



Multiple kernel support vector regression for pricing nifty option

Neetu Verma *, Sujoy Das, Namita Srivastava

Department of Mathematics and Computer Applications Maulana Azad National Institute of Technology, INDIA

**Corresponding author E-mail: neetukoshta@gmail.com*

Copyright © 2015 Neetu Verma et al. This is an open access article distributed under the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

The goal of present experiments is to investigate the use of multiple kernel learning as a tool for pricing options in the context of Indian stock market for Nifty index options. In this paper, fair price of an option is predicted by Multiple Kernel Support Vector Regression (MKLSVR) using linear combinations of kernels and Single Kernel Support Vector Regression (SKSVR). Prices of option highly depend on different money market conditions like deep-in-the-money, in-the-money, at-the-money, out-of-the-money and deep-out-of-the-money condition. The experimental study attempts to identify the forecasting errors with the help of mean square error; root mean square error, and normalized root mean square error between the market option prices and the calculated option prices by model for all market conditions. The results reflect that multiple kernel support vector regression performed fairly well in comparison to support vector regression with single kernel.

Keywords: Multiple Kernel Learning; Moneyiness; Option Pricing; Support Vector Regression.

1. Introduction

The behavior of financial market is uncertain, in terms of changes in many financial terms such as interest and exchange rates, index and asset prices etc. These changes make the financial market too risky. To minimize these risks, some financial instruments have been added in the financial market, called financial derivatives, whose value derives from the value of the underlying asset. An option is a type of the financial derivatives that gives the holder, the right to buy a specified quantity of the underlying asset at the fixed price called the strike price on or before the expiration date. The seller, however, has the obligation to sell this asset if the buyer of the call option decides to exercise his option to buy.

There are lots of methods to find option values such as analytical, numerical and machine learning methods. All the analytical and numerical methods are highly complex and also need rigorous mathematical computations [1]. Therefore, from mid-1990s machine learning techniques such as support vector regression (SVR) become a broad area for researchers in the field of finance. The main advantage of this method is that once the model has been trained with all optimal parameters, then forecasting of testing data are quite simple. Optimal Parameter selection of SVR is crucial as it reflects forecasting performance. Many researches have shown that using multiple kernels instead of a single may enhance the forecasting performance of regression function by automatically tuned kernel parameter and also improve predicting performance.

2. Related work

In recent years, machine learning methods have become popular for predicating. The reason being that all these methods are capable of predicting risky financial market as they can be used for nonlinear function approximation without any assumptions on the option pricing data [2-7]. Support Vector Machine (SVM) was proposed by Vapnik [8] based on the statistical learning theory. SVM is widely used in forecasting purposes because SVM is formulated as a quadratic programming problem, therefore, it gives good generalization performance and is not trapped in local minima; it always provides a global optimum solution [9]. It can be used for learning a variety of representations, such as radial basis functions, splines, polynomial estimators, and so on [10]. SVM has also been successfully applied for function

approximation, regression estimation, signal processing and time series forecasting [11-15]. In SVM, we need to select a proper kernel function and its parameter [9]. These parameters are chosen by either trial-and-error or cross validation over a range of their parameters. However, it is time-consuming too [9], [16]. To overcome this problem, researchers have investigated family of kernels instead of selecting a specific kernel. The approach is known as Multiple Kernel Learning. (MKL) [17]. Kernels can be combined in an umber of ways such as linear sum, direct sum, product, etc. [18]. MKL can be formulated as semi definite programming [19, 20] or quadratic programming [21], [22]. In this paper, MKL as proposed by Rakomatory [17] is used as a forecasting method for option pricing. It is basically a convex combination of basis kernels with optimal weights that are iteratively found by reduced gradient descent method. Recently, MKL has been successfully applied for stock market forecasting [16] and economic forecasting [23]. Nowadays, study fair option price is predicted using MKL. In this paper, forecasting methodology is discussed in Section 3, experimental setup in Section 4, results in Section 5 and conclusion in Section 6.

3. Forecasting methodology

The relationship between input and output in case of the option price is highly nonlinear. Therefore, forecasting an option becomes a highly challenging task. Kernel functions are most sought-after tools for analyzing and forecasting option price. Constructing a regression function using support vector regression with single and multiple kernel is the active research domain nowadays. Both models are capable of learning nonlinear relationships.

3.1. Support vector regression

Theory of Support Vector Regression (SVR) was proposed by Vapnik [8] in 1995. This regression model initially learned from the training data and used to predict the target values of testing input data. Let $\{x_i\}_{i=1}^N$ where $x_i \in \mathbb{R}^N$ is being the input data and $y_i \in \mathbb{R}$, be the output data. Consider a function that maps input data to higher-dimensional feature space using kernel function to make it a linear. It requires the calculation of the inner product of vectors in the input-space, and therefore, it needs only values of x_i of the training data [8, 10]. The regression model performs linear regression in the high-dimension feature space.

Regression function $f(x, w)$ defined as

$$f(x, w) = \langle w \cdot \varphi(x) \rangle + b \quad w \in \mathbb{R}_n, b \in \mathbb{R} \tag{1}$$

With at most ϵ -deviation from the target value y_i and the problem can be formulated as a convex optimization problem

$$\min_w \frac{1}{2} \|w\|^2 + c \sum_{i=1}^N (\xi_i + \xi'_i)$$

Such that

$$\begin{aligned} y_i - f(x, w) &\leq \epsilon + \xi_i \\ f(x, w) - y_i &\leq \epsilon + \xi'_i \\ \xi_i, \xi'_i &\geq 0 \end{aligned} \tag{2}$$

Above regression function uses ϵ -insensitive loss function [8, 10] given by equation (3)

$$L_\epsilon(y, f(x, w)) = \begin{cases} 0 & \text{if } |y - f(x, w)| \leq \epsilon \\ |y - f(x, w)| - \epsilon & \text{otherwise} \end{cases} \tag{3}$$

For regression function (2), the model complexity is usually controlled by ϵ and c using training data. Whereas parameter ϵ controls the distinction between ‘‘small’’ and ‘‘large’’ discrepancies and non-negative slack variables ξ_i, ξ'_i measure the deviation of all training data outside the tube. Regularization parameter $c > 0$ controls trade-off between the margin size and the amount up to which deviations larger than loss function (3) can be tolerated to avoid over-fitting. Those parameter selection strategies usually depend on the sparseness of training data [10]. The dual of the equation (2) with Lagrange multipliers given as follows:

$$\max_{\lambda_i, \lambda'_i} L(\lambda) = \sum_{i=1}^N y_i (\lambda_i - \lambda_i^*) - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\lambda_i - \lambda_i^*) (\lambda_j - \lambda_j^*) k(x_i, x_j) - \epsilon (\lambda_i + \lambda_i^*)$$

Such that $0 \leq \lambda_i, \lambda_i^* \leq c$

$$\sum_{i=1}^N (\lambda_i - \lambda_i^*) = 0 \tag{4}$$

$$k(x_i, x) = \phi(x_i) \cdot \phi(x) = \exp(-\gamma \|x_i - x\|^2) \tag{5}$$

Where kernel function defined by equation (5) is a radial basis function (RBF) and γ is the width parameter of the RBF kernel. It is also known as kernel complexity parameter, for regression problems model flexibility can be controlled by this parameter [10].

The solution of equation (4) gives the values of Lagrange's multipliers λ_i, λ_i^* then optimal regression function becomes

$$\begin{aligned} w &= \sum_{i=1}^n (\lambda_i - \lambda_i^*) \phi(x_i) \\ f(x) &= w \cdot \phi(x) + b \\ f(x) &= \sum_{i=1}^n (\lambda_i - \lambda_i^*) k(x_i, x) + b \end{aligned} \tag{6}$$

3.2. Multiple kernel support vector regression

In option market data is non-linearly distributed, so multiple kernels may be applied to cope with this type of distribution [16] a simple direct weighted sum of kernels can be defined as

$$\hat{K} = d_1 K_1 + d_2 K_2 + \dots + d_M K_M = \sum_{j=1}^M d_j K_j$$

$$K_j = [k_j(x_i, x)] = [\phi_j(x_i)' \phi_j(x)]$$

$$d_j \geq 0, \sum_{j=1}^M d_j = 1 \tag{7}$$

Where M is the total number of positive definite kernels on the same input space. \hat{K} is a non-negative linear combination of valid kernel which is a positive semi-definite matrix [19]. To avoid over fitting problem we have to control the size of the search space by doing sums of weights equal to 1 [16]. In case of MKLSVR equation (4) is reformulated in equation (8)

$$\min_{w, \xi, \xi'} \frac{1}{2} \sum_{i=1}^n \frac{1}{d_j} \|w\|^2 + c \sum_{i=1}^n (\xi_i + \xi'_i)$$

Such that

$$\begin{aligned} y_i - \sum_{j=1}^M d_j f_j(x_i, w) &\leq \varepsilon + \xi_i \\ \sum_{j=1}^M d_j f_j(x_i, w) - y_i &\leq \varepsilon + \xi'_i \\ \xi_i, \xi'_i &\geq 0 \\ d_j &\geq 0 \text{ where } j = 1, 2, \dots, M \\ \sum_{j=1}^M d_j &= 1 \end{aligned} \tag{8}$$

Again, taking dual of above optimization problems, which are respectively

$$\max_{\lambda, \lambda^*} L(\lambda) = \sum_{i=1}^n y_i (\lambda_i - \lambda_i^*) - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^M (\lambda_i - \lambda_i^*) (\lambda_i - \lambda_i^*) \sum_{j=1}^M d_j K_j(x_i, x_i) - \varepsilon (\lambda_i + \lambda_i^*)$$

Such that $0 \leq \lambda_i, \lambda_i^* \leq c$

$$\begin{aligned} \sum_{i=1}^N (\lambda_i - \lambda_i^*) &= 0 \\ d_j &\geq 0 \text{ where } j = 1, 2, \dots, M \\ \sum_{j=1}^M (d_j) &= 1 \end{aligned} \tag{9}$$

Since $L(\lambda)$ is convex and differentiable, reduced gradient method is applied for solving this problem. Once the gradient of $L(\lambda)$ is computed, optimum weight d can be obtained using a reduced gradient descent direction [17].

$$\frac{\partial L(\lambda)}{\partial d_j} = -\frac{1}{2} \sum_{i=1}^N (\lambda_i - \lambda_i^*) (\lambda_i - \lambda_i^*) K_j(x_i, x_i) \tag{10}$$

After solving gradient of $L(\lambda)$ one can find the descent direction D before updating the weight d of multiple kernels and it is given as

$$d \leftarrow d + \gamma 1D \tag{11}$$

Where $\gamma 1$ is step length for updating the weights. This updating procedure is repeated until the optimal value stops reducing with the optimal weight d and dual variables are λ_i . The predicted value of option can be obtained using this regression function

$$\begin{aligned} w &= \sum_{i=1}^N (\lambda_i - \lambda_i^*) \varphi(x_i) \\ f(x) &= \sum_{j=1}^M d_j w_j \cdot \varphi_j(x) + b \\ f(x) &= \sum_{i=1}^N (\lambda_i - \lambda_i^*) \sum_{j=1}^M d_j k_j(x, x) + b \end{aligned} \tag{12}$$

4. Experimental setup

All experiments are performed on S&P CNX Nifty index option pricing data collected from NSE website [25]. There are 22,840 data points from 23 January 2012 to 8 January 2014 [26]. The daily closing values are considered as the option value and to overcome the problems of over fitting data value having a volume less than 100 is discarded [7]. c and ϵ parameters control model complexity in SVR. To overcome the process of cross validation for value of c , analytical method is used to select this directly from the training data as proposed by V. cherkassky [24]. The main advantage of this selection is reducing time consumption and good generalization performance on large data set. Therregularization parameter is given as

$$c = \max(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|)$$

Where \bar{y} is the mean of the option values of training data, and σ_y is the deviation of option values of training data. Where parameter c can effectively handle outliers in the training data and parameter ϵ controls the width of the ϵ - insensitive zone that helps to control the number of support vectors in optimal regression function [10], [24]. The kernel selection and its parameters are usually based on distribution of input x_i values of the training data [10], [24]. Radial Basis Function (RBF) used in all experiments because the width of kernel parameter should reflect the distribution of input values of the option pricing training data. Therefore, in this experiment different width of kernel takes as input for obtaining the optimal value. For MKLSVR one group is used at a time, whereas in SKSVR width of kernel parameter is taken as the average weighted sum of kernel in a given group. The model selection is done by changing the width parameter. The value of c parameter value is selected as suggested by [24].The width parameter of kernel is divided into following ranges:

[.001, .002, .0030.01]

[.01, .02, .03 ... 0.1]

[.1, .2, .3 ... 1]

[1, 2, 3 ... 10]

Option value depends on five variables i.e. spot price, time to maturity, strike price, variance in the value of underlying asset (volatility) and risk free interest rate in case of non-dividend paying stock [1]. Hutchinson [2] used successfully

neural network in option pricing first time. Also, volatility and risk free interest rate not considered as input variable, therefore in this paper spot price, time to maturity, strike price used as input parameters Further moneyness i.e. The ratio of the spot price and strike price and time to maturity are taken as input and daily closing value of S&P CNX Nifty index option is taken as the output. The whole data set is divided in five different market conditions using k-mean clustering based on moneyness ($M=S/K$) for better estimation of option values. These conditions are defined by deep in-the-money ($1.18 \leq M$), in-the-money ($1.06 \leq M < 1.18$), at-the-money ($0.98 \leq M < 1.06$), out-of-the-money ($0.91 \leq M < 0.98$) and deep out-of-the-money ($M < 0.91$) market conditions. Data of every market condition are divided in 70% and 30% for training and testing purpose.

4.1. Performance metrics

The performance of the model is evaluated by the measures of the deviation between actual and theoretical option values. These measures are mean square error, root mean square error and normalized root mean square error. N represents the total number of option pricing data, y_i^1 is empirically evaluated option prices and y_i^2 is actual option prices from the market.

Mean Square Error (MSE)

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i^1 - y_i^2)^2$$

Root Mean Square Error (RMSE)

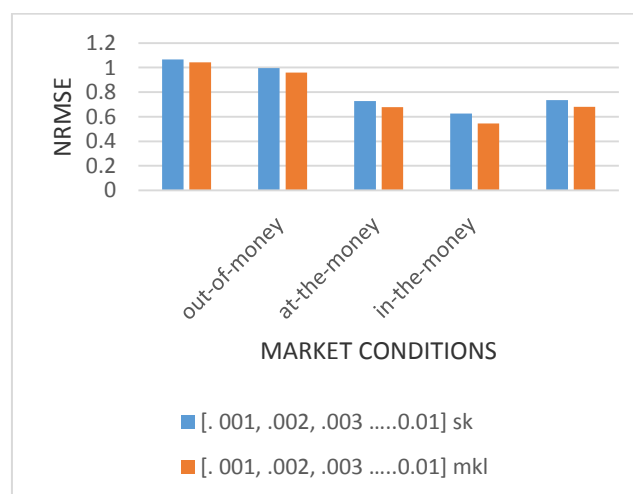
$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i^1 - y_i^2)^2}$$

Normalized Root Mean Square Error (NRMSE)

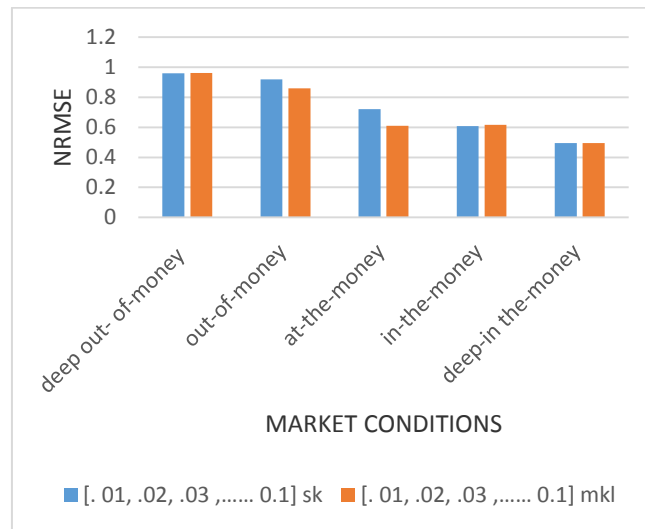
$$NRMSE = \frac{RMSE}{y_{\max}^2 - y_{\min}^2}$$

5. Main results and discussion

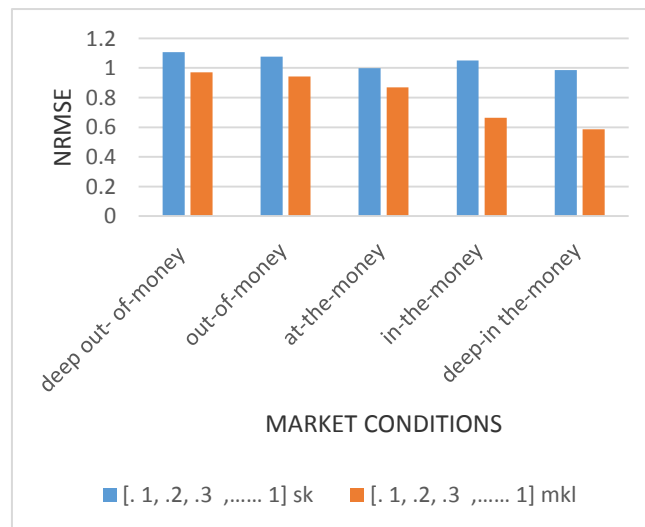
Pricing of option results is experimentally evaluated using multiple and single kernel SVR for different market conditions. Experiments are performed on MATLAB Intel(R) Xeon(R) CPU X5650@ 2.67 GHz 12 GB RAM along with simplemkl toolbox. Performance of all approach is measured using Mean Square Error (MSE), Root Mean Square Error (RMSE) and Normalized Root Mean Square Error (NRMSE). All experiments are performed using MKLSVR and SKSVR for option pricing with fixed regularization parameter c , ϵ and different width parameters. ϵ is taken as .01. c is taken as 1086.5 for in-the-money market condition, 1783.3 for deep in-the-money market condition, 532.2 for at-the-money market condition, 173.19 for out-of-the-money market condition and 34.4 for deep out-of-the-money market condition.



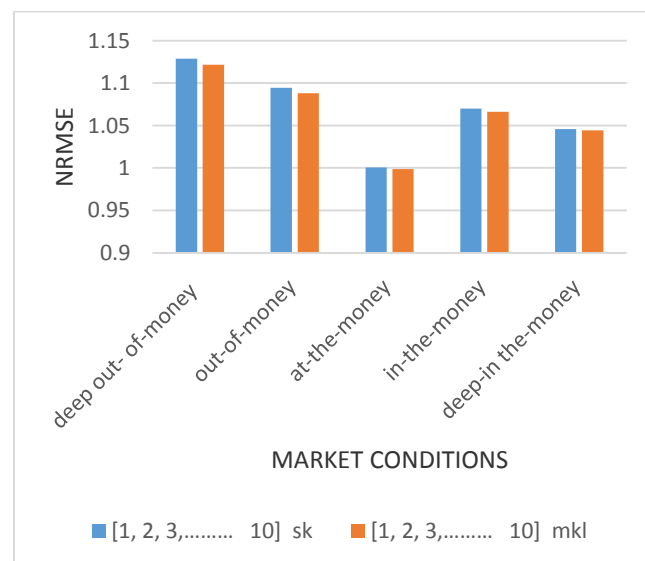
Graph 1: Performance of All Market Conditions in [.001, .002, .0030.01].



Graph 2: Performance of All Market Conditions in [0.01, 0.02, 0.03 ... 0.1].



Graph 3: Performance of All Market Conditions in [0.1, 0.2, 0.3 ... 1].



Graph 4: Performance of All Market Conditions in [1, 2, 3 ... 10].

It is clear from all the graphs that for different market conditions minimum error varies for different width of kernel. As the optimal width of kernel depends upon different market conditions, therefore width parameter of kernel is varied for obtaining optimal results under different market conditions. The width of kernel is kept in following ranges

[.001, .002, .003 ... 0.01], [.01, .02, .03, ... 0.1], [.1, .2, .3, ... 1] and [1, 2, 3, ... 10]. The performance on five different market conditions for different width of kernel parameter is shown in Table 1, 2, 3, 4 and 5.

Table 1: Performance of Both Models in “Deep Out-of-Money Market Condition”

Width of kernel parameter	Data points	MSE		RMSE		NRMSE	
		SK	MKL	SK	MKL	SK	MKL
[.001, .002, .003 ... 0.01]	4724	154.31	147.40	12.42	12.14	1.0661	1.0416
[.01, .02, .03, ... 0.1]	4724	124.67	125.57	11.17	11.21	.9578	.9610
[.1, .2, .3, ... 1]	4724	166.11	127.70	12.89	11.30	1.1070	.9694
[1, 2, 3, ... 10]	4724	172.55	170.44	13.14	13.06	1.1286	1.1216

Table 2: Performance of Both Models in “Out-of-Money Market Condition”

Width of kernel parameter	Data points	MSE		RMSE		NRMSE	
		SK	MKL	SK	MKL	SK	MKL
[.001, .002, .003 ... 0.01]	7025	2208.4	2046.1	46.99	45.23	.9972	.9592
[.01, .02, .03, ... 0.1]	7025	1867.6	1641.8	43.22	40.52	.9174	.8585
[.1, .2, .3, ... 1]	7025	2560.7	1964.9	50.60	44.33	1.0756	.9420
[1, 2, 3, ... 10]	7025	2650.1	2619.7	51.48	51.18	1.0944	1.0881

Table 3: Performance of Both Models in “at-the-Money Market Condition”

Width of kernel parameter	Data points	MSE		RMSE		NRMSE	
		SK	MKL	SK	MKL	SK	MKL
[.001, .002, .003 ... 0.01]	6110	5235.6	4538.5	72.36	67.37	.7278	.6780
[.01, .02, .03, ... 0.1]	6110	5123.1	3664.7	71.58	60.54	.7189	.6091
[.1, .2, .3, ... 1]	6110	9800.5	7487.7	98.99	86.53	.9948	.8697
[1, 2, 3, ... 10]	6110	9911.9	9879.1	99.56	99.39	1.0004	.9988

Table 4: Performance of Both Models in “in-the-Money Market Condition”

Width of kernel parameter	Data points	MSE		RMSE		NRMSE	
		SK	MKL	SK	MKL	SK	MKL
[.001, .002, .003 ... 0.01]	3589	7733.1	5899	87.94	76.81	.6257	.5463
[.01, .02, .03, ... 0.1]	3589	4886.6	4901.6	69.90	70.01	.4944	.4952
[.1, .2, .3, ... 1]	3589	21888	8689.0	147.94	93.22	1.05	.6637
[1, 2, 3, ... 10]	3589	22720	22588	150.73	150.29	1.066	1.069

Table 5: Performance of Both Models in “Deep in-the-Money Market Condition”

Width of kernel parameter	Data points	MSE		RMSE		NRMSE	
		SK	MKL	SK	MKL	SK	MKL
[.001, .002, .003 ... 0.01]	1392	20419	17625	142	132.75	.7352	.6804
[.01, .02, .03, ... 0.1]	1392	14513	14824	120.47	121.75	.6076	.6155
[.1, .2, .3, ... 1]	1392	36733	13661	191	116.88	.9858	.5867
[1, 2, 3, ... 10]	1392	41358	41234	203.37	203.06	1.0457	1.0441

6. Conclusion

In this study MKLSVR approach performed fairly well than SKSVR in pricing S&P CNX Nifty index call option. Fitness of both the approaches is good in case of deep-out-of-money market where as in case of out-of-money market it was more or less average. For in-the-money and at-the-money market condition model is workable. Both the models did not perform well in deep-in-the-money market condition.

References

- [1] J. C. Hull, *Options, futures, and other derivatives*, Prentice-Hall (1997).
- [2] Hutchinson, “A nonparametric approach to pricing and hedging derivative securities via learning networks”, *Journal of Finance*, Vol.49, No.3, (1998), pp.851-889.
- [3] Lajbeygier, Paul R., and Jerome T. Connor. "Improved option pricing using artificial neural networks and bootstrap methods." *International journal of neural systems*, Vol.8, No.4, 8.04 (1997), pp.457-471.
- [4] Yao, J., Li, Y., and Tan, C. L., “Option price forecasting using neural networks”, *Omega*, Vol.28, No.4, (2000), pp.455-466. [http://dx.doi.org/10.1016/S0305-0483\(99\)00066-3](http://dx.doi.org/10.1016/S0305-0483(99)00066-3).
- [5] Andreou, Panayiotis C., Chris Charalambous, and Spiros H. Martzoukos, “Pricing and trading European options by combining artificial neural networks and parametric models with implied parameters”, *European Journal of Operational Research*, Elsevier, Vol.185, No.3, (2008), pp.1415-1433. <http://dx.doi.org/10.1016/j.ejor.2005.03.081>.

- [6] Anant Saxena, "Valuation of S&P CNX Nifty Options: Comparison of Black-Scholes and Hybrid ANN Mode", *SAS Global Forum*, 162, (2008).
- [7] S. K. Mitra, "An Option Pricing Model That Combines Neural Network Approach and Black Scholes Formula", *Global Journals Inc. (USA)*, Vol.12, No.4, (2012), pp.6-16. (1998).
- [8] Vapnik, V. N., *the Nature of Statistical Learning Theory*, Springer, New York. John Wiley & Sons, (1995).
- [9] Abe, Shigeo. *Support vector machines for pattern classification*. Vol. 2. London: Springer, (2005).
- [10] Cherkassky, V., & Mulier, F., *Learning from data: Concepts, theory, and methods*, New York: Wiley, (2007). <http://dx.doi.org/10.1002/9780470140529>.
- [11] Vapnik, V., Golowich, S., & Smola, "Support vector method for function approximation, regression estimation and signal processing", *Advances in Neural Information Processing Systems*, Vol.9, (1997), pp.281-287.
- [12] Tay, F. E. H. and Cao, L., "Application of support vector machines in financial time-series forecasting", *Omega*, Vol.29, No.4, (2001), pp.309-317. [http://dx.doi.org/10.1016/S0305-0483\(01\)00026-3](http://dx.doi.org/10.1016/S0305-0483(01)00026-3).
- [13] Tay, F. E. H. and Cao, L., "Modified support vector machines in financial time series forecasting", *Neurocomputing*, Vol.48, No.1, (2002), pp.847-861. [http://dx.doi.org/10.1016/S0925-2312\(01\)00676-2](http://dx.doi.org/10.1016/S0925-2312(01)00676-2).
- [14] Kyoung-jae Kim, "Financial time series forecasting using support vector machines" *Neurocomputing*, Vol.55, (2003), pp.307 – 319. [http://dx.doi.org/10.1016/S0925-2312\(03\)00372-2](http://dx.doi.org/10.1016/S0925-2312(03)00372-2).
- [15] Lijuan Cao, "Support vector machines experts for time series forecasting", *Neurocomputing*, Vol.51, (2003), pp.321 – 339. [http://dx.doi.org/10.1016/S0925-2312\(02\)00577-5](http://dx.doi.org/10.1016/S0925-2312(02)00577-5).
- [16] 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, (2011), pp. 2177-2186. <http://dx.doi.org/10.1016/j.eswa.2010.08.004>.
- [17] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. "Simple multiple kernel learning" [J]. *J. Mach.Learn. Res.*, Vol.9, (2008), pp.2491-2521.
- [18] Gönen, Mehmet, and EthemAlpaydm. "Multiple kernel learning algorithms" *The Journal of Machine Learning Research*, Vol.12, (2011), pp.2211-2268.
- [19] Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. I. "Learning the kernel matrix with semidefinite programming". *Journal of Machine Learning Research*, Vol.5, (2004), pp.27-72.
- [20] Bach, F. R., Lanckriet, G. R. G., & Jordan, M. I. "Multiple kernel learning, conic duality, and the SMO algorithm". In: *Proceedings of the 21th international conference on machine learning*, (2004), pp. 6-13. <http://dx.doi.org/10.1145/1015330.1015424>.
- [21] Fung, G., Dundar, M., Bi, J., & Rao, B. "A fast iterative algorithm for fisher discriminant using heterogeneous kernels". In *Proceedings of the 21st international conference on machine learning*, (2004), p. 40. <http://dx.doi.org/10.1145/1015330.1015409>.
- [22] Kim, S., Magnani, A., & Boyd, S. "Optimal kernel selection in kernel fisher discriminant analysis". In *Proceedings of the 23rd international conference on machine learning*, (2006), pp. 465-472. <http://dx.doi.org/10.1145/1143844.1143903>.
- [23] Xiang-rong, Zhang, H. U. Long-ying, and Wang Zhi-sheng. "Multiple kernel support vector regression for economic forecasting". *Management Science and Engineering (ICMSE), International Conference on. IEEE*, (2010), pp. 129-134.
- [24] Cherkassky, Vladimir, and Yunqian Ma. 2004, "Practical selection of SVM parameters and noise estimation for SVM regression". *Neural networks*, Vol.17, No.1, pp.113-126. [http://dx.doi.org/10.1016/S0893-6080\(03\)00169-2](http://dx.doi.org/10.1016/S0893-6080(03)00169-2).
- [25] National stock exchange of India ltd. <http://www.nseindia.com/>. Accessed: January 2014.
- [26] Verma, Neetu, Namita Srivastava, and Sujoy Das. "Forecasting the Price of Call Option Using Support Vector Regression". (2014) *IOSR Journal of Mathematics*, Vol. 10, No.6, pp.38-43.