Multiple Kernel Support Vector Regression with Higher Norm in Option Pricing

Neetu Verma
Department of Mathematics and Computer Applications, MANIT, Bhopal, Madhya Pradesh, India.

Sujoy Das
Department of Mathematics and Computer Applications, MANIT, Bhopal, Madhya Pradesh, India.

Namita Srivastava
Department of Mathematics and Computer Applications, MANIT, Bhopal, Madhya Pradesh, India.

Abstract
The purpose of present study is to investigate a non-parametric model that improves accuracy of option prices found by previous models. In this study option prices are calculated using multiple kernel Support Vector Regression with different norm values and their results are compared. $L_1$-norm multiple kernel learning Support Vector Regression (MKLSVR) has been successfully applied to option prices forecasting. The advantages of $L_1$-norm multiple kernel learning is that it allows setting of kernel hyperparameter automatically which is one of the most tedious task. $L_1$-norm MKLSVR normally outperforms under all market conditions than single kernel Support Vector Regression. To further minimize forecasting error we adopt $L_p$-norm multiple kernel Support Vector Regression ($p > 1$) as it generalize well under all market conditions. In $L_p$-norm multiple kernel Support Vector Regression (MKLSVR), optimization part is solved using the Sequential Minimal Optimization algorithm (SMO). The $L_p$-norm SMOMKL SVR is evaluated on forecasting the option prices of European-style Nifty index options in India. Experimental results show that $L_p$-norm SMOMKL SVR performs better than $L_1$-norm MKL SVR with different methods.

Keywords: $L_p$-Norm Multiple Kernel Learning Support Vector Regression (SMOMKL SVR), option pricing, Smomkl.

Introduction
Financial derivative forecasting is an appealing yet difficult task in computational financial world. It follows a complex pattern and a stochastic behavior in stock price resulting highly nonlinear option pricing functions therefore accurate forecasting of any options is a significance issue for traders and investors in the financial markets. ‘Option’ which is the right (not the obligation) to buy (call option) or sell (put option) the underlying asset at a particular price i.e. strike price in the future. Options can reduce the financial risk on the future events [1]. To purchase any option all traders pay an amount of money called premium or option price. After the expiration of an option its premium is worthless and no one can exercised it.

As described by J.C. Hull [1], the option pricing is nonstationary, follows log normally distribution and are inherently noisy. Since 1970 many methods were adopted for dealing this type of problem. Black and Scholes proposed a formula for pricing option in 1973. As a result, pricing of option became easier and efficient than before. Later on traders have been realized that the assumptions in Black-Scholes do not agree with the real world finance. Ever since then many modifications on Black-Scholes formula has been proposed and numerous pricing models have been created and studied to find option prices [1]. It is difficult to justify certain assumptions for different parametric specification in the real world data as there exists nonlinear relationship between option price and various variables. Therefore in recent years, many researchers turned in to machine learning or nonparametric methods as they are capable to capture nonlinear relationship between input and output [2, 3].

Related Work
In this paper machine learning methods are used for forecasting the price of Nifty call option. There are different types of machine learning methods, which have been applied on option pricing. Hutchinson[2] was first to apply Artificial Neural Networks (ANNs) on American-style on S&P500 futures call options and compared results with the Black–Scholes model and found that ANNs are better than Black–Scholes. Lajbcygier, et al. [4] improved the Hybrid Neural Network using bootstrap methods to reduce bias in existing model. Panayiotis C. Andreou et al. [3] used another hybrid ANNs in which the target function is the residual between the actual call market price and resultant option prices of Black–Scholes or Corrado and Su models. He found that hybrid ANN model provides greater predictability than traditional approaches such as Black–Scholes model. M. M. Pires et al.
[5] compared the performance of a Multi-Layer Perceptron neural network and a Support Vector Regression in pricing American styled options. It was concluded that a Support Vector Regression approach provided promising results than that found with Multi-Layer Perceptron. Shian-Chang Huang et al. [6] combined the unscented Kalman filters (UKFs) and Support Vector Regression (SVR) to predicting option prices. The difference between the market option prices and the Black-Scholes option price is taken input to SVR for reducing the prediction errors. The performance of the new hybrid model is better than pure SVR models or UKFs models in option pricing. Saxena [7] studied European-style CNX Nifty Options traded at National Stock Exchange of India. He combined the BS model and Artificial Neural Networks (ANNs), for option pricing and concluded that hybrid model can improve the pricing performance of options under all market conditions. Ping Wang et al. [8] used Support Vector Regression (SVR) integrated with stochastic volatility models [9] for forecasting of currency option pricing. The results reveal that integrated model performed better than traditional approaches such as Garman-Kohlhagen Formula (GK) model and ANN Option pricing model. Panayiotis C. Andreou et.al. [10] used Support Vector Regression and Least Squares Support Vector Regression for pricing S&P 500 index call options with Deterministic Volatility Functions approach and compared results with the traditional Black Scholes model. He obtained promising results for the both SVR models. L. Xun, et al. [11] gave some modifications on three parametric methods, the binomial tree method, the finite difference method and the Monte Carlo method, to forecast the option prices and further refined the forecast results by nonparametric methods ANN and SVR by decreasing the nonlinear errors. He found that, compared with the standard and improved parametric option price forecasting methods, the ANN and SVR have higher forecasting accuracy. Chih-Ming Hsu et al. [12] compared the price of Taiwan Stock Exchange Capitalization Weighted Stock Index Options (TAIEX Options) by three approaches i.e. Black-Scholes (BS) model, Genetic Programming (GP) and Support Vector Regression (SVR) with all basic factors in the B-S model and the other factors in GP and SVR model. They concluded that, both GP and SVR forecasting models gave more promising results than Black-Scholes model. Above literature implied that there are so many researches, in which Support Vector Regression approach provided promising results than that found with Multi-Layer Perceptron.

Later on Sören Sonnenburg et al. [19] redefined MKL as a semi-infinite linear program using chunking algorithm to optimize the SVM multipliers and the kernel coefficients at the same time. That also can be generalizing on large problems, for regression and classification. Rakotomamonjy et al. [20] solve the MKL problem by iteratively using the sequential minimization optimization algorithm [18] to update lagrange multipliers and reduced gradient algorithm for kernel weights. This method for MKL is known as Simple MKL. Corinna Cortes [21] studies nonlinear kernel combinations based on a polynomial base kernels in the case of regression and the kernel ridge regression algorithm with projection-based gradient descent algorithm for solving the optimization problem and shown the effectiveness of existing algorithm.

When MKL problem is defined on the bases of norm it is called Lp-norm MKL problem. If the value of p=1 then L∞-norm of the kernel weights is also known as the simplex constraint, is mostly used in MKL methods [22]. Most MKL methods employ the Lp-norm for linear kernel combination weights [20]. These methods therefore represent a sparse kernel combination (i.e. only some basis kernels among many have significant weights) although it has good interpretability in kernel selection but non-smooth kernel combinations [22]. As pointed out by researchers [19, 23] that the kernels encode orthogonal or complementarily information, L1 norm MKL may lead to discarding useful information and may thus result in poor generalization performance due to sparse representation of kernel weights. Therefore they extended MKL problem to arbitrary norms means Lp-norms with p ≥ 1. Also he developed new insights on the connection between several existing MKL formulations and formulate them for arbitrary norms, this non-sparse MKL problem gave better accuracies than sparse MKL problem.

In order to improve the efficiency of L1 norm MKL of kernel weights, researchers tend to increase norm value for good generalization performance. Alternatively, MKL with Lp-norm constraint (p > 1) of the kernel weights gave robust results against noisy and redundant feature sets and keep all the information in the base kernels compared to L1-norm [23]. This significantly improves the predicting performance by leading non-sparse solutions, but higher value of p>1 promotes the risk of being sensitive to noise and incorporating redundant information from data [9]. Due to this fact the value of p varies from 1 to 5.

Now SMOMKL, proposed by S. V. N. Vishwanathan et.al. is used in option pricing frame work with sufficiently large value of p. Moreover, the SMOMKL regularised by the Bregman divergence, which is a convex optimization problem is simple, easy to implement and applicable on real world applications. Also it is significantly faster than other Lp-norm MKL solvers [9]. In this paper optimization part is solved by three methods for all type of datasets which have been taken to find out the effectiveness and efficiency of SMOMKL. Performance of SMOMKL is compared with reduced gradient descent method [20] and mirror descent method [24].

The paper is structured as follows. First introduce the theoretical analysis of Lp-norm and L1-norm multiple kernel Support Vector Regression then data preparation, parameter setting, and performance criteria are given. In second last section, the performance of MKLSVR with different
algorithm on each type of option pricing datasets is presented. Conclusion is presented in last section.

**Lp-Norm Multiple Kernel Learning Support Vector Regressions**

In this paper Lp-norm SMOMKL Support Vector Regression [9] is used for option pricing. It jointly learns kernel and SVM parameters from given set of data in option pricing. Let \((x_i, y_i)\) be data set with \(i=1,2, \ldots, N\), \(x_i = \{x_i^1, x_i^2\}\) here \(x_i^1\) represents time to maturity and \(x_i^2\) represents moneyness ratio of option pricing data. \(y_i\) is closing price of Nifty index options. Let \(K\) be given set of \(M\) base kernels correspond the feature maps \(\{\phi_k\}\), where \(k=1,2, \ldots, M\). We know that a variable has lognormal distribution if the natural logarithm of that variable is normally distributed. It is well known that stock price follows geometric Brownian motion [1]. The most widely used distribution in machine learning and statistics is the Gaussian or normal distribution. Hence selecting Radial or Gaussian basis kernel function for regression is better than others because it is easy to interpret and capture some of the most basic properties of a distribution, namely its mean and variance also it makes the least number of assumptions subject to the constraint of having a specified mean and variance [25].

Now consider the linear combination of \(M\) base kernels

\[
K = \sum_{k=1}^{M} d_k K_k
\]  

The primal can therefore be formulated as

\[
P_k = \min_{w,b} \frac{1}{2} \sum_{k=1}^{M} w_k^2 + c \sum_{i=1}^{N} (\xi_i + \xi_i^*) + \frac{\lambda}{2} \sum_{k=1}^{M} d_k^2
\]

Such that

\[
y_i - \sum_{k=1}^{M} w_k \phi_k (x_i) - b \leq \varepsilon + \xi_i
\]

\[
\sum_{k=1}^{M} w_k \phi_k (x_i) + b - y_i \leq \varepsilon + \xi_i^*
\]

\[
\xi_i, \xi_i^* \geq 0, d_1, d_2, \ldots, d_M \geq 0
\]

After differentiating with respect to \(w, b, \xi, \xi^*\) to get intermediate saddle point problem

\[
\min_{d \in \mathbb{R}^M, \alpha \in \mathbb{R}^N} L(y \alpha - \varepsilon \alpha) = \frac{1}{2} \sum_{k=1}^{M} d_k^2 + \lambda \sum_{k=1}^{M} \alpha_k K_k \alpha - \frac{1}{2} \sum_{k=1}^{M} \rho\gamma_k d_k
\]

Where \(Y = \text{Diagonal matrix with numeric values on the diagonal and all} \alpha_i \in [0, 1]\)

Again taking the Lagrangian of above equation and solving for \(d_k\),

\[
L = (y \alpha - \varepsilon \alpha) - \frac{1}{2} \sum_{k=1}^{M} d_k^2 + \frac{1}{2} \sum_{k=1}^{M} \alpha_k K_k \alpha - \frac{1}{2} \sum_{k=1}^{M} \rho\gamma_k d_k
\]

\[
\frac{\partial L}{\partial d_k} = 0 \Rightarrow \lambda (\sum_{k=1}^{M} d_k^2)^{\frac{2}{p-1}} - \gamma_k - \frac{1}{2} \alpha K_k \alpha = 0
\]

\[
\Rightarrow \lambda (\sum_{k=1}^{M} d_k^p)^{\frac{2}{p}} = \sum_{k=1}^{M} d_k (\gamma_k + \frac{1}{2} \alpha K_k \alpha)
\]

Since \(K\) is positive semi-definite, \(\alpha K_k \alpha \geq 0\)

and since \(\alpha \geq 0\) it is clear that the optimal value of \(\gamma_k\) is zero. Substitute above values in Lagrangian equation to get Lp-norm SMOMKL dual problem, assuming

\[
\frac{1}{P} \geq \frac{1}{q} = 1
\]

\[
D = \max_{\alpha \in \mathbb{R}^N} \frac{1}{P} \sum_{k=1}^{M} \left(\alpha_1 K_k \alpha_1\right)^{\frac{1}{q} - \frac{1}{p}} \sum_{k=1}^{M} \left(\alpha_1 K_k \alpha_1\right)^{\frac{2}{q}}
\]

All kernel weights can be found from the dual variables defined as

\[
d_k = \frac{1}{2} \lambda \sum_{i=1}^{M} \left(\alpha_1 K_k \alpha_1\right)^{\frac{1}{q} - \frac{1}{p}} \sum_{k=1}^{M} \left(\alpha_1 K_k \alpha_1\right)^{\frac{2}{q}}
\]

Above dual objective \(D\) is differentiable with respect to \(\alpha\). SMO algorithm is applied by selecting two variables at a time and optimizing using gradient or Newton methods. SMO-MKL algorithm terminate when the duality gap falls below a pre-specified threshold or until it converges.

**L1-Norm Multiple Kernel Learning Support Vector Regressions**

When L1-norm using in view of MKLSVR the primal can defined as

\[
P_k = \min_{w,b} \frac{1}{2} \sum_{k=1}^{M} w_k^2 + c \sum_{i=1}^{N} (\xi_i + \xi_i^*)
\]

Such that

\[
y_i - \sum_{k=1}^{M} w_k \phi_k (x_i) - b \leq \varepsilon + \xi_i
\]

\[
\sum_{k=1}^{M} w_k \phi_k (x_i) + b - y_i \leq \varepsilon + \xi_i^*
\]

\[
\xi_i, \xi_i^* \geq 0, d_1, d_2, \ldots, d_M \geq 0
\]

\[
\sum_{k=1}^{M} d_k = 1
\]

After introducing Lagrangian multipliers the dual can be defined as in matrix notation

\[
\min_{\lambda, \alpha} \max_{\varepsilon} \frac{1}{P} \sum_{k=1}^{M} \left(\alpha_1 K_k \alpha_1\right)^{\frac{1}{q} - \frac{1}{p}} \sum_{k=1}^{M} \left(\alpha_1 K_k \alpha_1\right)^{\frac{2}{q}}
\]

Thus, by simply differentiating the above dual function with respect to \(d_k\), we have

\[
\frac{\partial L}{\partial d_k} = -\frac{1}{2} \alpha' K_k \alpha
\]
Then Reduced Gradient algorithm [20], Mirror Descent based algorithm [24] and line-search for choosing step-sizes [20] are applied to solve MKLSVR problem.

Data Preparation
Data preparation is one of the most important and essential steps for obtaining a promising result when preparing a learning model. All models are applied on data of the S&P CNX Nifty index option which are collected from the website of National Stock Exchange of India (www.nseindia.co.in) for a period of two year from 23 January 2012 to 8 January 2014. The forecasting error may be high in thinly traded options as compared to highly traded options therefore only those options data are considered that had daily volume exceeded more than 100 lots per day. After doing this filtration there are 22,840 call option pricing data points. Which is further divided into five groups, each group is named according to the different market conditions. Black Scholes model requires five input parameters i.e. risk-free interest rate, spot price, strike price, time to maturity and volatility. Out of five input, risk-free interest rate and volatility are not included in this learning model [2]. The inputs to learning system are time-to-maturity and moneyness ratio. Time-to-maturity is the difference of expiration and current trading dates and expressed in years. This experiment focuses on three-month period for the time-to-maturity .To narrow the whole data set for a better training purpose, term moneyness (ratio of stock price and strike price[26]) is used for grouping the data. These groups are: deep in-the-money market condition, in-the-money market condition, at-the-money market condition, deep out-of-the-money market condition and out-of-the-money market condition. According to the option pricing theory call option is said to out-of-the-money market condition and deep out-of-the-money market condition when S<K, secondly call option is at-the-money market condition when S ≈ K and at last deep in-the-money market condition and in-the-money market condition when S > K. Numeric value of moneyness is used to indicate the presence of each market conditions on data set. If the value of moneyness is less than 0.91 then market condition is called deep in-the-money market conditions. If the value of moneyness is lie between 0.91 to 0.98 then market condition is called in-the-money market conditions. If the value of moneyness is lie between 0.98 to 1.06 then market condition is called at-the-money market conditions and if the value of moneyness is lie between 1.06 to 1.18 then market condition is called in-the-money market conditions. At last if the value of moneyness is greater than 1.18 then market condition is called deep-in-the-money market conditions.

Parameter Setting For Multiple Kernel Support Vector Regressions and Methodology
In this paper regression function for multiple kernel is obtained by minimizing the empirical risk on the given option pricing training data. For this purpose ε-insensitive loss function proposed by Vapnik [27] is used for minimization of empirical risk. Support Vector Regressions performs well if hyper-parameters ε, c and the kernel width parameter are tuned properly. Where Parameter ε controls the width of the ε-insensitive and parameter c controls the tradeoff between the model complexity (flatness) and the degree to which deviations larger than ε are tolerated in optimization formulation [28]. The type of kernel function and its parameters is based on domain of applicable data sets and may reflect distribution of input data of the training data set [28].These parameters are often selected by cross validation [10] or trail method [8].These methods are very time consuming therefore analytical method is applied on selection of parameter c directly from the option pricing training data as proposed by V. cherkassky. \( \frac{\mu}{\sigma} \) is the mean of the closing values of Nifty Index option of training data, and \( \sigma_i \) is the deviation on those closing values of training data then the value of c is determined by

\[
c = \max \left( \frac{\mu + 3\sigma_i}{\| y \|}, \frac{\mu - 3\sigma_i}{\| y \|} \right)
\]

Radial basis kernel function is chosen for forecasting of option prices. The width of kernel parameter should reflect the distribution of input values of the option pricing data [25], therefore we take the set of 20 basis kernel function and each element of this set represents the width of radial basis kernel function for multiple kernel support vector regression. i.e.

\[
W = \{ .001, \ldots, .9, \ldots, 1.1 \}
\]

Further for training and testing purpose each data is divided in ratio of 70% and 30%. This implies that the S&P CNX Nifty index option is tested and trained under five situations i.e. deep-in-the-money, in-the-money, out-of-the-money, deep-out-of-the-money and at-the-money market conditions. Further to make the model more significant, scaling is performed on the whole data set. In support vector regression kernel values depend on the dot products of input output feature vectors therefore always scaling each attribute in the range-1 to +1 or 0 to-1 is beneficial [29]. Steps involved in option pricing forecasting are as follows:-

1. Data filtration: Stock having trading volume less than 100 is not considered.
2. Preparing group of all data: Divide the whole data by k mean clustering into five market condition based on moneyness.
3. Preparing proper format: Transform option pricing data of each market condition to the format of LIBSVM package.
4. Normalization: Apply min-max normalization technique on each market condition data.
5. Partition of data: Data set divided in to 70% and 30% ratio for the purpose of training and testing.
6. Selection of model parameters: Set W is width of Radial basis kernel, \( \epsilon = 0.1 \) and calculate the value of c under each market conditions using equation (10). Value of c for deep out-of-money market condition is 0.4126, for out-of-money market condition is 0.6557, for at-the-money market condition is 0.9281, for deep in-the-money market condition is 0.9743 and for in-the-money market condition is 1.1197.
7. Option pricing forecasting: Option values are forecasted using simple MKL, MKL with mirror descent method and with SMO method.
Performance Criteria
To estimate the pricing performance of MKL methods with different algorithm under different market condition Mean Square Error (MSE) is considered. N represents total number of option pricing data, \( y_i \) is empirically evaluated option prices and \( y_i^2 \) is actual market option price. MSE is defined as
\[
MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i^1 - y_i^2)^2
\]

Experimental Results
The aim of experiment is to compare the performance of SMO-MKL method for \( L^p \)-norm with that of reduced gradient, mirror descent method for \( L_1 \)-norm. Experiments are carried out on Intel(R) Xeon(R) CPU X5650@ 2.67 GHz processor with 12 GB RAM using simplemkl toolbox. The SMO-MKL is available on [30] in LIBSVM interface. The performance under different market conditions with different norm value is shown in Table 1.1, 1.2, 2.1, 2.2, 3.1, 3.2, 4.1, 4.2, 5.1, and 5.2.

**Table 1.1:** Mean square error of \( L_1 \)-norm MKLSVR models under “deep out of the money market condition" in 10\(^{-3}\)

<table>
<thead>
<tr>
<th>Models</th>
<th>Data-Points</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple MKL</td>
<td>4724</td>
<td>22.40</td>
</tr>
<tr>
<td>Mirror descent</td>
<td>4724</td>
<td>22.60</td>
</tr>
</tbody>
</table>

**Table 1.2:** Mean square error of \( L_p \)-norm MKLSVR Model with SMO under “deep out of the money market condition" in 10\(^{-3}\)

<table>
<thead>
<tr>
<th>Value of p</th>
<th>Data-Points</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.01</td>
<td>4724</td>
<td>22.172.00</td>
</tr>
<tr>
<td>1.05</td>
<td>4724</td>
<td>22.172.33</td>
</tr>
<tr>
<td>1.10</td>
<td>4724</td>
<td>22.142.66</td>
</tr>
<tr>
<td>1.15</td>
<td>4724</td>
<td>22.103</td>
</tr>
<tr>
<td>1.20</td>
<td>4724</td>
<td>22.054</td>
</tr>
<tr>
<td>1.33</td>
<td>4724</td>
<td>21.945</td>
</tr>
<tr>
<td>1.66</td>
<td>4724</td>
<td>21.67</td>
</tr>
</tbody>
</table>

**Table 2.1:** Mean square error of \( L_1 \)-norm MKLSVR models under “out of the money market condition" in 10\(^{-3}\)

<table>
<thead>
<tr>
<th>Models</th>
<th>Data-Points</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple MKL</td>
<td>7025</td>
<td>38.40</td>
</tr>
<tr>
<td>Mirror descent</td>
<td>7025</td>
<td>50.58</td>
</tr>
</tbody>
</table>

**Table 2.2:** Mean square error of \( L_p \)-norm MKLSVR Model with SMO under “out of the money market condition" in 10\(^{-3}\)

<table>
<thead>
<tr>
<th>Value of p</th>
<th>Data-Points</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.01</td>
<td>7025</td>
<td>29.732.00</td>
</tr>
<tr>
<td>1.05</td>
<td>7025</td>
<td>29.642.33</td>
</tr>
<tr>
<td>1.10</td>
<td>7025</td>
<td>29.262.66</td>
</tr>
<tr>
<td>1.15</td>
<td>7025</td>
<td>28.863</td>
</tr>
<tr>
<td>1.20</td>
<td>7025</td>
<td>28.394</td>
</tr>
<tr>
<td>1.33</td>
<td>7025</td>
<td>27.415</td>
</tr>
<tr>
<td>1.66</td>
<td>7025</td>
<td>25.03</td>
</tr>
</tbody>
</table>

**Table 3.1:** Mean square error of \( L_1 \)-norm MKLSVR models under “at the money market condition” in 10\(^{-3}\)

<table>
<thead>
<tr>
<th>Models</th>
<th>Data-Points</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple MKL</td>
<td>6110</td>
<td>25.90</td>
</tr>
<tr>
<td>Mirror descent</td>
<td>6110</td>
<td>40.00</td>
</tr>
</tbody>
</table>

**Table 3.2:** Mean square error of \( L_1 \)-norm MKLSVR Model with SMO under “at the money market condition” in 10\(^{-3}\)

<table>
<thead>
<tr>
<th>Value of p</th>
<th>Data-Points</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.01</td>
<td>6110</td>
<td>12.252.00</td>
</tr>
<tr>
<td>1.05</td>
<td>6110</td>
<td>12.182.33</td>
</tr>
<tr>
<td>1.10</td>
<td>6110</td>
<td>11.982.66</td>
</tr>
<tr>
<td>1.15</td>
<td>6110</td>
<td>11.783</td>
</tr>
<tr>
<td>1.20</td>
<td>6110</td>
<td>11.534</td>
</tr>
<tr>
<td>1.33</td>
<td>6110</td>
<td>11.015</td>
</tr>
<tr>
<td>1.66</td>
<td>6110</td>
<td>9.55</td>
</tr>
</tbody>
</table>

**Table 4.1:** Mean square error of \( L_1 \)-norm MKLSVR models under “in the money market condition” in 10\(^{-3}\)

<table>
<thead>
<tr>
<th>Models</th>
<th>Data-Points</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple MKL</td>
<td>3589</td>
<td>18.60</td>
</tr>
<tr>
<td>Mirror descent</td>
<td>3589</td>
<td>44.38</td>
</tr>
</tbody>
</table>

**Table 4.2:** Mean square error of \( L_1 \)-norm MKLSVR Model with SMO under “in the money market condition” in 10\(^{-3}\)

<table>
<thead>
<tr>
<th>Value of p</th>
<th>Data-Points</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.01</td>
<td>3589</td>
<td>6.792.00</td>
</tr>
<tr>
<td>1.05</td>
<td>3589</td>
<td>6.812.33</td>
</tr>
<tr>
<td>1.10</td>
<td>3589</td>
<td>6.662.66</td>
</tr>
<tr>
<td>1.15</td>
<td>3589</td>
<td>6.463</td>
</tr>
<tr>
<td>1.20</td>
<td>3589</td>
<td>6.354</td>
</tr>
<tr>
<td>1.33</td>
<td>3589</td>
<td>6.045</td>
</tr>
<tr>
<td>1.66</td>
<td>3589</td>
<td>5.33</td>
</tr>
</tbody>
</table>

**Table 5.1:** Mean square error of \( L_1 \)-norm MKLSVR models under “deep in the money market condition” in 10\(^{-3}\)

<table>
<thead>
<tr>
<th>Models</th>
<th>Data-Points</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple MKL</td>
<td>1392</td>
<td>25.20</td>
</tr>
<tr>
<td>Mirror descent</td>
<td>1392</td>
<td>45.76</td>
</tr>
</tbody>
</table>

**Table 5.2:** Mean square error of \( L_1 \)-norm MKLSVR Model with SMO under “deep in the money market condition” in 10\(^{-3}\)

<table>
<thead>
<tr>
<th>Value of p</th>
<th>Data-Points</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.01</td>
<td>1392</td>
<td>19.322.00</td>
</tr>
<tr>
<td>1.05</td>
<td>1392</td>
<td>19.222.33</td>
</tr>
<tr>
<td>1.10</td>
<td>1392</td>
<td>18.582.66</td>
</tr>
<tr>
<td>1.15</td>
<td>1392</td>
<td>18.003</td>
</tr>
<tr>
<td>1.20</td>
<td>1392</td>
<td>17.484</td>
</tr>
<tr>
<td>1.33</td>
<td>1392</td>
<td>15.705</td>
</tr>
<tr>
<td>1.66</td>
<td>1392</td>
<td>12.87</td>
</tr>
</tbody>
</table>

© Research India Publications.  http://www.ripublication.com
Conclusion
The present studies on S&P CNX Nifty index option shows that the Lp-norm MKL SVR clearly shrink the forecast error. It successfully captures the nonlinear behavior of option prices. In addition, the mirror descent approach is slightly better than the simple MKL for L1-norm. Hence from all the table it is clear that the lowest error is occur under “in-the-market condition”, so it is beneficial to invest money under this condition. we conclude that for option pricing framework, the Lp-norm SMOMKL have higher forecasting accuracy than L1-norm MKL, and gave suitable alternatives to forecast option prices in the financial markets.

References


