explained by Deboeck and Yaser [1, 2], the financial time series are inherently noisy, nonstationary, and deterministically chaotic. In the past, many methods were proposed for tackling this kind of problem. For instance, the linear models for forecasting the future values of stock prices include the autoregressive (AR) model [3], the autoregressive moving average (ARMA) model [4], and the autoregressive integrated moving average (ARIMA) model [4]. Over the last decade, nonlinear approaches have received increasing attention in financial time series prediction and have been proposed for a satisfactory answer to the problem. For example, Yao and Tan [5] used time series data and technical indicators as the input of neural networks to increase the forecast accuracy of exchange rates; Cao and Tay [6, 7] applied support vector machine (SVM) in financial forecasting and compared it with the multilayer back-propagation (BP) neural network and the regularized radial basis function (RBF) neural network; Qi and Wu [8] proposed a multilayer feed-forward network to forecast exchange rates; Pai and Lin [9] invested a hybrid ARIMA and support vector machines model in stock price forecasting; Pai et al. [10] presented a hybrid SVM model to exploit the unique strength of the linear and nonlinear SVM models in forecasting exchange rate; Kwon and Moon [11] proposed a hybrid neurogenetic system for stock trading; Hung and Hong [12] presented an improved ant colony optimization algorithm in a support vector regression (SVR) model, called SVRCACO, for selecting suitable parameters in exchange rate forecasting; Jiang and He [13] introduced local grey SVR (LG-SVR) integrated grey relational grade with local SVR for financial times eries forecasting; and so on.

In comparison with the previous models, SVR with a single kernel function can exhibit better prediction accuracy because it conceives the structural risk minimization principle which considers both the training error and the capacity of the regression model [14, 15]. However, the researchers have to determine in advance the type of kernel function and the associated kernel hyper parameters for SVR. Unsuitably chosen kernel functions or hyper parameter settings may lead to significantly poor performance [16, 17].

In recent years there has a lot of interest in designing principled regression algorithms over multiple cues, based

on the intuitive notion that using more features should lead to better performance and decreasing the generalization error. When the right choice of features is unknown, learning linear combinations of multiple kernels is an appealing strategy. The approach with a optimization process is called multiple kernel learning (MKL). A first step towards a more realistic model of MKL was achieved by Lanckriet et al. [18], who showed that, given a candidate set of kernels, it is computationally feasible to simultaneously learn a support vector machine and a linear kernel combination at the same time. In MKL we need to solve a joint optimization problem while also learning the optimal weights for combing the kernels. Several practitioners have adopted the linear multiple kernels to deal with the practical problems. For example, Rakotomamonjy et al. [19] addressed the MKL problem through a weighted 2-norm regularization formulation and proposed an algorithm, named Simple MKL, for solving this MKL problem. Bach [20] proposed the asymptotic model consistency of the group Lasso. Zhang and Shen [21] presented multimodal multitask learning algorithm for joint prediction of multiple regression and classification variables in Alzheimer's disease. Especially, Chi-Yuan Yeh and his coworkers [22] developed a two-stage MKL algorithm by incorporating sequential minimal optimization and the gradient projection method. The new method [22] performed better than previous ones for forecasting the financial time series. Previous approaches to multiple kernel learning (MKL) have promoted sparse kernel combinations to support interpretability and scalability. Unfortunately, sparsity at the kernel level may harm the generalization performance of the learner, therefore ℓ_1-norm MKL is rarely observed to outperform trivial baselines in practical applications [23]. To allow for robust kernel mixtures that generalize well, the researchers extend ℓ_1-norm MKL to arbitrary norms, that is, ℓ_p-norm MKL ($1 \leq p < \infty$). For example, Marius Kloft et al. developed two efficient interleaved strategies for ℓ_p-norm MKL and showed that it can achieve better accuracy than ℓ_1-norm MKL for real-world problems [23]; Francesco Orabona et al. presented a MKL optimization algorithm based on stochastic gradient descent for ℓ_p-norm MKL, which possessed a faster convergence rate as the number of kernels grows [24].

In this paper, a multiple kernel learning framework is established for learning and predicting the stock prices. We present a regression model for the future values of stock prices, that is, ℓ_p-norm multiple kernel support vector regression (ℓ_p-norm MK-SVR), where $1 \leq p < \infty$. We decompose the optimization problem into smaller subproblem and adopt the interleaved optimization strategy to solve the regression model. Our experimental results show that ℓ_p-norm MK-SVR performs a better performance.

The rest of this paper is arranged as follows. Section 2 details the processing of the ℓ_p-norm MK-SVR model construction and describes the algorithm for our regression model. Experimental results are presented in Section 3. Section 4 concludes the paper and provides some future research directions.

2. Forecasting Methodology

2.1. ℓ_p-Norm Multiple Kernel Support Vector Regression. In this section, the idea of ℓ_p-norm multiple kernel support vector regression (ℓ_p-norm MK-SVR) is introduced formally.

Let $\{\mathbf{x}_i, y_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbf{R}^n$ and $y_i \in \mathbf{R}$, be the training set. Each y_i is the desired output value for the input vector $\mathbf{x}_i$. Consider a function $\phi(\mathbf{x}_i) : \mathbf{R}^n \rightarrow \mathbf{H}$ that maps the samples into a high, possibly infinite, dimensional space. A regression model is learned from the previous and used to predict the target values of unseen input vectors. SVR is a nonlinear kernel-based regression method which tries to locate a regression hyperplane with small risk in high-dimensional feature space [14]. Considering the soft margin formulation, the objective function and constraints for SVR should be solved, as follows:

$$\begin{aligned} \min_{\widetilde{\mathbf{w}},b} \quad & \frac{\lambda}{2}\langle \widetilde{\mathbf{w}}, \widetilde{\mathbf{w}}\rangle + \frac{1}{N}\sum_{i=1}^{l}\left(\xi_i + \widehat{\xi}_i\right) \\ \text{s.t.} \quad & \left(\langle \widetilde{\mathbf{w}}, \phi(\mathbf{x}_i)\rangle + b\right) - y_i \leq \varepsilon + \xi_i, \\ & y_i - \left(\langle \widetilde{\mathbf{w}}, \phi(\mathbf{x}_i)\rangle + b\right) \leq \varepsilon + \widehat{\xi}_i, \\ & \xi, \widehat{\xi}_i \geq 0, \quad i = 1, 2, \ldots, N. \end{aligned} \tag{1}$$

SVR model usually uses a single mapping function ϕ and hence a single kernel function K. Although the SVR model has good function approximation and generalization capabilities, it is not fit for dealing with a data-set which has a locally varying distribution. For resolving this problem, we can construct a MK-SVR model. Combining multiple kernels instead of using a single one, ℓ_p-norm MK-SVR model can catch up the varying distribution very well. Therefore we can use the composite feature map ϕ which has a block structure:

$$\phi(\mathbf{x}) = \left[\sqrt{d_1}\phi_1(\mathbf{x}) \times \sqrt{d_2}\phi_2(\mathbf{x}) \times \cdots \times \sqrt{d_M}\phi_M(\mathbf{x})\right] \tag{2}$$

to map the input space to the feature space, where $d_1, d_2, \ldots, d_M$ are weights of component functions. Given a set of base kernels K_k which correspond the previous feature maps $\{\phi_k\}(k = 1, 2, \ldots, M)$, linear MK-SVR aims to learn a linear combination of the base kernels as $K = \sum_k d_k K_k$. In learning with MK-SVR we aim at minimizing the loss on the training data with respect to the optimal kernel mixture $\sum_k d_k K_k$ in addition to regularizing $\mathbf{d}$ to avoid overfitting. The primal can therefore be formulated as

$$\begin{aligned} \min_{\widetilde{\mathbf{w}},b} \quad & \frac{\lambda}{2}\left(\sum_k ||\widetilde{\mathbf{w}}_k||_2\right)^2 + \frac{1}{N}\sum_{i=1}^{l}\left(\xi_i + \widehat{\xi}_i\right) + \widetilde{\mu}\widetilde{\Omega}[\mathbf{d}] \\ \text{s.t.} \quad & \left(\langle \widetilde{\mathbf{w}}, K(\mathbf{x}_i)\rangle + b\right) - y_i \leq \varepsilon + \xi_i, \\ & y_i - \left(\langle \widetilde{\mathbf{w}}, K(\mathbf{x}_i)\rangle + b\right) \leq \varepsilon + \widehat{\xi}_i, \\ & \widetilde{\mu} > 0, \\ & d_1, d_2, \ldots, d_M \geq 0, \\ & \xi, \widehat{\xi}_i \geq 0, \quad i = 1, 2, \ldots, N. \end{aligned} \tag{3}$$

Previous research to MK-SVR employs the regularizer of the form $\widetilde{\Omega}[\mathbf{d}] = \|\mathbf{d}\|_1 (\mathbf{d} = (d_1, d_2, \ldots, d_M))$ which can promote sparse kernel mixtures. However, sparsity is not always desirable, since the information carried in the zero-weighted kernels is lost. Therefore we propose to use nonsparse and thus more robust kernel mixtures by employing an ℓ_p-norm constraint with $p > 1$, that is, $\widetilde{\Omega}[\mathbf{d}] = \|\mathbf{d}\|_p^2$, and $\|\mathbf{d}\|_p = (\sum_k {d_k}^p)^{1/p}$, $1 < p < \infty$. In (3), let $\sqrt{d_k}\widetilde{\mathbf{w}}_k = \mathbf{w}_k$, $C = 1/n\lambda$, $\widetilde{\mu} = \mu\lambda$, and the first equation be divided with λ, then the following ℓ_p-norm MK-SVR is obtained:

$$\begin{aligned}
\min_{\mathbf{w},b} \quad & \frac{1}{2}\sum_k \frac{\|\mathbf{w}_k\|_2^2}{d_k} + C\sum_{i=1}^{l}\left(\xi_i + \widehat{\xi}_i\right) + \mu\|\mathbf{d}\|_p^2 \\
\text{s.t.} \quad & (\langle \mathbf{w}, K(\mathbf{x}_i)\rangle + b) - y_i \le \varepsilon + \xi_i, \\
& y_i - (\langle \mathbf{w}, K(\mathbf{x}_i)\rangle + b) \le \varepsilon + \widehat{\xi}_i, \\
& \mu > 0, \\
& d_1, d_2, \ldots, d_M \ge 0, \\
& \xi, \widehat{\xi}_i \ge 0, \quad i = 1, 2, \ldots, N.
\end{aligned} \tag{4}$$

An alternative approach previous equations has been considered by studiers. For example, Zien and Ong [25] upperbound the value of the regularizer $\|\mathbf{d}\|_1$ and incorporate the regularizer as an additional constraint into the optimization problem. According to this thought, ℓ_p-norm MK-SVR model (4) can be transformed into the following form:

$$\begin{aligned}
\min_{\mathbf{w},b} \quad & \frac{1}{2}\sum_k \frac{\|\mathbf{w}_k\|_2^2}{d_k} + C\sum_{i=1}^{l}\left(\xi_i + \widehat{\xi}_i\right) \\
\text{s.t.} \quad & (\langle \mathbf{w}, K(\mathbf{x}_i)\rangle + b) - y_i \le \varepsilon + \xi_i, \\
& y_i - (\langle \mathbf{w}, K(\mathbf{x}_i)\rangle + b) \le \varepsilon + \widehat{\xi}_i, \\
& \|\mathbf{d}\|_p^2 \le 1, \\
& d_1, d_2, \ldots, d_M \ge 0, \\
& \xi, \widehat{\xi}_i \ge 0, \quad i = 1, 2, \ldots, N.
\end{aligned} \tag{5}$$

It can be shown (see the Appendix for details) that the dual of (5) is

$$\begin{aligned}
\max_{\widehat{\boldsymbol{\alpha}},\boldsymbol{\alpha}} \quad & \left(\mathbf{y}^T(\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}) - \boldsymbol{\varepsilon}(\widehat{\boldsymbol{\alpha}} + \boldsymbol{\alpha})\right) \\
& - \left\| \left(\frac{1}{2} d_k \sum_{i=1}^{N}\sum_{j=1}^{N} (\widehat{\alpha}_i - \alpha_i)\left(\widehat{\alpha}_j - \alpha_j\right) K_k\left(\mathbf{x}_i, \mathbf{x}_j\right) \right)_{k=1}^{M} \right\|_{p^*} \\
\text{s.t.} \quad & \mathbf{1}^T(\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}) = 0, \\
& 0 \le \widehat{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \le C\mathbf{1}, \\
& d_1, d_2, \ldots, d_M \ge 0,
\end{aligned} \tag{6}$$

where $\mathbf{y} = (y_1, y_2, \ldots, y_N)^T$, $\boldsymbol{\varepsilon} = (1, 1, \ldots, 1)^T$, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_N)^T \in \mathbf{R}^N$, $\widehat{\boldsymbol{\alpha}} = (\widehat{\alpha}_1, \widehat{\alpha}_2, \ldots, \widehat{\alpha}_N)^T \in \mathbf{R}^N$, and $p^* = p/(p-1)$ is the dual norm of p. Suppose the optimal $\widehat{\alpha}_i^*$, $\alpha_i^* (i = 1, 2, \ldots, N)$ and $d_1^*, d_2^*, \ldots, d_M^*$ are found by solving (6), the regression hyperplane for ℓ_p-norm MK-SVR model is given by

$$f^*(\mathbf{x}) = \sum_{i=1}^{N} (\widehat{\alpha}_i^* - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}) + b^*, \tag{7}$$

where $b^* = y_j + \varepsilon - \sum_{i=1}^{N} (\widehat{\alpha}_i^* - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}_j)$ is obtained from any $\widehat{\alpha}_i^*$ and α_i^*, with $0 < \widehat{\alpha}_i^*, \alpha_i^* < C1$. In the following section, an efficient algorithm is proposed for solving the optimization problem (6).

2.2. An Optimistic Algorithm. ℓ_p-norm MK-SVR model (6) can be trained with several algorithms, for example, the Sequential Minimal Optimization algorithm [26] and multi-kernel learning with online-bath optimization [24]. In this paper, the interleaved optimization is used for the optimization scheme according to the idea of [23]. As a matter of fact, we can exploit the structure of ℓ_p-norm MK-SVR cost function by alternating between optimizing the linear combination of the base kernels $K = \sum_k d_k K_k$ and the remaining variables as $\widehat{\boldsymbol{\alpha}}$ and $\boldsymbol{\alpha}$. We can do so by setting up a two-stage optimization algorithm. The basic idea of the algorithm is to divide the optimization variables of ℓ_p-norm MK-SVR problem (6) into two groups, $(\widehat{\boldsymbol{\alpha}}, \boldsymbol{\alpha})$ on one hand and $\mathbf{d} = (d_1, d_2, \ldots, d_M)$ on the other. Our procedure will alternatingly operate on those two stages via a block coordinate descent algorithm. Therefore the optimization $\mathbf{d}$ will be carried out analytically and the $(\widehat{\boldsymbol{\alpha}}, \boldsymbol{\alpha})$ will be computed in the dual. The two stages are iteratively performed until the specified stopping criterion is met, as shown in Figure 1.

In the first stage, the variables $(\widehat{\boldsymbol{\alpha}}, \boldsymbol{\alpha})$ are kept fixed, that is, the $(\widehat{\boldsymbol{\alpha}}, \boldsymbol{\alpha})$ are known. Then the optimal $\mathbf{d}$ in ℓ_p-norm MK-SVR model (6) can be calculated analytically by the following process.

According to (A.4), let

$$\begin{aligned}
L = & \sum_{i=1}^{N} y_i(\widehat{\alpha}_i - \alpha_i) - \varepsilon\sum_{i=1}^{N}(\widehat{\alpha}_i + \alpha_i) \\
& - \frac{1}{2}\sum_{k=1}^{M} d_k \sum_{i=1}^{N}\sum_{j=1}^{N} (\widehat{\alpha}_i - \alpha_i)\left(\widehat{\alpha}_j - \alpha_j\right) K_k\left(\mathbf{x}_i, \mathbf{x}_j\right) \\
& + \beta\left(\frac{1}{2}\|\mathbf{d}\|_p^2 - \frac{1}{2}\right) - \boldsymbol{\gamma}^T\mathbf{d}.
\end{aligned} \tag{8}$$

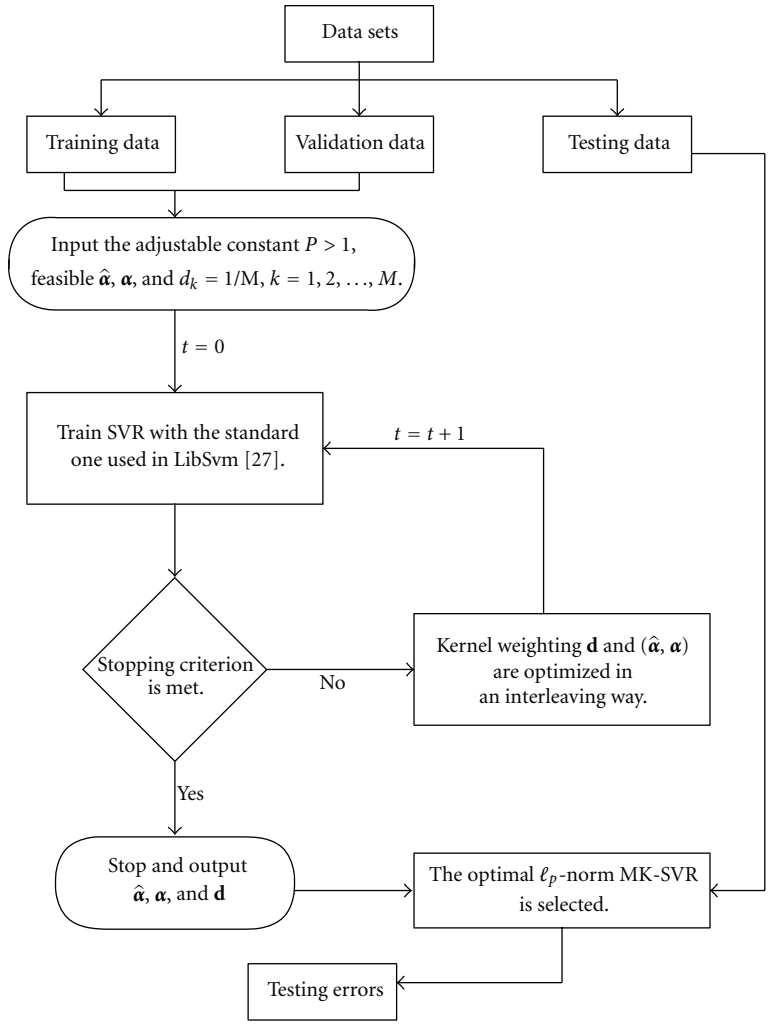

Figure 1: ℓ_p-norm MK-SVR model learning algorithm (see [27]).

Set the L's first partial derivatives with respect to d_k, and let it be 0:

$$\begin{aligned}\frac{\partial L}{\partial d_k} = 0 &\Longrightarrow \beta\left(\sum_k d_k^p\right)^{2/p-1} d_k^{p-1} \\ &= \gamma_k + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\hat{\alpha}_i - \alpha_i)\left(\hat{\alpha}_j - \alpha_j\right)K_k\left(\mathbf{x}_i, \mathbf{x}_j\right) \\ &\Longrightarrow \beta\left(\sum_k d_k^p\right)^{2/p} \\ &= \sum_k d_k\left(\gamma_k + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\hat{\alpha}_i - \alpha_i)\right. \\ &\qquad \left.\times\left(\hat{\alpha}_j - \alpha_j\right)K_k\left(\mathbf{x}_i, \mathbf{x}_j\right)\right).\end{aligned} \tag{9}$$

In the optimal point $\gamma_k = 0$ holds, so the previous equation yields

$$\begin{aligned}d_k = \frac{1}{2\beta}&\left(\sum_k\sum_{i=1}^{N}\sum_{j=1}^{N}(\hat{\alpha}_i - \alpha_i)\left(\hat{\alpha}_j - \alpha_j\right)K_k\left(\mathbf{x}_i, \mathbf{x}_j\right)\right)^{(1/q)-(1/p)} \\ &\times\left(\sum_{i=1}^{N}\sum_{j=1}^{N}(\hat{\alpha}_i - \alpha_i)\left(\hat{\alpha}_j - \alpha_j\right)K_k\left(\mathbf{x}_i, \mathbf{x}_j\right)\right)^{q/p},\end{aligned} \tag{10}$$

where $(1/p) + (1/q) = 1$, and $k = 1, 2, \ldots, M$.

In the second stage, the following algorithm is used. We give a chunking-based training algorithm (Algorithm 1) via analytical update for ℓ_p-Norm MK-SVR. Kernel weighting $\mathbf{d}$ and $(\hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha})$ are optimized in an interleaving way. The basic idea of this algorithm is to divide the optimal problem into an inner subproblem and an outer subproblem. The algorithm alternates between solving the two subproblems until convergence.

1. Input $f_{m,i} = \widehat{f}_i = \widehat{\alpha}_i = 0$, $g_{m,i} = \widehat{g}_i = \alpha_i = 0$, $\forall i = 1, 2, \ldots, N$; $L = S = -\infty$, $d_k = (\sqrt{1/k})^p$, $\forall k = 1, 2, \ldots, M$.
2. Iterate
   (1) Select $2Q$ variables based on the gradient of (6): $\widehat{\boldsymbol{\alpha}}_Q = \widehat{\alpha}_{i1}, \ldots, \widehat{\alpha}_{iQ}$, $\boldsymbol{\alpha}_Q = \alpha_{i1}, \ldots, \alpha_{iQ}$.
   (2) Store $\widehat{\boldsymbol{\alpha}}_Q^{\text{old}} = \widehat{\boldsymbol{\alpha}}_Q$, $\boldsymbol{\alpha}_Q^{\text{old}} = \boldsymbol{\alpha}_Q$ and new $\widehat{\boldsymbol{\alpha}}_Q$, $\boldsymbol{\alpha}_Q$ can be obtained according to (6) with respect to the selected variables.
   (3) Update gradient $f_{m,i} \leftarrow f_{m,i} + \sum_{q=1}^{Q} (\widehat{\alpha}_{iq} - \widehat{\alpha}_{iq}^{\text{old}}) K_k(\mathbf{x}_{iq}, \mathbf{x}_i)$, $g_{m,i} \leftarrow g_{m,i} + \sum_{q=1}^{Q} (\alpha_{iq} - \alpha_{iq}^{\text{old}}) K_k(\mathbf{x}_{iq}, \mathbf{x}_i)$, $\forall k = 1, 2, \ldots, M$, $i = 1, 2, \ldots, N$.
   (4) Compute the quadratic terms $S_k = (1/2) \sum_i (f_{m,i} - g_{m,i})(\widehat{\alpha}_i - \alpha_i)$, $\forall k = 1, 2, \ldots, M$.
   (5) $L_{\text{old}} = L$, $L = \sum_i y_i (\widehat{\alpha}_i - \alpha_i) - \sum_i \varepsilon (\widehat{\alpha}_i + \alpha_i)$, $S_{\text{old}} = S$, $S = \sum_k d_k S_k$.
   (6) If $|1 - (L - S)/(L_{\text{old}} - S_{\text{old}})| \geq \varepsilon$, update d_k with (10), $\forall k = 1, 2, \ldots, M$,
   else
      break
   endif
3. Output $\widehat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}, \mathbf{d}$.

Algorithm 1

In every iteration process, the inner subproblem ($\widehat{\boldsymbol{\alpha}}$ and $\boldsymbol{\alpha}$ step) identifies the constraint that maximises (6) with fixing kernel weighting $\mathbf{d}$. The outer subproblem ($\mathbf{d}$ step) is also called the restricted master problem. d_k is computed with the (10), $k = 1, 2, \ldots, M$.

The interleaved optimization algorithm is depicted in Algorithm 1, and the details of it are as follows.

2.2.1. Initialization. Assume the original values of $\widehat{\alpha}_i$ and α_i are 0, for all $i = 1, 2, \ldots, N$, and the initial value of d_k is $\sqrt{1/k}^p$, for all $k = 1, 2, \ldots, M$, where $p > 1$ is a constant.

2.2.2. Chunking and Carrying out with SVR. In the iteration process, the procedure is standard in chunking-based SVR solvers and is carried out by SVM$^{\text{light}}$, where Q is chosen as described in [28]. We implement the greedy second-order working set selection strategy of [28]. Rather than compute the gradient repeatedly, we speed up variable selection by caching, separately for each kernel. The cache needs to be updated every time we change $\widehat{\boldsymbol{\alpha}}_Q$ and $\boldsymbol{\alpha}_Q$ in the reduced variable optimisation. In Algorithm 1, (4) and (5) compute the objective values of SVR. Finally, the analytical value of $\mathbf{d}$ is carried out in (10).

2.2.3. Stopping Criterion. When the duality gap falls below a prespecified threshold, that is, $|1 - ((L - S)/(L_{\text{old}} - S_{\text{old}}))| < \varepsilon$, we terminate the algorithm and output $\widehat{\boldsymbol{\alpha}}$, $\boldsymbol{\alpha}$, $\mathbf{d}$.

3. Experimental Results

In this section, two experiments on a real financial time series have been carried out to assess the performance of ℓ_p-norm MK-SVR. The motivation behind the two experiments are to compare the performance of our proposed method with that of other methods, that is, single kernel support vector regression (SKSVR) [29] and ℓ_1-norm MK-SVR [22]. All calculations are performed with programs developed in MATLAB R2010a.

Table 1: The data sets for the first experiment.

Dataset	Training	Validating	Testing
data1	2003/1–2006/12	2007/1–2007/3	2007/4–2007/6
data2	2003/4–2007/3	2007/4–2007/6	2007/7–2007/9
data3	2003/7–2007/6	2007/7–2007/9	2007/10–2007/12

3.1. Experiment I. Firstly, we compare the performance of ℓ_p-norm MK-SVR with that of SKSVR. In this experiment, the daily stock closing prices of Shanghai Stock Index in China for the period of January 2003 to December 2007 are used, and the training/validating/testing data set is generated by a one-season moving-window testing approach. Following the way done in [29], three data sets, data1 to data3, are formed. For instance, data1 contains the daily stock closing prices from January 2003 to December 2006 are selected as the training data set, the daily stock closing prices from January 2007 to March 2007 are selected as the validating data set, the daily stock closing prices from April 2007 to June 2007 are selected as the testing data set. The corresponding time periods for data 1 to data 3 are listed in Table 1.

According to [29], we can derive training patterns $(\mathbf{x}_t, y_t)$ based on the original daily stock closing prices $\mathbf{P} = \{p_1, \ldots, p_t \ldots\}$ for SKSVR and ℓ_p-norm MK-SVR. Let $\text{EMA}_n(t) = \text{EMA}_n(t-1) + \alpha \times (p_t - \text{EMA}_n(t-1))$ be the n-day exponential moving average of the tth day, where p_t is the tth day daily stock closing prices and $\alpha = 2/(n+1)$, then the output variable y_t can be defined as

$$y_t = \text{RDP}_{+5}(t) = \frac{\text{EMA}_3 t - \text{EMA}_3(t-5)}{\text{EMA}_3(t-5)} \times 100. \quad (11)$$

Let $\mathbf{x}_t = (x_{t,1}, x_{t,2}, x_{t,3}, x_{t,4}, x_{t,5})$ be the input vector and let $\text{RDP}_{-n}(t) = (100 \times (p_t - p_{t-n}))/p_{t-n}$ be the lagged relative difference in percentage of price (RDP). Moreover, We can obtain a transformed closing price $\widetilde{\text{EWA}}_n(t)$ by subtracting a n-day EMA from the closing price, that is,

$$\widetilde{\text{EWA}}_n(t) = p_t - \text{EWA}_n(t). \quad (12)$$

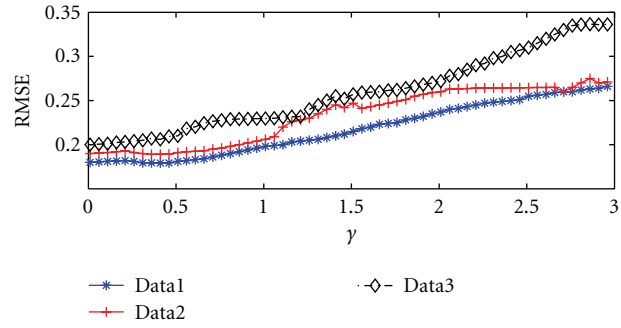

Figure 2: Forecasting performance of SKSVR with different hyperparameters.

Table 2: The comparison of RMSE values between SKSVR and ℓ_p-norm MK-SVR.

Methods	Data1	Data2	Data3
SKSVR	0.179	0.183	0.197
ℓ_p norm ($p = 1.05$)	**0.161**	0.177	0.186
ℓ_p norm ($p = 1.001$)	0.163	**0.174**	0.189
ℓ_p norm ($p = 1.15$)	0.166	0.179	**0.183**

Based on in the previously mentioned, the input variables can be defined as $x_{t,1} = \widetilde{\mathrm{EWA}}_{15}(t-5)$, $x_{t,2} = \widetilde{\mathrm{RDP}}_{-5}(t-5)$, $x_{t,3} = \widetilde{\mathrm{RDP}}_{-10}(t-5)$, $x_{t,4} = \widetilde{\mathrm{RDP}}_{-15}(t-5)$, and $x_{t,3} = \widetilde{\mathrm{RDP}}_{-20}(t-5)$. We adopt the root mean squared error (RMSE) for performance comparison, that is,

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2}, \tag{13}$$

where y_t and $\hat{y}_t$ are desired output and predicted output, respectively.

There are three parameters that should be determined in advance for SKSVR, that is, C, ε, and γ for using RBF kernel. The forecasting performance of SKSVR is examined with $C = 1$ and $\varepsilon = 0.005$. Because the forecasting performance obtained by SKSVR is effected by the parameter γ, we try with different settings of it from 0.01 to 3 with a stepping factor of 0.05. Figure 2 shows the RMSE for performance on the three data sets by SKSVR. The figure shows that SKSVR requires different γ settings for different data sets to obtain the best performance. For example, the best performance for data 1 occurs when $0.35 \le \gamma \le 0.45$. The best RMSE values obtained by SKSVR are listed in Table 2.

For ℓ_p-norm MK-SVR training model, we adopt RBF kernel $K(\mathbf{x}, \mathbf{x}_k) = \exp\{-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/\sigma^2\}$. A kernel combining 60 different RBF kernels is considered, that is, $0.01 \le 1/\sigma^2 \le 3$ with step 0.05. Hence, the kernel matrix is combined with a weighted sum of 60 kernel matrices, that is, $\widetilde{K} = d_1K_1 + d_2K_2 + \cdots + d_{60}K_{60}$ where d_1 denotes the kernel weight for the first kernel matrix with $1/\sigma^2 = 0.01$ and d_2 denotes the kernel weight for the second kernel matrix with $1/\sigma^2 = 0.06$, and so on. For the three data sets, the RMSE values obtained by ℓ_p-norm MK-SVR are listed in Table 2, too. Obviously when $p = 1.05$, 1.001, and 1.15, ℓ_p-norm MK-SVR model performs better than SKSVR one for data1 data set, data2 data set, and data3 data set, respectively.

Table 3: The data sets for the second experiment.

Dataset	Training	Validating	Testing
D-I	2008/1–2010/12	2011/1–2011/3	2011/4–2011/6
D-II	2008/4–2011/3	2011/4–2011/6	2011/7–2011/9
D-III	2008/7–2011/6	2011/7–2011/9	2011/10–2011/12

Table 4: The comparison of RMSE values between ℓ_1-norm MK-SVR and ℓ_p-norm MK-SVR.

Methods	D-I	D-II	D-III
ℓ_1 norm	0.182	0.189	0.178
ℓ_p norm ($p = 6/5$)	**0.175**	0.183	0.179
ℓ_p norm ($p = 4/3$)	0.185	**0.181**	0.180
ℓ_p norm ($p = 8/7$)	0.190	0.191	**0.171**

3.2. Experiment II. Secondly, we compare the performance of ℓ_p-norm MK-SVR with that of ℓ_1-norm MK-SVR. In this experiment, the daily stock closing prices of Shanghai Stock Index in China for the period of January 2008 to December 2011 are used, and the training/validating/testing data set is generated by a one-season moving-window testing approach. Following the way done in Tay and Cao [29], three data sets, D-I to D-III, are formed. The corresponding time periods for D-I to D-III are listed in Table 3.

We also adopt RMSE (13) for performance comparison. For ℓ_1-norm MK-SVR and ℓ_p-norm MK-SVR training model, a kernel combining 40 different RBF kernels is considered, that is, $1/\sigma^2 \in \{0.01, 0.02, \ldots, 0.09, 0.1, 0.2, \ldots, 0.9, 1, 2, \ldots, 9, 10, 20, \ldots, 100, 200, 300, 400\}$. Hence, the kernel matrix is combined with a weighted sum of 40 kernel matrices, that is, $\widetilde{K} = d_1K_1 + d_2K_2 + \cdots + d_{40}K_{40}$ where d_1 denotes the kernel weight for the first kernel matrix with $1/\sigma^2 = 0.01$ and d_2 denotes the kernel weight for the second kernel matrix with $1/\sigma^2 = 0.02$, and so on. For the three data sets, the RMSE values obtained by ℓ_1-norm MK-SVR and ℓ_p-norm MK-SVR are listed in Table 4. Obviously when $p = 6/5, 4/3$, and $8/7$, ℓ_p-norm MK-SVR model performs better than ℓ_1-norm MK-SVR one for D-I data set, D-II data set, and D-III data set, respectively. Figure 3 shows the forecasting results for D-I and D-II by the two regression models.

Furthermore, we can use a statistical test proposed by Diebold and Mariano [30] to assess the statistical significance of the forecasts by ℓ_p-norm MK-SVR model. The loss-differential series of ℓ_1-norm MK-SVR and ℓ_p-norm MK-SVR are shown in Figures 4 and 5. According to [30], we adopt the asymptotic test $S_1 = \bar{d}/\sqrt{(2\pi\hat{f}_d(0))/T}$ as the test statistic, where $d_i = r_{1i}^2 - r_{2i}^2$ is the loss-differential series of ℓ_1-norm MK-SVR and ℓ_p-norm MK-SVR models, r_1 and r_2 denote the forecasting errors; $2\pi\hat{f}_d(0)$ is the weighted sum of the available sample autocovariances:

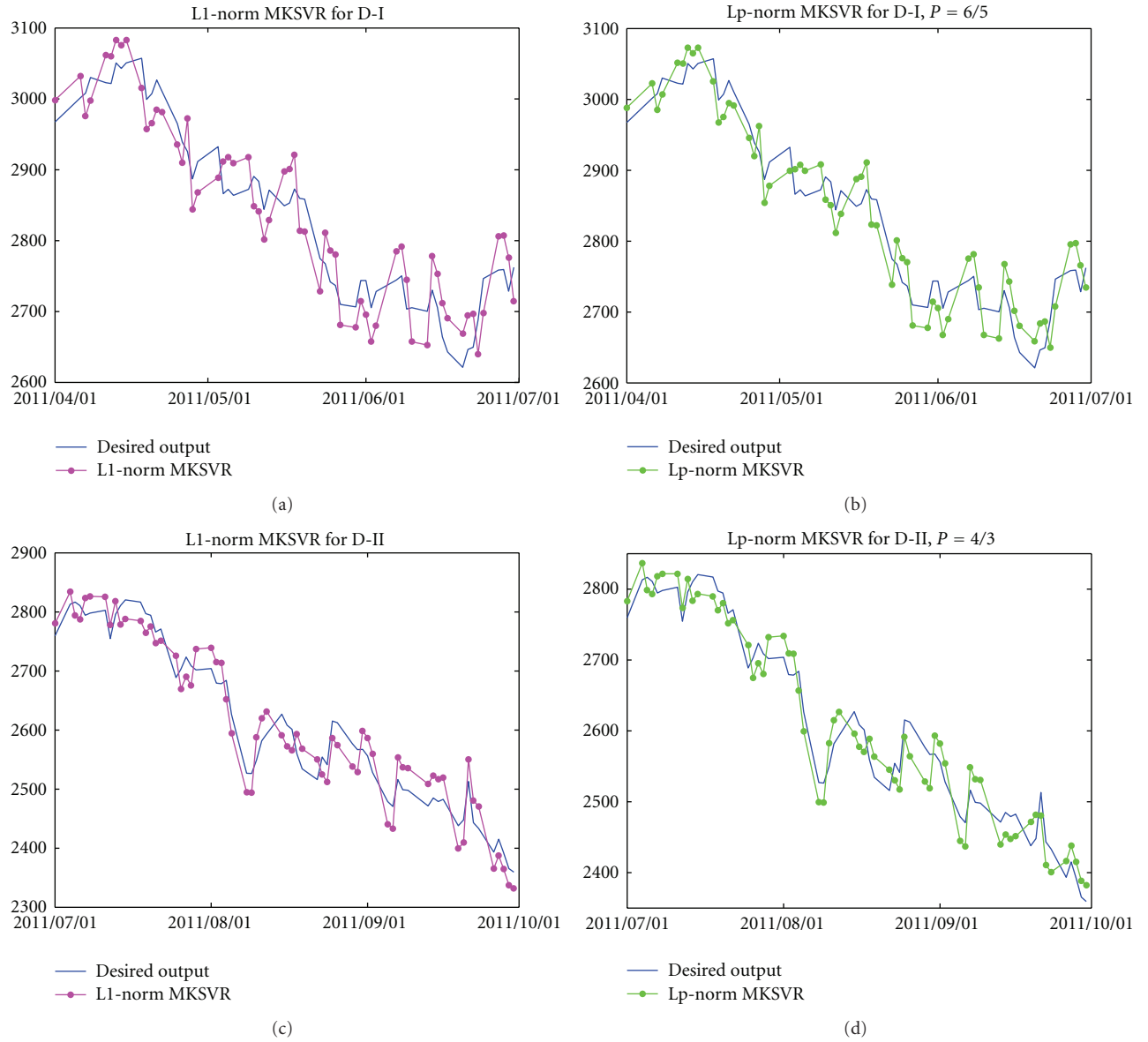

Figure 3: Forecasting results by ℓ_1-norm MK-SVR and ℓ_p-norm MK-SVR.

$2\pi \hat{f}_d(0) = \sum_{\tau=-(T-1)}^{T-1} 1 * (\tau/S(T))\hat{\gamma}_d(T)$, where T is the sample size, $\hat{\gamma}_d(T) = (1/T)\sum_{t=|\tau|+1}^{T}(d_t - \overline{d})(d_{t-|\tau|} - \overline{d})$, and $1 * (\tau/S(T))$ is the lag window, defined as

$$1 * \left(\frac{\tau}{S(T)}\right) = \begin{cases} 1, & \text{if } \left|\frac{\tau}{S(T)}\right| \leq 1, \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

where $S(T) = k - 1$; k reports the number of forecasting steps ahead.

We denote U_1 as the forecasting accuracy of ℓ_1-norm MK-SVR and U_p as the forecasting accuracy of ℓ_p-norm MK-SVR. Under the null hypothesis: $U_1 = U_p$, the test was performed at the 0.05 and 0.10 significant levels [12]. The test results are shown in the following Table 5. For the three data sets, all asymptotic tests reject $H_0 : U_1 = U_p$. The test result shows that ℓ_p-norm MK-SVR model indeed improves the forecasting accuracy in comparison with ℓ_1-norm MK-SVR model.

Table 5: Asymptotic test.

Stock closing prices	$\alpha = 0.05$	$\alpha = 0.10$
D-I	$S_1 = 1.756$, P value = 0.0359	$S_1 = 1.756$, P value = 0.0359
D-II	$S_1 = 1.832$, P value = 0.0416	$S_1 = 1.832$, P value = 0.0416
D-III	$S_1 = 1.579$, P value = 0.0258	$S_1 = 1.579$, P value = 0.0258

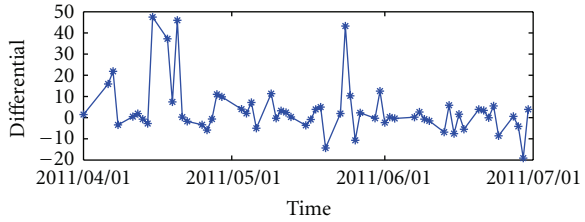

Figure 4: Loss differential (ℓ_1-MKSVR to ℓ_p-MKSVR) of D-I.

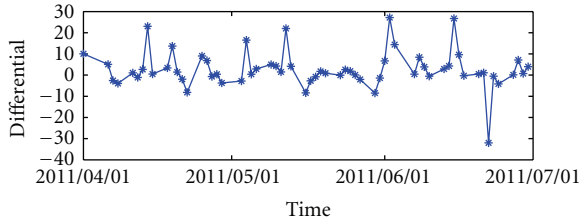

Figure 5: Loss differential (ℓ_1-MKSVR to ℓ_p-MKSVR) of D-II.

We briefly mention that the superior performance of ℓ_p-norm MK-SVR model ($p > 1$) is not surprising. When we use the sparsity-inducing norm ($p = 1$), some of the kernel weights are forced to become zero, and the corresponding kernel will be eliminated leading to some information loss. The daily stock closing prices do not carry large parts of overlapping information, and the information is discriminative. So a nonsparse kernel mixture can access more information and perform more robustly.

4. Summary and Prospect

In this paper, an ℓ_p-norm MK-SVR model for stock market price forecasting is proposed. The model conceives an optimization scheme of unprecedented efficiency and provides a really efficient implementation. In an empirical evaluation, we show that ℓ_p-norm MK-SVR can improve predictive accuracies on relevant real-world data sets. Although we focus on volatility forecasting of stock markets in this paper, our ℓ_p-norm MK-SVR model could be applied to more general financial forecasting problems. Therefore in the future we will apply our ℓ_p-norm MK-SVR model for other financial markets, such as exchange markets.

Appendix

ℓ_p-Norm MK-SVR Dual Formulation

In this appendix, we detail the dual formulation of ℓ_p-norm MK-SVR. We again consider ℓ_p-norm MK-SVR with a general convex loss,

$$\begin{aligned}
\min_{\mathbf{w},b} \quad & \frac{1}{2}\sum_{k}\frac{\|\mathbf{w}_k\|_2^2}{d_k} + C\sum_{i=1}^{l}\left(\xi_i + \hat{\xi}_i\right) \\
\text{s.t.} \quad & \left(\langle \mathbf{w}, K(\mathbf{x}_i)\rangle + b\right) - y_i \le \varepsilon + \xi_i, \\
& y_i - \left(\langle \mathbf{w}, K(\mathbf{x}_i)\rangle + b\right) \le \varepsilon + \hat{\xi}_i, \\
& \|\mathbf{d}\|_p^2 \le 1, \\
& d_1, d_2, \ldots, d_M \ge 0, \\
& \xi, \hat{\xi}_i \ge 0, \quad i = 1, 2, \ldots, N.
\end{aligned} \tag{A.1}$$

In the following, we build the Lagrangian of (A.1). By introducing Lagrangian multipliers $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_N)^T \in \mathbf{R}^N$, $\hat{\boldsymbol{\alpha}} = (\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_N)^T \in \mathbf{R}^N$, $\beta \in \mathbf{R}_+$, and $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_M)^T \in \mathbf{R}^M$, the Lagrangian saddle point problem is given by

$$\begin{aligned}
\sup_{\substack{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}},\eta_1,\eta_2 \\ \beta\ge 0,\gamma\ge 0}} \inf_{\mathbf{w},b} \Bigg\{ & \frac{1}{2}\sum_{k}\frac{\|\mathbf{w}_k\|_2^2}{d_k} + C\sum_{i=1}^{l}\left(\xi_i + \hat{\xi}_i\right) \\
& + \sum_{i=1}^{N}\alpha_i\left(\langle \mathbf{w}, K(\mathbf{x}_i)\rangle + b - y_i - \varepsilon - \xi_i\right) \\
& + \sum_{i=1}^{N}\hat{\alpha}_i\left(y_i - \langle \mathbf{w}, K(\mathbf{x}_i)\rangle - b - \varepsilon - \hat{\xi}_i\right) \\
& - \boldsymbol{\gamma}^T\mathbf{d} - \eta_1\xi - \eta_2\hat{\xi} + \beta\left(\frac{1}{2}\|\mathbf{d}\|_p^2 - \frac{1}{2}\right)\Bigg\}.
\end{aligned} \tag{A.2}$$

Set the Lagrangian's first partial derivatives with respect to $\mathbf{w}$, b, ξ_i, and $\hat{\xi}_i$, and let them be 0 to reveal the optimality conditions

$$\begin{aligned}
& \mathbf{w}_k = d_k\sum_{i=1}N(\hat{\alpha}_i - \alpha_i)K_k(\mathbf{x}_i), \quad k = 1, 2, \ldots, M, \\
& \mathbf{1}^T(\hat{\alpha}_i - \alpha_i) = 0, \quad \mathbf{1} = (1, 1, \ldots, 1)^T, \\
& C = \eta_{1i} + \alpha_i, \\
& C = \eta_{2i} + \hat{\alpha}_i, \quad i = 1, 2, \ldots, N.
\end{aligned} \tag{A.3}$$

Resubstituting the previous equations to the Lagrangian yields the following

$$\begin{aligned}
\sup_{\substack{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}},\beta\ge 0,\gamma\ge 0, \\ \mathbf{1}^T(\hat{\alpha}_i-\alpha_i)=0}} \inf_{\varepsilon,\mathbf{d}} \Bigg\{ & \sum_{i=1}^{N}y_i(\hat{\alpha}_i - \alpha_i) - \varepsilon\sum_{i=1}^{N}(\hat{\alpha}_i + \alpha_i) - \frac{1}{2}\sum_{k=1}^{M}d_k \\
& \cdot \sum_{i=1}^{N}\sum_{j=1}^{N}(\hat{\alpha}_i - \alpha_i)\left(\hat{\alpha}_j - \alpha_j\right)K_k\left(\mathbf{x}_i, \mathbf{x}_j\right) \\
& + \beta\left(\frac{1}{2}\|\mathbf{d}\|_p^2 - \frac{1}{2}\right) - \boldsymbol{\gamma}^T\mathbf{d}\Bigg\}
\end{aligned} \tag{A.4}$$

which can also be written as

$$\sup_{\substack{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}},\beta\geq 0,\gamma\geq 0,\\ \mathbf{1}^T(\hat{\alpha}_i-\alpha_i)=0}} \left\{ \sup_{\varepsilon} \left\{ \varepsilon \sum_{i=1}^{N}(\hat{\alpha}_i+\alpha_i) - \sum_{i=1}^{N} y_i(\hat{\alpha}_i-\alpha_i) \right\} \right.$$

$$\left. - \beta \sup_{\mathbf{d}} \left\{ \frac{1}{\beta} \sum_{k=1}^{M} \left(\frac{1}{2} d_k \sum_{i=1}^{N}\sum_{j=1}^{N} (\hat{\alpha}_i-\alpha_i)\left(\hat{\alpha}_j-\alpha_j\right) K_k\left(\mathbf{x}_i,\mathbf{x}_j\right) + \gamma_k \right) - \frac{1}{2}\|\mathbf{d}\|_p^2 \right\} - \frac{1}{2}\beta \right\}. \tag{A.5}$$

For standard support vector regression formulations, the hinge loss function can be defined as $f(\mathbf{w};(\mathbf{x},y)) = \max\{0, |\langle \mathbf{w},\mathbf{x}\rangle - y| - \varepsilon\}$. This loss is also convex with a subgradient bounded by $\|\mathbf{x}\|$. As is known to all, the Fenchel-Legendre conjugate of a function f is defined as $f^*(x) = \sup_u x^T u - f(u)$, and the dual form is denoted by $\|\cdot\|_*$ (the norm defined via the identity $(1/2)\|\cdot\|_*^2 = ((1/2)\|\cdot\|^2)^*$. According to (A.3), (A.5), and Fenchel-Legendre conjugate of the hinge loss function, we can obtain the following dual:

$$\max_{\hat{\boldsymbol{\alpha}},\boldsymbol{\alpha},\beta\geq 0,\gamma\geq 0} \left(\mathbf{y}^T(\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha}) - \boldsymbol{\varepsilon}(\hat{\boldsymbol{\alpha}}+\boldsymbol{\alpha})\right)$$

$$- \frac{1}{\beta} \left\| \left(\frac{1}{2} d_k \sum_{i=1}^{N}\sum_{j=1}^{N} (\hat{\alpha}_i-\alpha_i)\left(\hat{\alpha}_j-\alpha_j\right) K_k\left(\mathbf{x}_i,\mathbf{x}_j\right) + \gamma_k \right)_{k=1}^{M} \right\|_{p^*}^2$$

$$- \frac{1}{2}\beta, \tag{A.6}$$

where $\mathbf{y} = (y_1, y_2, \ldots, y_N)^T$, $\boldsymbol{\varepsilon} = (1,1,\ldots,1)^T$, $\mathbf{1}^T(\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha}) = 0$, $0 \leq \hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \leq C1$, and $p^* = p/(p-1)$.

In the following, we find $\hat{\beta}$ at optimality. Let us solve $\partial L/\partial\beta = 0$ for the unbounded β; then we can obtain the optimal β as

$$\hat{\beta} = \left\| \left(\frac{1}{2} d_k \sum_{i=1}^{N}\sum_{j=1}^{N} (\hat{\alpha}_i-\alpha_i)\left(\hat{\alpha}_j-\alpha_j\right) K_k\left(\mathbf{x}_i,\mathbf{x}_j\right) + \gamma_k \right)_{k=1}^{M} \right\|_{p^*}. \tag{A.7}$$

Obviously, $\hat{\beta} \geq 0$, so we can ignore the corresponding constraint from the optimization problem and plug (A.7) into (A.6). Then the following dual optimization problem for ℓ_p-norm MK-SVR is written as

$$\max_{\hat{\boldsymbol{\alpha}},\boldsymbol{\alpha},} \left(\mathbf{y}^T(\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha}) - \boldsymbol{\varepsilon}(\hat{\boldsymbol{\alpha}}+\boldsymbol{\alpha})\right)$$

$$- \left\| \left(\frac{1}{2} d_k \sum_{i=1}^{N}\sum_{j=1}^{N} (\hat{\alpha}_i-\alpha_i)\left(\hat{\alpha}_j-\alpha_j\right) K_k\left(\mathbf{x}_i,\mathbf{x}_j\right) + \gamma_k \right)_{k=1}^{M} \right\|_{p^*}$$

$$\text{s.t.} \quad \mathbf{1}^T(\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha}) = 0,$$

$$0 \leq \hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \leq C1,$$

$$d_1, d_2, \ldots, d_M \geq 0. \tag{A.8}$$

For the choice of ℓ_p norm, $\gamma = 0$ holds in the optimal point so that the γ-term can be discarded [23]. Therefore the previous equations reduce to an optimization problem that depends on $\hat{\boldsymbol{\alpha}}$ and $\boldsymbol{\alpha}$ as

$$\max_{\hat{\boldsymbol{\alpha}},\boldsymbol{\alpha},\gamma\geq 0} \left(\mathbf{y}^T(\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha}) - \boldsymbol{\varepsilon}(\hat{\boldsymbol{\alpha}}+\boldsymbol{\alpha})\right)$$

$$- \left\| \left(\frac{1}{2} d_k \sum_{i=1}^{N}\sum_{j=1}^{N} (\hat{\alpha}_i-\alpha_i)\left(\hat{\alpha}_j-\alpha_j\right) K_k\left(\mathbf{x}_i,\mathbf{x}_j\right) \right)_{k=1}^{M} \right\|_{p^*}$$

$$\text{s.t.} \quad \mathbf{1}^T(\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha}) = 0,$$

$$0 \leq \hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \leq C1,$$

$$d_1, d_2, \ldots, d_M \geq 0. \tag{A.9}$$

Now, ℓ_p-norm MK-SVR model has been constructed.

Acknowledgments

The authors would like to thank the handling editor and the anonymous reviewers for their constructive comments, which led to significant improvement of the paper. This work was partially supported by the National Natural Science Foundation of China under Grant no. 51174236.

References

[1] J. W. Hall, "Adaptive selection of U.S. stocks withneural nets," in *Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Nancial Markets*, G. J. Deboeck, Ed., John Wiley & Sons, New York, NY, USA, 1994.

[2] Y. S. Abu-Mostafa and A. F. Atiya, "Introduction to financial forecasting," *Applied Intelligence*, vol. 6, no. 3, pp. 205–213, 1996.

[3] D. G. Champernowne, "Sampling theory applied to autoregressive schemes," *Journal of the Royal Statistical Society B*, vol. 10, pp. 204–231, 1948.

[4] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, Prentice Hall, Englewood Cliffs, NJ, USA, 3rd edition, 1994.

[5] J. Yao and C. L. Tan, "A case study on using neural networks to perform technical forecasting of forex," *Neurocomputing*, vol. 34, pp. 79–98, 2000.

[6] L. Cao and F. E. H. Tay, "Financial forecasting using support vector machines," *Neural Computing and Applications*, vol. 10, no. 2, pp. 184–192, 2001.

[7] L. J. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1506–1518, 2003.

[8] M. Qi and Y. Wu, "Nonlinear prediction of exchange rates with monetary fundamentals," *Journal of Empirical Finance*, vol. 10, no. 5, pp. 623–640, 2003.

[9] P. F. Pai and C. S. Lin, "A hybrid ARIMA and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497–505, 2005.

[10] P. F. Pai, W. C. Hong, C. S. Lin, and C. T. Chen, "A hybrid support vector machine regression for exchange rate prediction,"

International Journal of Information and Management Sciences, vol. 17, no. 2, pp. 19–32, 2006.

[11] Y. K. Kwon and B. R. Moon, "A hybrid neurogenetic approach for stock forecasting," *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 851–864, 2007.

[12] W. M. Hung and W. C. Hong, "Application of SVR with improved ant colony optimization algorithms in exchange rate forecasting," *Control and Cybernetics*, vol. 38, no. 3, pp. 863–891, 2009.

[13] H. Jiang and W. He, "Grey relational grade in local support vector regression for financial time series prediction," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2256–2262, 2012.

[14] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.

[15] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel- Based Learning Methods*, Cambridge University Press, Cambridge, UK, 2000.

[16] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1–3, pp. 131–159, 2002.

[17] K. Duan, S. S. Keerthi, and A. N. Poo, "Evaluation of simple performance measures for tuning SVM hyperparameters," *Neurocomputing*, vol. 51, pp. 41–59, 2003.

[18] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[19] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[20] F. R. Bach, "Consistency of the group lasso and multiple kernel learning," *Journal of Machine Learning Research*, vol. 9, pp. 1179–1225, 2008.

[21] D. Zhang, D. Shen, and The Alzheimer's Disease Neuroimaging Initiative, "Multimodal multi-task learning for joint prediction of multiple regression and classification variables in Alzheimer's disease," *NeuroImage*, vol. 59, pp. 895–907, 2012.

[22] C. Y. Yeh, C. W. Huang, and S. J. Lee, "A multiple-kernel support vector regression approach for stock market price forecasting," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2177–2186, 2011.

[23] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "ℓ_p-norm multiple kernel learning," *Journal of Machine Learning Research*, vol. 12, pp. 953–997, 2011.

[24] F. Orabona, L. Jie, and B. Caputo, "Multi kernel learning with online-batch optimization," *Journal of Machine Learning Research*, vol. 13, pp. 165–191, 2012.

[25] A. Zien and C. S. Ong, "Multiclass multiple kernel learning," in *Proceedings of the 24th International Conference on Machine Learning (ICML'07)*, pp. 1191–1198, June 2007.

[26] S. V. N. Vishwanathan, Z. Sun, N. Theera-Ampornpunt, and M. Varma, "Multiple kernel learning and the SMO algorithm," in *Advances in Neural Information Processing Systems*, 2010.

[27] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," in *ACM Transactions on Intelligent Systems and Technology (TIST '11)*, vol. 2, no. 3, pp. 1–27, ACM, 2011.

[28] R. E. Fan, P. H. Chen, and C. J. Lin, "Working set selection using second order information for training support vector machines," *Journal of Machine Learning Research*, vol. 6, pp. 1889–1918, 2005.

[29] F. E. H. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *Omega*, vol. 29, no. 4, pp. 309–317, 2001.

[30] F. X. Diebold and R. S. Mariano, "Comparing predictive accuracy," *Journal of Business and Economic Statistics*, vol. 20, no. 1, pp. 134–144, 2002.