Journal of Computer Applications Research and Development (JCARD), ISSN 2248-9304(Print), ISSN 2248-9312 (Online), Volume 1, Number 1, April-June (2011)
pp. 19-26 © PRJ Publication
http://www.prjpublication.com/JCARD.asp

JCARD

© PRJ PUBLICATION

RIDGE REGRESSION MODEL FOR THE PREDICTION OF STOCK INDEX WHEN MULTICOLLINEARITY OCCURS

K. V. Sujatha [1] and S. Meenakshi Sundaram [2]
1. Research Scholar, Sathyabama University, Chennai 600 119
2. Department of Mathematics, Sathyabama University, Chennai 600 119
Email: sujathacenthil@gmail.com Email: sundarambhu@rediffmail.com

ABSTRACT

Stock prices are very dynamic and susceptible to quick changes because of the underlying nature of the financial domain. Difficulty in prediction of stock prices comes from the complexities associated with the market dynamics when parameters are constantly shifting and are not fully defined. Recently a lot of interesting work has been done in area of applying Machine Learning Algorithm for analyzing and predicting stock prices. In this paper an attempt is made to predict the daily closing prices of BSE sensex data using the daily opening price, high price and low price. Due to the volatile nature of the closing prices, the basic assumption of normality for parametric modeling of the data is not met. There by nor parametric neural network models are proposed. In addition the prediction variables considered are multi collinear and hence classical Ridge Regression model is fitted. Computation nonparametric neural network model surpassed classical statistical model in predicting the daily prices. Standard error measures are used to validate the prediction ability of the proposed models.

Keywords: Multilayer Perceptron, Normality, Multicollinearity, Ridge Regression, Mean Absolute Error.

I. INTRODUCTION

Interest in financial markets has increased in the last couple of decade, among participants, private-sector market operators, policymakers and academics. . It is often argued that financial market is very efficient. Therefore, it is difficult to make short term and long term forecasting efficiently. A large body of literature has been accumulated over many years concerning the validity of the efficient market hypothesis (EMH) with respect to stock markets. Predicting the stock market is very difficult since it depends on several known and unknown factors. The power of neural networks is its ability to model a nonlinear process without a priori knowledge about the nature of the process. In this paper we have made an attempt to explore the emerging field of artificial neural network to the complex task of modeling stock prices in the Indian context. We examined both Feed Forward Neural Network (FFN) using back propagation algorithm with early stopping and Radial Basis Neural Network (RBN).

The network inputs, architecture, training strategies were decided based on experimental results. The trained networks are then used for price prediction of stock. There have been a number of attempts to apply ANN to the task of modeling security prices ([2], [4], [6], [7], [8] and [10]). When it comes to performing a predictive analysis of the security prices, it is very difficult to built one general model that will fit every market and every security. Experience has shown that such models tend to be specific to markets and asset classes and a general model may not be applicable across markets and asset classes. One of the major problems faced in modeling financial markets is the fact that the information comes in from very large number of sources, and at least some of the price movements are direct results of the expectation by market participants of that very moment [11]. It has been observed that financial markets get affected by virtually anything that has a bearing on the economy.

In Indian context stock price and trend prediction using neural network was done [9]. In recent years, the genetic algorithm (GA) and the neural network system (NNs) have been widely used to solve financial decision-making problems due to their excellent performances in treating non-linear data with self-learning capability. ([3], [8])

The research work is organized as follows: Section 2, briefly discuss Multivariate normality, the fundamental concepts of ANN, Multicollinearity and Ridge Regression: Section 3, discuss checking multivariate normality , construction of ANNs for modeling BSE SENSEX data and fitting Ridge Regression and Section 4, is Conclusion.

II. METHODOLOGY

A. *Multivariate Normality*

Many Multivariate normality tests are available to test the Marginal normality in a multivariate data. The Shapiro-Wilk test is used if sample size is less than or equal to 5000. Mardia's skewness and kurtosis coefficients are computed and tests of significance are performed for these coefficients using asymptotic distributions. These tests are generally effective for testing multivariate normality. Henze-Zirkler test statistic is found with the associated p-value using the lognormal distribution.

Let $X' = (X_1, \ldots, X_n)$ be a p x n matrix of n observations on a p −dimensional vector with sample mean and covariance $\bar{X} = n^{-1}(X_1 + \ldots. + X_n)$ and $S = n^{-1}\bar{X}'\bar{X}$ where $\bar{X}' = (X_1 - X, \ldots, X_n - X)$. Create the n x n matrix:

$$D_\beta(P,Q) = \int'_{R^d} \left|\hat{P}(t) - \hat{Q}(t)\right|^2 \varphi_\beta(t) \qquad (1)$$

And define multivariate measures of skewness and kurtosis as:

$$b^*_{1p} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} d^3_{ij} \qquad (2)$$

and

$$b^*_{2p} = \frac{1}{n}\sum_{i=1}^{n} d^4_{ii} \qquad (3)$$

An omnibus test based on these measures is given by

$$M_p = \frac{n b^*_{1p}}{6} + \frac{n\left(b^*_{2p} - p(p+2)\right)^2}{8p(p+2)} \qquad (4)$$

$$\sim \chi^2\left(\frac{p(p+1)(p+2)}{6} + 1 \right) \qquad (5)$$

The Henze-Zirkler test is based on a nonnegative functional D(.,.) that measures the distance between two distribution functions and has the property that $D(N_d(0, I_d), Q) = 0$ if and only if $Q = N_d(0, I_d)$ where $N_d(\mu.\sum d)$ is a *d*-dimensional normal distribution.

The distance measure *D*(.,.) can be written as

$$D_\beta(P,Q) = \int_{R^d} |\hat{P}(t) - \hat{Q}(t)|^2 \varphi_\beta(t)dt \qquad (6)$$

were $\widehat{P(t)}$ and $\widehat{Q(t)}$are the Fourier transforms of P and Q, and$\varphi_\beta(t)$ is a weight or a kernel function. The density of the normal distribution $N_d(0,\beta^2 I_d)$is used as $\varphi_\beta(t)$

$$\varphi_\beta(t) = (2\pi\beta^2)^{\frac{-d}{2}} \exp\left(\frac{-|t|^2}{2\beta^2}\right), t \in R^d \quad (7)$$

were $|t| = (t' t)^{0.5}$. The parameter depends on n as

$$\beta_d(n) = \frac{1}{\sqrt{2}}\left(\frac{2d+1}{4}\right)^{1/(d+4)} n^{1/(d+4)} \quad (8)$$

The test statistic computed is called $T_\beta($ and is approximately distributed as a log normal. The log normal distribution is used to compute the null hypothesis probability.

B. Multilayer Perceptron

The multilayer perceptron (MLP) or radial basis function (RBF) network is a function of predictors (also called inputs or independent variables) that minimize the prediction error of target variables (also called outputs). The MLP procedure fits a particular kind of neural network called a multilayer perceptron. The multilayer perceptron uses a feed forward architecture and can have multiple hidden layers. It is one of the most commonly used neural network architectures. MLP optionally rescales covariates or scale dependent variables before training the neural network. MLP optionally divides the dataset into training, testing, and holdout data. The neural network is trained using the training data. The training data or testing data, or both, can be used to track errors across steps and determine when to stop training. The holdout data is completely excluded from the training process and is used for independent assessment of the final network. Units in the hidden layers can use the hyperbolic or sigmoid activation functions. Units in the output layer can use the hyperbolic, sigmoid, identity, or soft max activation functions.

The neural network can be built using batch, online, or mini-batch training. Gradient descent and scaled conjugate gradient optimization algorithms are available. The MLP procedure uses random number generation during random assignment of partitions, random sub sampling for initialization of synaptic weights, random sub sampling for automatic architecture selection, and the simulated annealing algorithm used in weight initialization and automatic architecture selection. There are three rescaling options: standardization, normalization, and adjusted normalization. All rescaling is performed based on the training data, even if a testing or holdout sample is defined.

C. Radial Basis Function

Radial basis function (RBF) networks have a static Gaussian function as the nonlinearity for the hidden layer processing elements. The RBF procedure fits a radial basis function neural network, which is a feed forward, supervised learning network with an input layer, a hidden layer called the radial basis function layer, and an output layer. The hidden layer transforms the input vectors into radial basis functions. Like the MLP (multilayer perceptron) procedure, the RBF procedure performs prediction and classification. RBF optionally rescales covariates (predictors with scale measurement level) or scale dependent variables before training the neural network. There are three rescaling options: standardization, normalization, and adjusted normalization. RBF optionally divides the dataset into training, testing, and holdout data. The neural network is trained using the training data. The testing data can be used to determine the “best” number of hidden units for the network.

The holdout data is completely excluded from the training process and is used for independent assessment of the final network. Units in the hidden layer can use the normalized radial basis function or the ordinary radial basis function. Normalized radial basis function uses the soft max activation function so the activations of all hidden units are normalized to sum to 1. This is the default activation function for all units in the hidden layer. Ordinary radial basis function uses the exponential activation function so the activation of the hidden unit is a Gaussian "bump" as a function of the inputs. The best model is the one which has minimum error. The models which have the minimum sum of square error and relative error value for both MLP and RBF are found.

Rescaling of Variables:

$$\text{Standardization} = \frac{X - Mean}{s} \tag{9}$$

$$\text{Normalization} = \frac{X - Minimum}{Maximum - Minimum} \tag{10}$$

$$\text{Adjusted Normalization} = \left[2 * \frac{X - Minumum}{Maximum - Minimum} \right] - 1 \tag{11}$$

For normalizing dependent variable normally and adjusted normally a correction factor ε is applied to the rescaling formula to ensure that all the rescaled dependent variable values will be within the range of the activation function. Error Functions that are used are sum of square error and relative error which are measured for training, testing and hold out.

$$\text{Sum of Square Error, } E_T(\omega) = \sum_{m=1}^{M} E_m(\omega) \tag{12}$$

$$\text{Relative Error} = \frac{\sum_{m=1}^{M}\left(Y_r^{(m)} - \hat{Y}_r^{(m)}\right)^2}{\sum_{m=1}^{M}\left(Y_r^{(m)} - \bar{Y}_r\right)^2} \tag{13}$$

$$\bar{Y}_r = \text{The mean of } Y_n^{(m)}$$

D. ***Ridge Regression***

Multiple Linear Regressions are very sensitive to predictors being in a configuration of near-collinearity: when this happens, the model parameters become unstable (large variances) and can therefore no longer be interpreted. From a mathematical standpoint, near-collinearity makes the *X'X* matrix ill-conditioned. Exact collinearity occurs when at least one of the predictors is a linear combination of other predictors. *X* is not a full rank matrix anymore, the determinant of *X* is exactly 0, and inverting *X'X* is not just difficult, it is downright impossible because the inverse matrix simply does not exist. Ridge Regression is a variant of ordinary Multiple Linear Regression whose goal is to circumvent the problem of predictor's collinearity. It gives-up the Least Squares (LS) as a method for estimating the parameters of the model, and focuses instead of the *X'X* matrix. This matrix will be artificially modified so as to make its determinant appreciably different from 0.

By doing so, it makes the new model parameters somewhat biased (whereas the parameters as calculated by the LS method are unbiased estimators of the true parameters). But the variances of these new parameters are smaller than that of the LS parameters and in fact, so much smaller than their Mean Square Errors (MSE) may also be smaller than that of the parameters of the LS model.

This is an illustration of the fact that a biased estimator may outperform an unbiased estimator provided its variance is small enough.

III. RESULTS AND FINDINGS

The most widely tracked and popular stock index in India is the BSE SENSEX. We used daily open, high, low and closing prices for the trading period from January 2007 till 26 October 2009 (689 days).

A. *Multivariate Normality*

The Marginal and the joint normality of the variables are checked using Shapiro- Willks test , Mardia's Skewness and Kurtosis and Henze-Zirkler tests. From the Table I and II it can be clearly seen that the condition of Normality is violated.

TABLE I Marginal Normality Tests

Variable	Test	Test Statistic	p-value
OPEN	Shapiro-Wilk	0.963	0.000
HIGH	Shapiro-Wilk	0.964	0.000
LOW	Shapiro-Wilk	0.962	0.000
CLOSING	Shapiro-Wilk	0.963	0.000

TABLE II Joint Normality

Test	Coefficients	Test Statistic	p-value
Mardia Skewness	23.025	2660.152	0.000
Mardia Kurtosis	70.070	87.273	0.000
Henze-Zirkler		15.034	0.000

As the normality assumptions are not met the prediction of the closing prices is made with non parametric neural network model.

B. *Neural Network Model*

The variables high, low and open are considered as inputs and the closing price as the target variable. The partitioning of the variables is done in such a way that, out of the entire lot 57.8% of the data are considered for training, 14.9 % for testing and 27.3% as hold out. This partition is decided based on lesser error value. In the remaining part of this section we discuss, with specific reference to the above data, the construction of ANNs that perform reasonable well. In Multilayer perceptron as the data set is very large online training was chosen. Since the training is chosen to be online the optimization algorithm was fixed to Gradient Descent. The dependent variable was rescaled standardized, normalized and adjusted normalized. The covariates were also rescaled in the above said option to standardized, normalized and adjusted normalized. The active function of the hidden layer was set to hyperbolic tangent and then to sigmoid function for each rescaling.

The error values (sum of square error and relative error) were measured for all the above models by checking the active function of the output layer to identity, hyperbolic tangent and sigmoid. The MLP model which gave the minimum error was the one in which the active function of the output layer and hidden layer are identity and hyperbolic tangent respectively, after normalizing dependent variable and standardizing the covariates. The units in the hidden layers are first set to normalized Radial Basis function.

The input variables are rescaled to normalized, adjusted normalized and standardized. With these rescaled variables, the output variables are transformed to normalized, standardized and adjusted normalized and fed into the network. The error values were very less for the model in which the dependent variable is normalized and the inputs were adjusted normalized with the hidden units having normalized Radial Basis function. The predicted value of this best model was calculated.

C. Ridge Regression

The collinearity analysis of the predictor variables revealed that they are highly correlated (Table III).

Table III Pearson Correlation Matrix

	OPEN	HIGH	LOW
OPEN	1.000		
HIGH	0.999	1.000	
LOW	0.998	0.998	1.000

Thus Ridge Regression Model is considered for this data. For different values of lambda the coefficients of the variables are found. The regression line with the minimum error value is considered as the best model in predicting the closing prices. The ridge regression with minimum error is when
$\lambda = 0.1$ and the regression equation is

$$\textbf{Closing} = 495.552 + 0.305\ \textbf{Open} + 0.327\ \textbf{High} + 0.334\ \textbf{Low}$$

The MAE and MAPE values for the proposed models are given in Table IV. From the table it is very obvious that Multilayer Perceptron Model predicts the stock prices more precisely than Radial Basis function and Ridge Regression

Table IV Error Values

	MAE	MAPE	RMSE
Multilayer Perceptron	76.66384325	0.005484821	107.2680832
Radial Basis Function	266.385149	0.019066797	337.0878103
Ridge Regression	152.9715	0.011344	198.9701

Table V Network Structure for Multilayer Perceptron Model

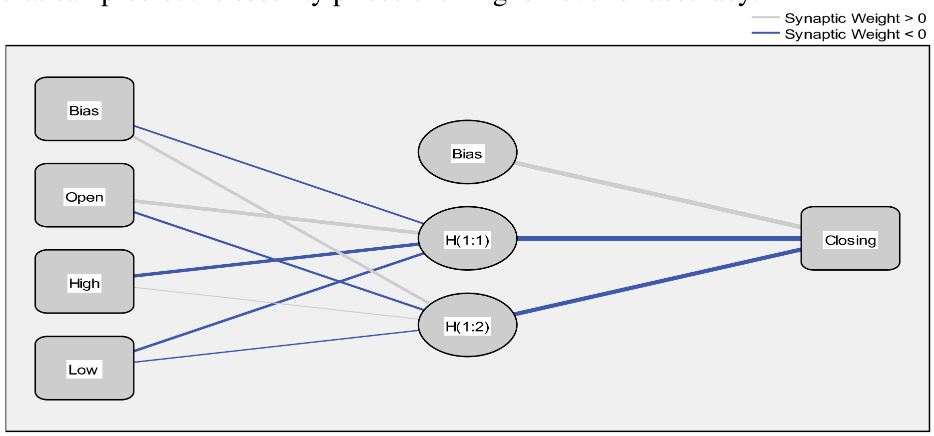
Network Information

Input Layer	Covariates	1	**Open**
		2	**High**
		3	**Low**
	Number of Units		**3**
	Rescaling Method for Covariates		**Standardized**
Hidden Layer(s)	Number of Hidden Layers		**1**
	Number of Units in Hidden Layer 1		**2**
	Activation Function		**Hyperbolic tangent**
Output Layer	Dependent Variables	1	**Closing**
	Number of Units		**1**
	Rescaling Method for Scale Dependents		**Normalized**
	Activation Function		**Identity**
	Error Function		**Sum of Squares**

For the best fitted Multilayer Perceptron Model the Network Structure and the parameter Estimate values are given in Table V and Table VI respectively. The architecture of the network is given in Figure 1.

IV. CONCLUSION

The Multilayer Perceptron Model with covariates standardized with hyperbolic tangent as the activation function of the hidden layer and the dependent variable normalized with identity as the activation function has minimum MAE, MAPE and RMSE value when compared with the Radial Basis Function and Ridge Regression. It has been observed that the error increases gradually during the validation period. Thus, an appropriate approach may be to retrain the network periodically. There is considerable scope to build on these results further and build ANN models that can predict the security prices with higher level of accuracy.



Hidden layer activation function: Hyperbolic tangent

Output layer activation function: Identity

Figure 1NETWORK ARCHITECTURE

Table VI Estimated Values of the Parameters

Parameter Estimates

Predictor		Predicted		
		Hidden Layer 1		Output Layer
		H(1:1)	H(1:2)	Closing
Input Layer	(Bias)	-.128	.282	
	Open	.367	-.209	
	High	-.332	.019	
	Low	-.264	-.041	
Hidden Layer 1	(Bias)			.513
	H(1:1)			-.717
	H(1:2)			-.393

REFERENCES

[1] Andersen, J.V., *Estimating the level of Cash Invested in Financial Markets*, Physica A, 344, 1 / 2, 168-173, 2004.

[2] Cao, Q., Leggio, K. B., and Schniederjans, M. J., *A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market*, Computers and Operations Research, 32, 2499- 2512, 2005.

[3] Huang Fu-yuan, *Forecasting Stock Price Using a Genetic Fuzzy Neural Network,* International Conference on Computer Science and Information Technology, 549 – 552, 2008.

[4] Jasic, T. and Wood, D., *The profitability of daily stock market indices trades based on neural network predictions: case study for the S&P 500, the DAX, the Topix and the FTSE in the period 1965-1999,* Applied Financial Economics, 14, 285-297 , 2004.

[5] Jovita Nenortaite, Rimvydas Simutis Kaunas, *Adapting Particle Swarm Optimization to Stock Markets,* Proceedings of the 2005 5th International Conference on Intelligent Systems Design and Applications 2005.

[6] Kaastra, L and Boyd, M.,*Designing a neural network for forecasting financial and economic time series* , Neurocomputing(10) 1995.

[7] K. V. Mardia, *Measures of multivariate Skewness and Kurtosis with applications*, Biometrika, Volume 57, 1970, pp. 519-530.

[8] K. V. Mardia, *Tests of Univariate and Multivariate Normality*, In Krishnaiah P.R. (ed.), Hand book of Statistics, Volume 1, Chapter 9, Amsterdam, North Holland, 1980.

[9] K.V. Sujatha and S. Meenakshi Sundaram, *A Combined PCA-MLP model for Predicting Stock Index*, Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India, Article No. 17, ACM, New York, NY, U.S.A., 2010.

[10] K. V. Sujatha and S. Meenakshi Sundaram, *Regression, Theil's and MLP Forecasting Models of Stock Index*, International Journal of Computer Engineering and Technology, Vol. 1, Issue 2, 2010, pp. 90-101.

[11] K. V. Sujatha and S. Meenakshi Sundaram, *A MLP, RBF Neural Network Model for Prediction in BSE SENSEX Data Set*, Proceedings of National Conference on Applied Mathematics, 2010.

[12] K. V. Sujatha and S. Meenakshi Sundaram, *Stock Index Prediction Using Regression and Neural Network Models under Non Normal Conditions*, Proceeding of International Conference in Recent trends in Robotics and Communication Technologies", 2010, pp. 58-62.

[13] K. V. Sujatha and S. Meenakshi Sundaram, *Non Parametric Models for Predicting Stock Prices,* Proceeding of National Conference on Recent Trends in Statistics and Computer Applications 2011.

[14] K. V. Sujatha and S. Meenakshi Sundaram, *Nonparametric Modeling of Stock Index,* National Journal of Advances in Computing and Management, Volume 1, Issue 2, 2011.

[15] K. V. Sujatha and S. Meenakshi Sundaram, *Neural Network And PLS Regression Models for Predicting Stock Prices,* CiiT International Journal of Artificial Intelligent Systems and Machine Learning June 2011.

[16] Lam, M., Neural *Network Techniques for Financial Performance Prediction*, Integrating Fundamental and Technical Analysis, Decision Support Systems, 37(4), 567- 581, 2004.

[17] Nygren, K., *Stock Prediction – A Neural Network Approach*, Master's Thesis, Royal Institute of Technology, KTH, Sweden, 2004.

[18] Pritam R. Charkha, *Stock Price Prediction and Trend Prediction using Neural Networks*, First International Conference on Emerging Trends in Engineering and Technology, 592- 594, 2008.

[19] Rajesh M. V., Archana R, A Unnikrishnan, R Gopikakumari, Jeevamma Jacob, *Evaluation of the ANN Based Nonlinear System Models in the MSE and CRLB Senses,* World Academy of Science, Engineering and Technology, 48, 2008

[20] Soros, G.,*The Alchemy of Finance: Reading the Mind of the Market*, 1987, John Wiley & Sons Inc., New York.

[21] Zhang, G., Patuwo, B. E., and Hu, M. Y., *Forecasting with Artificial Neural Networks: The State of the Art*, International Journal of Forecasting,14, 35-62, 1998.