MACHINE LEARNING FOR APPLE STOCK PREDICTIONS

Sangeeta Kumari, Rui Li

The Ohio State University

ABSTRACT

The following paper compares different machine learning techniques for prediction of closing price of Apple stocks. The report describes various algorithms like regression, Kalman filter, maximum likelihood estimation with Generalized Auto Regressive Conditional Heteroskedasticity Model, recurrent neural networks (RNNs) and compares their accuracies and complexities. Since RNNs have a memory attached, we chose it for the problem, expecting it to give better results. To avoid the vanishing gradient problem in RNN, we chose Long Short-Term Memory (LSTM). It was noticed that stochastic linear regression gave the least mean square error.

Index Terms— stock prediction, LSTM, regression, Kalman filter, time series

1. INTRODUCTION

Stock predictions is a well-researched time series prediction problem and various algorithms have been used in past to predict the future closing price. Support Vector Machines (SVM) [1] and Deep multilayer perceptrons [2] have heavily been used to tackle the stock predictions but they do not have any notion of temporal memory. Therefore, recurrent neural network (RNN) [3] sounds like a better option for such problems but to overcome the difficulties with training RNNs, LSTM [4] is used and has been shown to perform better than other methods.

The logarithmic return simplifies the computation of continuously compounded return to linear arithmetics, and keeps the property of time series [5]. Several models in quantitative finance has been proposed to analyze the implicit relation in logarithmic return time series, such as Autoregressive Moving Average (ARMA) model [6], Autoregressive Conditional Heteroskedasticity (ARCH) model [7] and Autoregressive integrated Moving Average (ARIMA) model [8]. This paper will use two different methods, Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model [9] and Kalman Filter [10], to forecast the logarithmic return as the time series built from closing prices.

Thanks to Dr. Fosler-Lussier as our instructor.

2. DATA DESCRIPTION

Apple's (AAPL) stock data from April 1st 2010 to March 30th 2017 was taken from Yahoo Finance. The data includes date, open, high, close, volume and adjacent close. Last 100 entries in the data was included in the test set and the rest was used for training.

2.1. Data Preprocessing

The data was rescaled using normalization or standardization.

2.1.1. Standardization

Standardization is the rescaling of the data such that the mean and the standard deviation is 0 and 1 respectively. Since this technique assumes that the observation fits a Gaussian distribution, the histogram of values was plotted to verify the nature of data. For training the regression models, the data was normalized using StandardScaler of scikit-learn which standardizes the value using the following formula:

$$x_{std} = (x - \mu)/\sigma \tag{1}$$

where

$$\mu = \sum x/N \tag{2}$$

$$\sigma = \sqrt{2\sum(x - \mu)/N} \tag{3}$$

and N is number of data samples.

2.1.2. Normalization

Normalization is the rescaling of data from the original range so that the values are within the range 0 and 1. For LSTM training, the data was normalized using MinMaxScaler of scikit-learn which normalizes the value using the following formula:

$$x_{nml} = (x - min)/(max - min) \tag{4}$$

where min and max are the minimum and maximum values pertaining to x.

2.2. Time Series and Logarithmic Return

A time series [5] is defined as a sequence of data indexed by ordered time points. This model is widely used in many fields such as weather forecasting, econometrics, and transportation. Two interesting aspects in time series are time series analysis and time series forecasting.

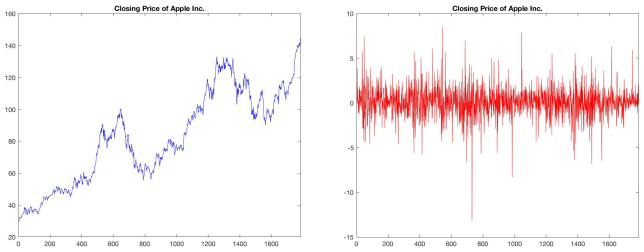
In quantitative finance, the logarithmic return is usually used in analyzing the rate of return on investment. Given close prices, the logarithmic return of the stocking is defined as [5]:

$$R = \ln\left(\frac{V_f}{V_i}\right) \tag{5}$$

where V_f denote the closing price for today and V_f is the closing price at the next day. The V_f thus can be obtained by computing

$$V_f = e^R * V_i \tag{6}$$

with given V_i and forecasted R. Figure 1 shows the closing price and logarithmic return of Apple Inc. in past 1800 days.



(b) Closing Price (c) Logarithmic Return

Fig. 1. Closing price and Logarithmic return for Apple Inc.

3. METHODS

3.1. Regression Models

Regression models describe the relationship between a dependent variable, y, and independent variable or predictor variables, X. X is a matrix of observations on predictor variables and is also known as design matrix. For the simplicity of the model, we assume the features of the dataset to be independent of each other.

3.1.1. Linear Regression

A general linear regression model is as follows:

$$y_i = w_0 + \sum_{j=1}^{p} w_j x_{ij} \tag{7}$$

Since we are using dates to predict the future price, our equation gets reduced to:

$$y = w_1 x + w_0 \tag{8}$$

where x is date.

The goal is to find the values of w0 and w1 such that it minimizes the objective function, J (mean square error) which is as follows:

$$J(w) = \frac{1}{2}\sum_i (y_i - (w_1 x_i + w_0))^2 \tag{9}$$

3.1.2. Stochastic Gradient Descent (SGD)

The coefficients used in simple linear regression equation (7), can be found by gradient descent (GD).

$$w_i = w_i - \alpha \frac{dJ}{dw_i} \tag{10}$$

In case of very large datasets, using GD can be quite costly since we only take a single step for one pass over the training set - thus, the larger the training set, the slower the algorithm updates the weights and the longer it may take until it converges to the global cost minimum. Therefore, we use Stochastic Gradient Descent (SGD) where instead of accumulating the weight updates, we update the weights after each training samples.

3.1.3. Support Vector Regression (SVR)

In SVM regression, the input X is first mapped onto a m-dimensional feature space using some fixed (nonlinear) mapping, and then a linear model is constructed in this feature space. Using mathematical notation, the linear model (in the feature space) $f(x, w)$is given by:

$$f(x, w) = w \cdot x + b \tag{11}$$

Since our data has been standardized to zero mean, the bias term is dropped. In $\epsilon - SV$ regression, our goal is to find $f(x)$ that has at most ϵ deviation from the actual target values y_i for all training data, and is also flat. Flatness of (6) means a small w. One way to ensure this is to minimize the Euclidean norm i.e. $||w||^2$. This can be written as a convex optimization problem by requiring:

$$minimize : \frac{1}{2}||w||^2 \tag{12}$$

subject to:

$$y_i - w \cdot x_i \leq \epsilon \tag{13}$$

$$w \cdot x_i - y_i \leq \epsilon \tag{14}$$

Different kernels that we can use to map the data to higher dimensional space are radial basis function and polynomial functions.

For all the regression methods, 5-fold cross validation was done to avoid overfitting.

3.2. Generalized Auto Regressive Conditional Heteroskedasticity

The generalized autoregressive conditional heteroskedasticity process is proposed by Robert F. Engle in 1982 and won Nobel Memorial Prize for Economics at 2003. This model is built to provide a mathematical description for estimating volatility in financial markets [9]. The GARCH process is an extension of ARCH process. Let r_t denote the logarithmic return at time t, F_t is all history information relative to time periods from start to t, then the conditional expectation μ and conditional variance σ^2 of r_t is:

$$\mu_t = E[r_t|F_t - 1] \tag{15}$$

$$\sigma^2 = E[(r_t - \mu_t)^2|F_{t-1}] \tag{16}$$

Assume that r_t is a simple time series, such as ARMA(p, q) [6].

$$r_t = \phi_0 + \sum_{i=1}^{p} \phi_i r_{t=i} - \sum_{i=1}^{q} \theta_i a_{t-i} + a_t \tag{17}$$

where error term a_t is distributed random variables, and in typical ARMA model, a_t is sampled from a normal distribution: $N(0, \sigma^2)$.

Then GARCH model construct relationship between the square of variance of residual sequence a_t.

$$a_t = u_t \sqrt{h_t} \tag{18}$$

$$h_t = k + \sum_{i=1}^{q} G_i h_{t-i} + \sum_{i=1}^{p} A_i \epsilon_{t-i}^2 \tag{19}$$

To satisfy the fat tail property of return rate, the u_t are usually sampled from normal distribution or student-t distribution. The GARCH process can measure the implied volatility of a time series due to price spikes, such as the huge price spike during the simmer of 2000 in California [11].

3.3. Kalman Filter

A Kalman Filter [10] is used to estimate a discrete time controlled process variable $x \in R^n$. The discrete time controlled process is denoted by a linear stochastic difference equation:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{20}$$

, with a measurement $z \in R^m$ which is

$$z_k = Hx_k + v_k \tag{21}$$

,

w_k and v_k are process noise and measurement noise respectively. They are assumed to be independent, white, and with normal probability distributions

$$p(w) \sim N(0, Q) \tag{22}$$

$$p(v) \sim N(0, R) \tag{23}$$

The algorithm to compute Kalman Filter for estimating variable x includes two parts as stated in paper [10]. The first part is for time updating, which predict the results in the next time point. Let $\hat{x_k}$ denote the posteriori state estimate at step k with measurement z_k, and let $\hat{x}_k^-$ denote the priori state estimate at step k given knowledge of the process prior to step k. The time update process can be represented as,

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \tag{24}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{25}$$

The second part is measurement update, which is calculated by,

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{26}$$

$$\hat{x}_= \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{27}$$

$$P_k = (I - K_k H)P_k^- \tag{28}$$

3.4. Long Short-Term Memory (LSTM)

Recurrent neural networks have a cycle which feeds activations from the previous time step back in as an input and influences the activations of the current time step. This allows these networks to store temporal information for a dynamic indefinite number of steps in contrast to the fixed number of time steps of feed forward networks.

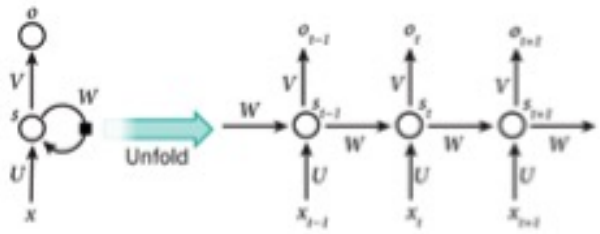


Fig. 2. LSTM Network Structure

LSTM is a type of RNN that overcomes the problem of vanishing gradient problem. Instead of neurons, LSTM networks have memory blocks that are connected through layers.

A block contains gates that manage the block's state and output. They operate upon an input sequence and each gate within a block uses the sigmoid activation units to control whether they are triggered or not, making the change of state and addition of information flowing through the block conditional. There are three types of gates within a unit:

- Forget Gate: conditionally decides what information to throw away from the block.
- Input Gate: conditionally decides which values from the input to update the memory state.

- Output Gate: conditionally decides what to output based on input and the memory of the block.

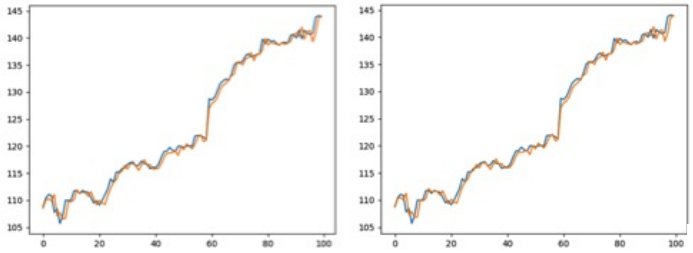
Each unit is like a mini-state machine where the gates of the units have weights that are learned during the training procedure.

4. RESULTS

Table 1 shows the Root Mean Square Error (MSE) of all the techniques. Figure 3 to Figure 7 shows the final forecast result for each of methods used in this paper.

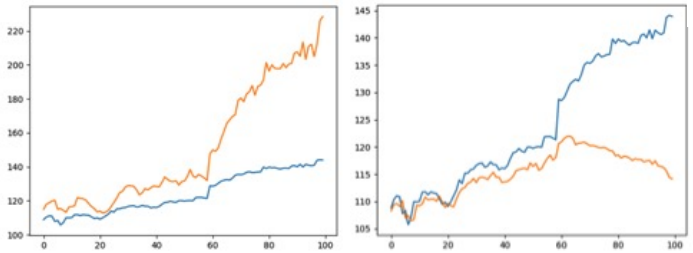
Technique	Train RMSE	Test RMSE
Linear Regression	1.10	1.03
SGD Regression	1.12	1.12
SVR with RBF kernel	1.29	12.61
SVR with Polynomial kernel	11.44	35.68
GARCH	-	1.59
Kalman Filter	-	1.18
LSTM	2.14	3.54

Table 1. Root Mean Square Error for all methods



(a) Linear Regression (b) SGD Regression

Fig. 3. Forecast result of Linear Regression and SGD Regression



(a) Polynomial Kernel (a) RBF Kernel

Fig. 4. Forecast result of SVR with different kernels

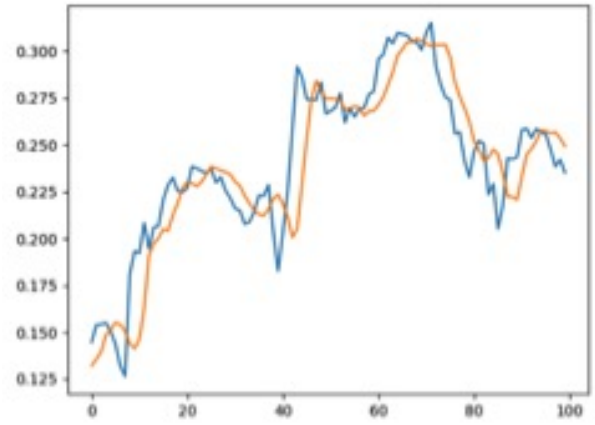
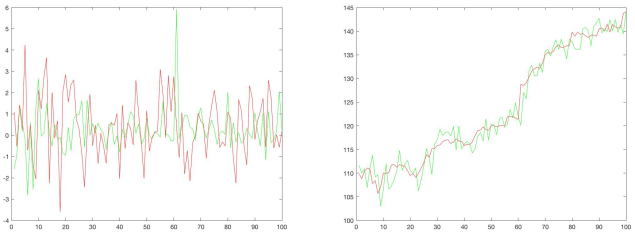
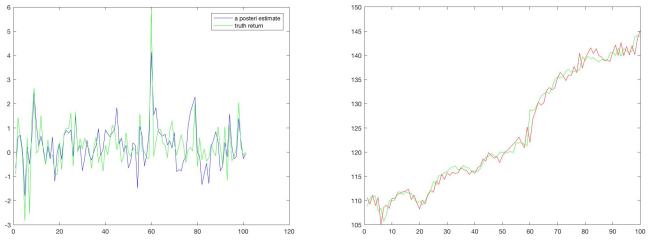


Fig. 5. Forecast result of LSTM



(a) Estimation of Log-Return (b) Closing Price Forecast

Fig. 6. Forecast result of Garch(1,1) Model. (a) shows the forecast of the logarithmic return and (b) is the estimation of closing price computed accordingly from (a).



(a) Estimation of Log-Return (b) Closing Price Forecast

Fig. 7. Forecast result of Kalman Filter. (a) shows the forecast of the logarithmic return and (b) is the estimation of closing price computed accordingly from (a).

5. REFERENCES

[1] Kyoung-jae Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1, pp. 307–319, 2003.

[2] Herbert Jaeger, "The ?echo state? approach to analysing

and training recurrent neural networks-with an erratum note," *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, no. 34, pp. 13, 2001.

[3] Armando Bernal, Sam Fok, and Rohit Pidaparthi, "Financial market time series prediction with recurrent neural networks," 2012.

[4] Felix Gers, *Long short-term memory in recurrent neural networks*, Ph.D. thesis, Universität Hannover, 2001.

[5] Ruey S Tsay, *Analysis of financial time series*, vol. 543, John Wiley & Sons, 2005.

[6] Allan I McLeod and William K Li, "Diagnostic checking arma time series models using squared-residual autocorrelations," *Journal of Time Series Analysis*, vol. 4, no. 4, pp. 269–273, 1983.

[7] Tim Bollerslev, "Modelling the coherence in short-run nominal exchange rates: a multivariate generalized arch model," *The review of economics and statistics*, pp. 498–505, 1990.

[8] Steven C Hillmer and George C Tiao, "An arima-model-based approach to seasonal adjustment," *Journal of the American Statistical Association*, vol. 77, no. 377, pp. 63–70, 1982.

[9] Christopher G Lamoureux and William D Lastrapes, "Heteroskedasticity in stock return data: volume versus garch effects," *The journal of finance*, vol. 45, no. 1, pp. 221–229, 1990.

[10] Greg Welch and Gary Bishop, "An introduction to the kalman filter," 1995.

[11] Reinaldo C Garcia, Javier Contreras, Marco Van Akkeren, and João Batista C Garcia, "A garch forecasting model to predict day-ahead electricity prices," *IEEE transactions on power systems*, vol. 20, no. 2, pp. 867–874, 2005.