CS230

Exploration of predicting power of ARIMA,Facebook Prophet and LSTM on time series

Yi Jiang, Vicki Jingyue Zhang
Department of Computer Science
Stanford University
yijiang1, vjz1140@stanford.edu

Abstract

The goal of our project is to predict the future price movements of securities using historical information. We used S&P 500 (SPY) daily data starting from January 1950 to December 2018 and the Federal Reserve (the Fed) minutes schedule from 1997 to 2020 to make price predictions applying autoregressive integrated moving average (ARIMA), Facebook Prophet (fbprophet) and long short-term memory (LSTM). We de-noised closing prices using Daubechies wavelet and Kalman filter and differenced the price series as stationary input to LSTM. Trading decisions will then be made based on the price prediction. After comparing the errors and trading performance, we conclude the combination of Kalman filter and LSTM are superior.

1 Introduction

Predicting future securities price movement has been a widely studied topic in the financial industry. Without a doubt, correctly forecasting the price actions can generate tremendous profits. However, due to many noises and complex coherent nonlinear correlations, traditional linear regression models cannot deliver sound predictions. Therefore, we decided to explore other prediction methods to achieve better accuracy. We began with ARIMA (1,1,0) using SPY daily adjusted closing price as the regressors, parameters (1,1,0) are chosen based on stationarity, autocorrelation and partial autocorrelation of the regressors. After seeing the erroneous predictions, we employed Facebook Prophet with SPY daily adjusted closing price and Fed minutes meeting schedule as independent variables. Lastly, we applied LSTM using SPY daily log returns. We studied the Morlet wavelet heat spectrum of closing prices from 1950 to the present, to help select the seasonality period for Facebook Prophet and the input window length for LSTM. To better assist LSTM to discover vital relations from price data, we de-noised the data using two approaches, Daubechies wavelet, and Kalman filter to compare the results with non-filtered series. Trading decisions were then made based on predictions from Facebook Prophet and LSTM, followed by trading performance evaluation.

2 Related work

L.Weng implemented RNN [1] to predict price by feeding sliding two consecutive three-day windows to predict the next three-day window. Inspired by her approach, we applied twelve sliding five-day windows (60 days in total) to predict the next five-day window. Sixty as window size was recognized by wavelet heat spectrum which captures low-frequency cyclical signals [2]. Regarding ARIMA, R.Nau provided an extensive tutorial [3] on how to process data and choose the parameters. We implemented ARIMA according to R.Nau's approach by examining the data stationarity using

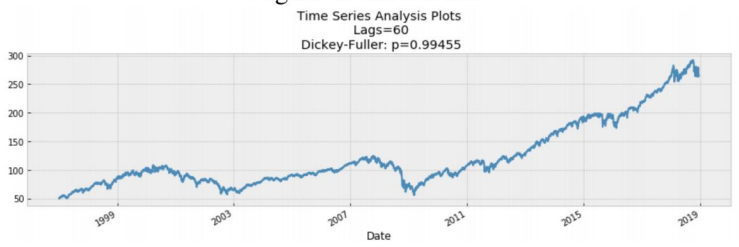
CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

Dickey-Fuller and autocorrelation (ACF) and partial autocorrelation (PACF). For Facebook Prophet, Y.Pan proposed a method [4] to apply the model on stock price time series. To fully utilize Facebook Prophet's power, we also introduced the Fed minutes meeting schedule as holidays to examine the "holiday effect".

3 Dataset and Features

We gathered SPY historical daily adjusted closing prices from Yahoo! Finance ranging from January 1950 to December 2018, and the Fed minutes schedule from Bloomberg terminal from 1997 to 2020. In ARIMA we applied the daily prices from 1997 to 2018 (5.5k data points in total) to fit the curve. In Facebook Prophet, we split daily price data from 1997 to 2014 as training set and 2014 to 2018 as the testing set. In LSTM, we pre-processed the price data using Daubechies wavelet and Kalman filter as two separate approaches, then derived log return on the de-noised data to feed LSTM. We used the same training/test split in LSTM as in fbprophet to better compare the performance.

Figure 1: Time Series



4 Methods

The general algorithm for Autoregressive Integrated Moving Average (ARIMA) is:

$$\hat{Y_t} = \mu + \phi_1 * Y_{(t-1)} + ... + \phi_p * Y_{(t-p)} + \theta_1 * e_{(t-1)} - ... - \theta_q * e_{(t-q)}$$

ARIMA model forecasts a time series by making it stationary and using lags of dependent variables and/or lags of forecast errors as predictors. In ARIMA(p,d,q) model, p is the number of autoregressive terms, d is the order of differences required for stationarity, and q is the number of lagged forecast errors. AR term (p) can be interpreted as "partial difference" operator and MA term (q) as "partial cancel" operator for an order of difference. Therefore AR term is preferred when a series is "under-differenced" and MA term is preferred when "over-differenced," and the two terms could cancel each other out.

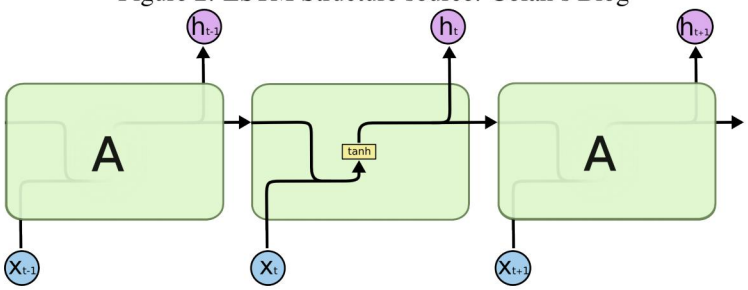
Facebook Prophet is capable of automatically detecting trend changes and decomposing data into trend, seasonal, and holiday effect, whose related function is:

$$\hat{Y_t} = g(t) + s(t) + h(t) + \epsilon_t$$

Trend changes are incorporated in the model by explicitly defining changing points and applying sparse priors that conform to Laplace(0, τ) so that changes in trend could be automatically added. By using Fourier transform, the seasonal effect in time series can be extracted. Regarding holiday effect, the model incorporates holiday calendars into the forecast.

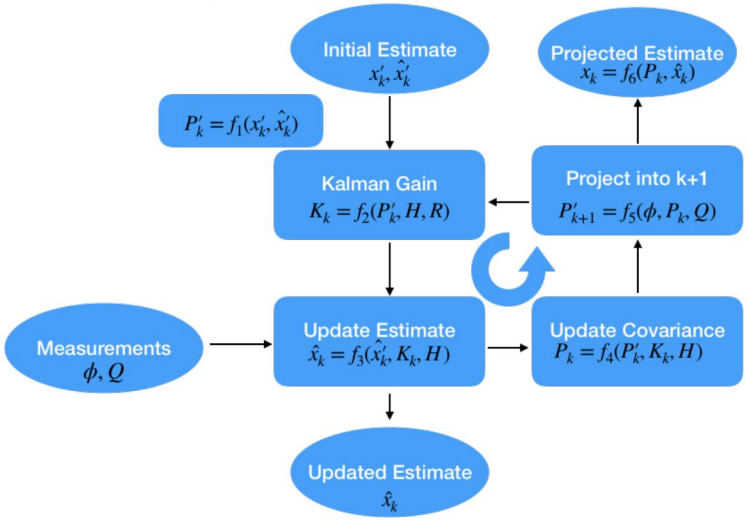
LSTM model is a unique form of recurrent neural network (RNN) that can learn long-term dependencies without suffering from gradient vanishing by implementing forget gates and input states. In each state, sigmoid function drops out certain input from previous states and the tanh function filters new input to include in the model. The framework is shown in Figure 2.

Figure 2: LSTM Structure source: Colah's Blog



We used two approaches to de-noise the data. Daubechies wavelet, an orthogonal wavelet, has an advantage in which the signal can be presented compactly because the number of convolutions at each scale is proportional to the width of the wavelet basis at that scale [5]. We also used Kalman filter, whose framework is shown in Figure 2.

Figure 3: Kalman Filter Framework

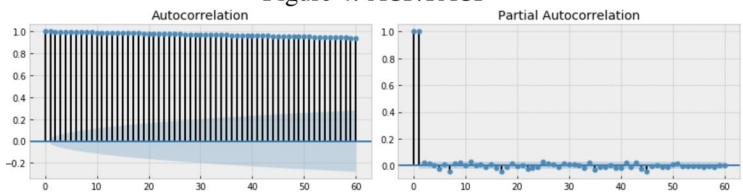


It is one of the simplest Bayesian networks where the true values are estimated over time by including incoming measurements in the equation, where the probability of predicted states is associated with measurements up to the current step. Kalman filter is a recursive calculation method to estimate the state of a process by minimizing mean of squared error [6].

5 Experiments/Results/Discussion

For ARIMA, based on standard approach, we set parameters to (1,1,0) (Figure 3). The process began with determining the necessary order of differencing to achieve stationary and remove seasonality. From Dickey-Fuller test we noticed price data is not stationary while the first order of differencing is stationary; therefore we picked d=1. Then we chose p and q by using ACF/PACF combination. In general, ACF shows the correlation between a time series and lags of itself, and PACF captures the correlation of times series of its lags that is not explained by correlations at lower-order-lags. Take AR signature ACF/PACF combination as an example, a sharp drop in PACF chart at lag k shows no significant explanation power from partial autocorrelation beyond lag k, and gradual change in bar length in ACF indicates a better explaining power from AR term. From the ACF/PACF chart, we picked p=1 and q=0.

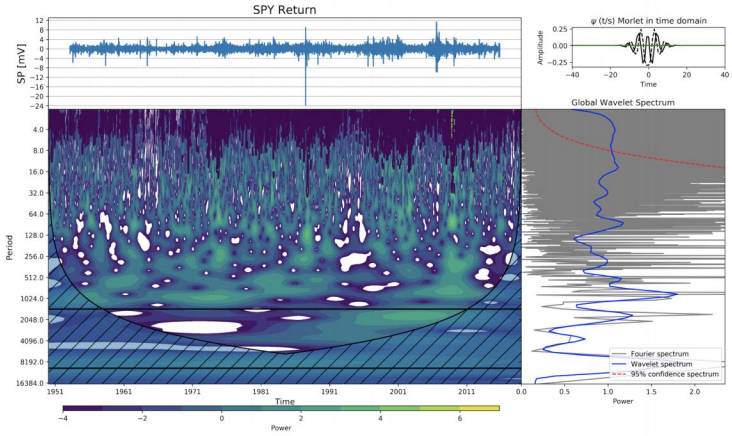
Figure 4: ACF/PACF



However, the results were not satisfying, especially when there was a trend change. Moreover, the result was highly sensitive to the selection of regression window length and period choice, because the corresponding ACF/PACF charts were shaped differently, making it difficult to pick proper parameters. Therefore we started to work with another state-of-the-art method.

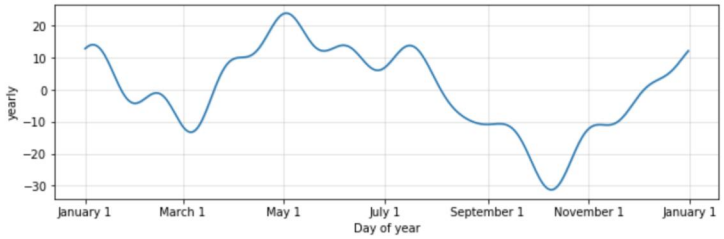
To better understand the data and pick hyper parameters, we applied wavelet heat spectrum based to discover infrequent cyclical signals. Specifically, the wavelet is Morlet wavelet, a complex nonorthogonal function, that contains more oscillations, combining both positive and negative peaks into a single broad peak. In terms of the choice of parameters, it is based on empirical derivation [7]. The basic idea is to pick a scale that gives finer resolution and still gives a big enough sampling scale. The reason why we did not choose Daubechies to generate heatwave spectrum is that spectrum generated by orthogonal wavelet analysis contains more discrete blocks, making it visually challenging to interpret the choppy wavelet spectrum.

Figure 5: Morlet Heat Wave Spectrum



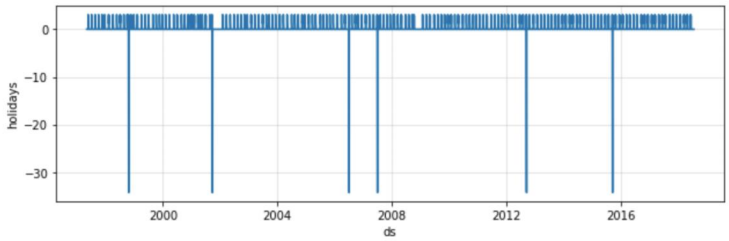
In Facebook Prophet, we set length of yearly seasonality as 1000 trading days and quarter seasonality as 60 trading days. The reason is that in Morlet wavelet heatwave spectrum, we could see a detectable cyclical signal in 60 days and 1000 days. After applying Facebook Prophet, we noticed strong monthly seasonality and holiday effect in our data. From Figure 6 we could see a notable downtrend from May to September, and an uptrend in November. The former supports the trading adage "Sell in May and Go Away," the latter supports "Halloween effect."

Figure 6: Seasonal Effect - Seasonal Anomalies



From Figure 7, it is easy to see the Fed minutes meeting has a significant negative influence on market movements.

Figure 7: Holiday Effect - Fed Minutes Meeting Schedule



In LSTM, we chose two hidden layers of size 256x128 based on experiment results. The dropout rate is 0.1, and epoch is 200. Training and testing split is 80/20 from 1997 to 2018. Mini-batch size is five because the change over the five-day window is supposed to be less noisy and overcome some market irrationality based on our understanding of the financial market. The evaluators we use are training error, test error, and trading performance. Additionally, the rule to initialize trading positions could be summarized as buy/hold if the prediction is higher in 5 days when there is no/existing position, sell/no action if the projection is lower when there is current/no position.

To research the necessity of de-noising data, we compared results on filtered data and non-filtered data. From test error, we could see the accuracy is better or no worse than non-filtered version. Between Kalman filtered version and non-filtered version, though the test errors are close, the trading performance differs by a large scale, where Kalman generated 25.93%, and non-filtered make -10.98%. Concerning over/underfitting, training errors are all higher than test errors across all approaches. We interpret the results as the choppier market movement in training period than test period; also we set dropout to avoid overfitting. Therefore, there is no evidence showing our models overfit the data.

Table 1: Facebook Prophet and LSTM Results

	Training Error	Testing Error	Testing Period Performance
Facebook Prophet	N/A	7.49E-05	-2.29%
LSTM w/o Filter	9.62E-03	7.76E-05	-10.98%
LSTM w/ Wavelet Filter	1.53E-04	8.45E-05	10.55%
LSTM w/ Kalman Filter	1.62E-04	7.74E-05	25.93%

6 Conclusion/Future Work

From training error, test error, and trading performance, "Kalman and LSTM" combination generates the best result. The superior result mainly comes from that Kalman filter effectively removes noise and makes the trend more detectable, and LSTM is capable of learning long-term nonlinear dependencies. To prove our understanding and further enhance the performance, the future work will include larger data scope and more data features. More single name stocks from different industry sectors and geographic regions will be put into the test. Meanwhile, we will apply multivariate models such as Vector Auto Regression (VAR) to process data and selectively add multi features in the learning algorithm to better capture information from available data.

7 Contributions

We are proud to present the project, which is truly a labor of love and a result of top-notch teamwork. Yi Jiang, a software engineer in Fintech and loving dad of three little boys, leads the project effectively while coping with demanding day job and family responsibility. Yi and Vicki, the group of two engineers, work side by side, putting in hours of serious effort and thought in researching, coding and report writing. We would love to express our gratitude to the course instructors, teaching assistants (especially Cristian Bartolome) for empowering us with deep learning knowledge and providing

helpful guidance in building the project. We will continue the journey wholeheartedly and sincerely hope to come up with a reliable model to change the landscape of investment analysis.

References

Library: Keras, Fbprophet, Waipy, Pykalman, Statsmodels & Pandas

T.Vincent https://www.digitalocean.com/community/tutorials/a-guide-to-time-series-forecasting-with-arima-in-python-3

Y.Pan 2018 https://towardsdatascience.com/analysis-of-stock-market-cycles-with-fbprophet-package-in-python-7c36db32ecd0

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bazil A. Sansom (2017) *Re-thinking the housing-macro nexus: housing market churn in aggregate demand formation and stability, a wavelets based analysis.* 21st FMM Conference "The Crisis of Globalisation"

[3] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[4] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.

[5] S. C. Shiralashetti (2014) *An application of the Daubechies Orthogonal Wavelets in Power system Engineering* P. G. Department of Studies in Mathematics

[6] Greg Welch and Gary Bishop (2000) *An Introduction to the Kalman Filter* Department of Computer Science University of North Carolina at Chapel Hill

[7] Christopher Torrence and Gilbert P. Compo (1998) *A Practical Guide to Wavelet Analysis* Program in Atmospheric and Oceanic Sciences, University of Colorado, Boulder, Colorado

Code Repository

`https://github.com/yijiang0923/cs230_timeseries_predication`