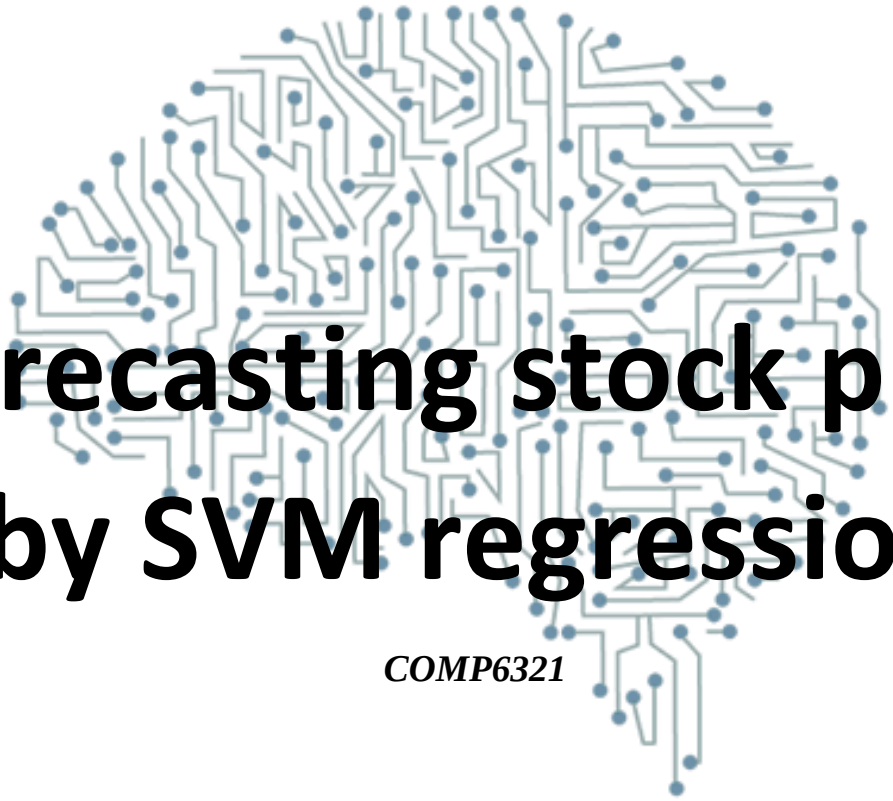


Forecasting stock price by SVM regression

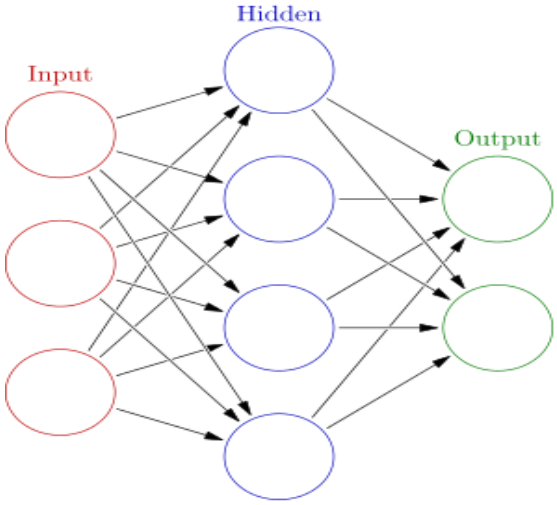
COMP6321

Bharath Parthasarathy (40036730)

1. INTRODUCTION

Stock price forecasting is very important in the investor point of view, it gives the investor an idea of what the day's closing rates. Thus giving them the opportunity to make decisions on their investment. Time series forecasting is one of the most challenging application, like the forecasting of the stock prices, due to the vast number of uncertainties that affect the outcome.

Since the data to be used for predicting the outcome in such problems is so random, using linear models in such instances is not feasible. Neural Networks have been used in a wide range of prediction algorithms and application, even in the case of stock market forecasting, this project shows two such instances or feature set, one feature set is to use the open price, high and the lows, and the other is to just use the closing prices of recent history in prediction.

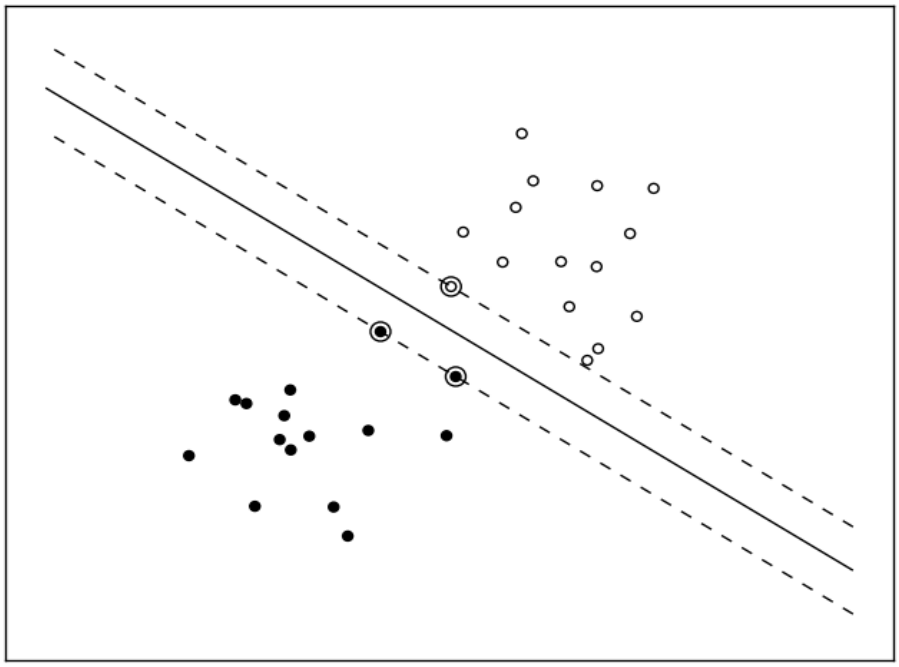


So what is Neural Networks? Neural networks are layers of organized and interconnected nodes that contain an activation function. In consists of input layer, hidden layers (one or more) and an output layer. All the calculations that go into generation of the output is carried out by the hidden layers, through a system of weighted connections. These weighted connections between the nodes of the network are trained through different processes or algorithms, to predict the desired output through backpropagation in a number of iterations called 'epochs'

2. SUPPORT VECTOR MACHINES

Support Vector Machines (SVMs) is a novel neural network model, which tries to implement structural risk minimization. Unlike the traditional neural networks or machine learning algorithms in general, that try to reduce the training error, SVMs try to reduce the upper bound of the generalization error.

SVMs construct hyperplanes or a set of hyperplanes in a high-dimensional space, which can be used for classification and regression. The original problem might be in a finite dimensional space, but most of the times it so happens that the data in the finite dimensional space might not be linearly separable, in that original space. For this reason SVMs map the finite-dimensional space into a much higher-dimensional space, making it easier to separate even the original non-linearly separable data.



To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products of pairs input data vectors may be

computed easily in terms of the variables in the original space, by defining them in terms of a kernel function *k(x, y)* selected to suit the problem.

Let us consider the set of data points $G = \{(x_i , d_i)\}_i^n$,

Where x_i is the input vector, d_i is the desired value and n is the total number of data vectors,

SVMs approximate the function using the following:

$$y = f(x) = w\phi(x) + b$$

Where, φ(x) is the high dimensional feature space which is non-linearly mapped from the input space x, and w and b are estimated using,

$$R_{SVMs}(C) = C\frac{1}{n}\sum_{i=1}^{n} L_\varepsilon(d_i, y_i) + \frac{1}{2}\|w\|^2,$$

$$L_\varepsilon(d\ ,y\ ) = \begin{cases} |d-y| - \varepsilon & |d-y| \geq \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

Where, $C(1/n)\sum_{i=1}^{n} L_\varepsilon(d_i, y_i)$ is the empirical error measured by a Ɛ- sensitive loss function L, and the second term $\frac{1}{2}\|w\|^2$ is the regularization term. C is the regularized constant, which is the trade-off between the empirical risk and the regularization term.

k(x, y) is the kernel function, and the value of the kernel is the product of two vectors x_i and x_j in the feature space φ(x_i) and φ(x_j) , that is,

$$K(x_i, x_j) = \phi(x_i) * \phi(x_j)$$

The kernel function that we would be using for the stock market prediction is the Radial basis function (RBF)

$$K(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2), \gamma > 0$$

where, γ ,r and d are kernel parameters.

3. METHOD

In this implementation for the forecasting of the stock prices using SVM, we would be using the following steps:

1. We would be using a single data set, but with two different feature extractions, thus we need to transform this data into suitable SVM format.
2. Implement the SVM regression algorithm.
3. Use cross-validation to find the best combination of the kernel parameters γ and C, from a predetermined set.
4. Use the best parameter of γ and C to train the whole training set.
5. Test the prediction for a test set.

3.1 Data Set

The data set selected for training and prediction of the model are the closing prices of Haier of the Shanghai Stock Exchange between December 2009 and November 2010. There are two different feature set that were extracted from this data set:

i) For the first feature set the days open price, the high and low for the day were used to train the model using the training set and then predict for the test set.
ii) For the second feature used for prediction, only the closing prices were used to calculate an entirely new input and output parameters that were used for training and prediction. The calculation of these new parameters is shown in the section below.

3.2 Data Preprocessing

To enhance the predicting ability of the model, in the second feature set, the original closing prices were transformed into relative difference in the percentage (RDP) of the price compared to the previous days.

The input variables determined using RDP are (RDP-5, RDP-10, RDP-15 and RDP-20), and also one transformed closing price calculated by subtracting a 150-day exponential moving average from the present day closing price.

The output variable RDP+5 is obtained by first smoothing the closing price with a 3-day exponential moving average.

Indicator	Calculation
Input variables	
EMA15	$P(i) - \overline{EMA_{15}(i)}$
RDP-5	$(p(i) - p(i-5))/\, p(i-5) * 100$
RDP-10	$(p(i) - p(i-10))/\, p(i-10) * 100$
RDP-15	$(p(i) - p(i-15))/\, p(i-15) * 100$
RDP-20	$(p(i) - p(i-20))/\, p(i-20) * 100$

3.3 Cross-Validation

The two parameters γ and C, are not know beforehand which ones are best suited for this problem for prediction. The goal is to identify the best γ and C so that the classifier can accurately predict the output of the unknown data or the testing data.

Therefore the common methodology used in such processes is to divide the training set into different subsets and train the model for different γ and C and see which one gives the least error or performs the best on the n^{th} subset, and use these obtained values of γ and C on the unknown test set.

4. RESULTS

The values used in the implementation for γ and C are:

Γ = {(2 * exp(-9)), (2 * exp(-6)), (2 * exp(-3)), (2 * exp(0)), (2 * exp(3))}
C = (2 * exp(-3)) (2 * exp(0)) (2 * exp(3)) (2 * exp(6)) (2 * exp(9))

1. For the first implementation using the open prices, high and low

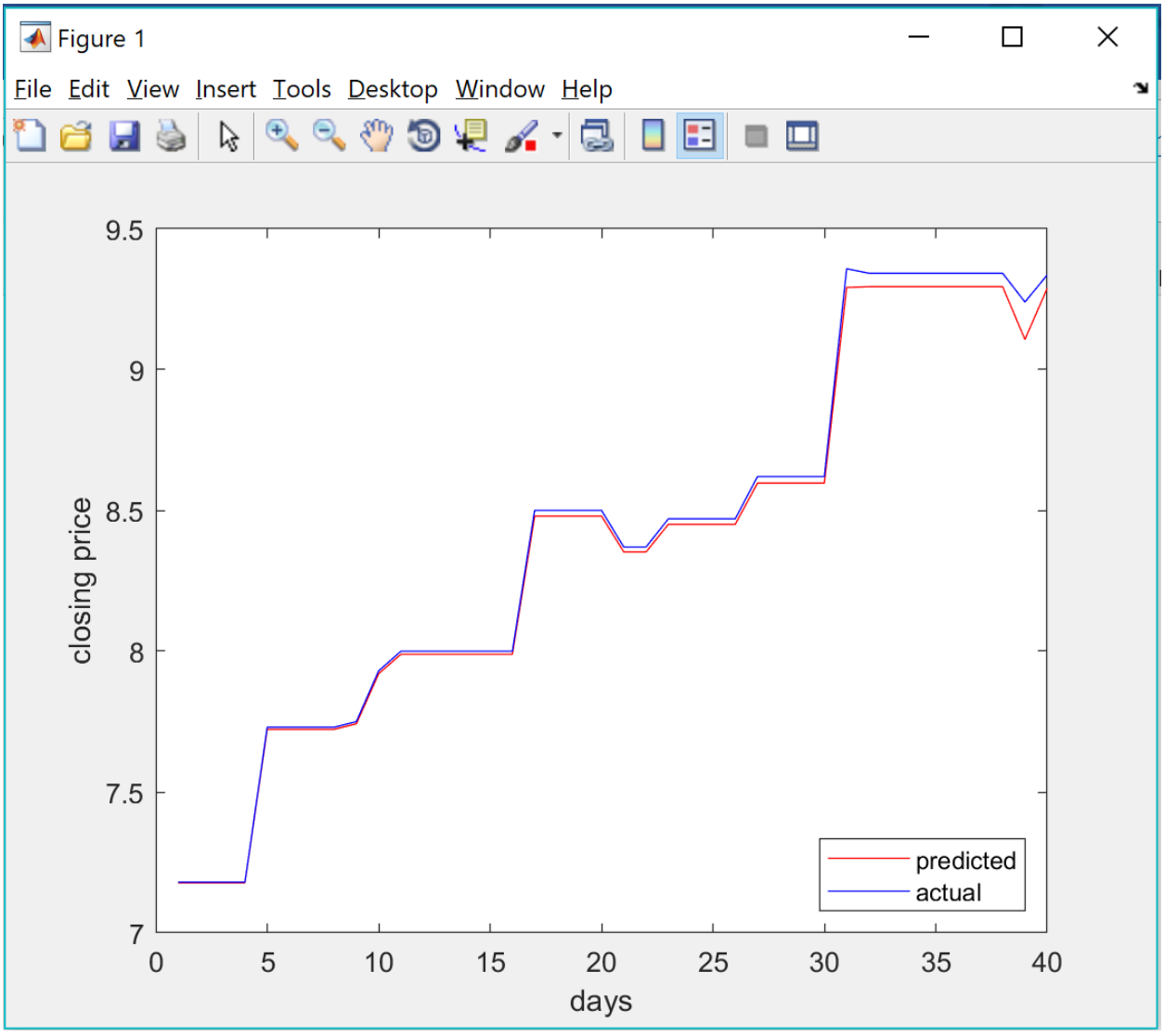
Average mean square errors for each combination of γ and C

Combination 1	Mean square error
1	0.0886705269789691
2	0.0309490735158578
3	0.0309481374482458
4	0.0309480919749073
5	0.850461841820145
6	0.0886818264336110
7	0.0309453817024283
8	0.0309452204804673
9	0.0309451996540347
10	0.304834995723060
11	0.0864678269204940
12	0.0178666338905375
13	0.0178790167651207
14	0.0178792190492365
15	0.427087112567122
16	0.105551861762116
17	0.00476336529965669
18	0.00474985046164684
19	0.00475045006978129
20	0.00474980349592257
21	0.452668263618973
22	0.370650078659101
23	0.00352141498930254
24	0.00411038480767247
25	0.00412309338107520

The minimum mean square error is 0.00352141498930254 for

γ = 40.1710738463753

C = 40.1710738463753



2. For the implementation using the feature set obtained from the closing prices:

Average mean square errors for each combination of γ and C

Combination 1	Mean square error
1	0.444811564461241
2	0.448803431577946
3	0.448803404144562
4	0.448803401740255
5	0.448803401809628
6	0.445100758839860
7	0.448598377173613
8	0.448598923108671
9	0.448598930287553
10	0.448598931228250
11	0.406713006027292
12	0.299691986054915
13	0.299831048765012
14	0.299831091951486
15	0.299831104726072
16	0.345425799848966
17	0.186923733032108
18	0.186977000527017
19	0.186976986918830
20	0.186976995853671
21	0.296635178138990
22	0.185893943535002
23	0.177999541625880
24	0.174157066727753
25	0.540477393708098

The minimum mean square error is 0.174157066727753 for

γ = 40.1710738463753

C = 806.857586985470

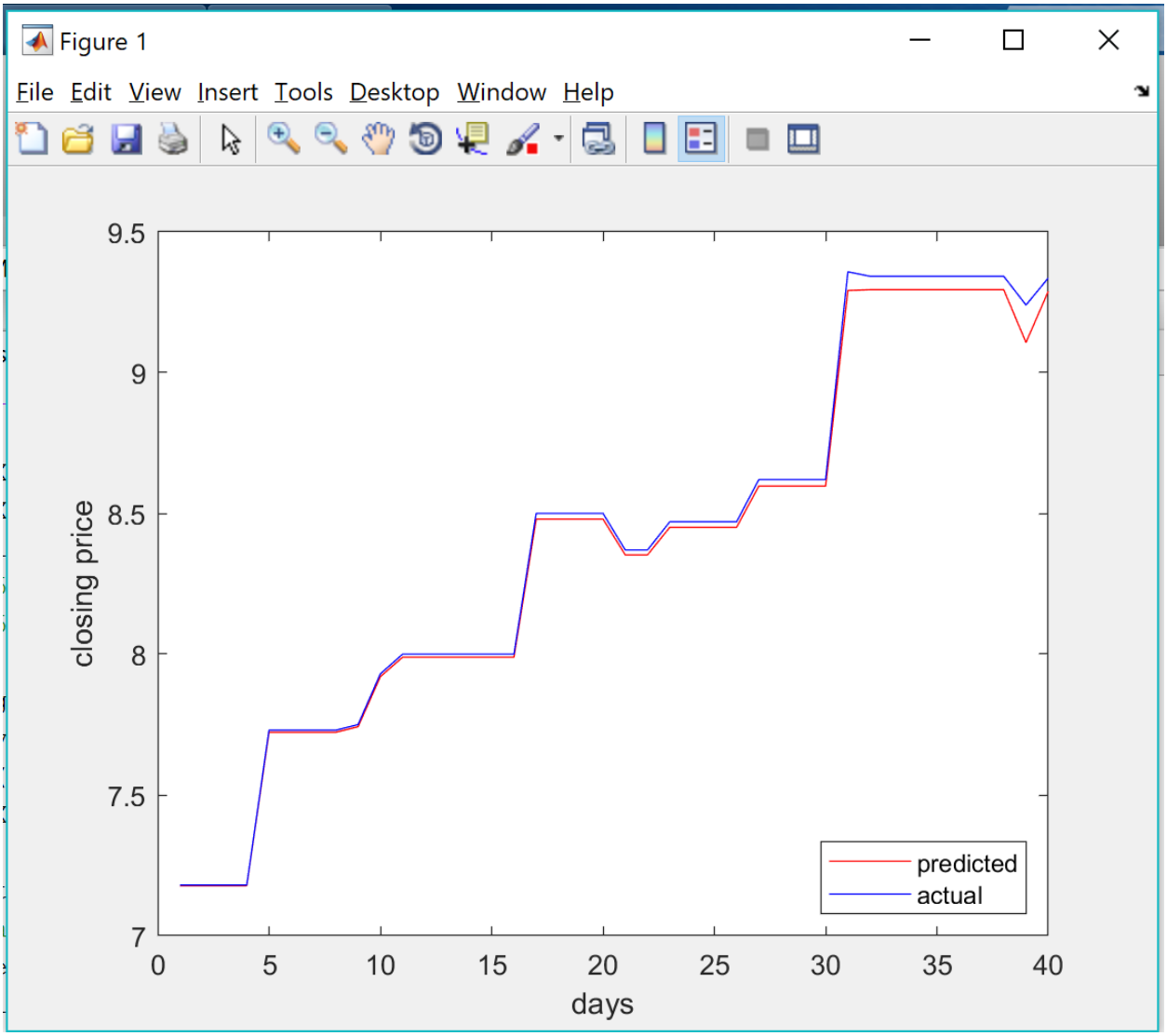


Figure 1
File Edit View Insert Tools Desktop Window Help
9.5
9
8.5
8
7.5
7
closing price
0
5
10
15
20
25
30
35
40
days
predicted
actual

5. REFERENCES

[1] Yukun Bao, Yansheng Lu, Jinlong Zhang. Forecasting Stock Prices by SVMs Regression. Springer-Verlag Berlin Heideberg (2004).

[2] Thomason M. The practitioner methods and tool. Journal of Computational Intelligence in Finance 1999;7(3):36–45.

[3] https://en.wikipedia.org/wiki/Support_vector_machine

[4] http://www.stockstar.com/

[5] https://www.nasdaq.com/symbol/hrely/historical