Stock Closing Price Prediction and Analysis

Introduction

Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. Successful estimation of future stock prices can yield significant profits[1]. Several financial investment decisions are based on such forecasts and analysis.

The aim of this project is to estimate the closing price of stocks using bayesian analysis to aid in investment decisions with regards to purchase/sale of stocks. Closing price of stocks is the final value of stock price before the market closes for the day. Since stock market prediction is prone to fluctuations with a lot of money at stake, bayesian analysis could be very helpful as it can capture uncertainty in parameter estimation and hence in predictions.

Data Analysis

The daily stock price data of Amazon for five years period starting from 1st January 2014 till 6th December 2019 is collected. Data is extracted using get_data_yahoo method in the pandas_datareader package which downloads the stocks data from yahoo finance for a given company and time-frame.

Since the data being considered in this project is the daily closing price of stocks, it can be considered a time series. A time series can be decomposed into these 3 major components (Fig. **??**):

- Trend - Trend tries to capture the slope of the series. It mimics the upward and downward slope of the observed values.
- Seasonality - Seasonality is the constant factor present in the series which is repeated after certain interval of time.
- Residual - Whatever part of the time series is left after removing trend and seasonality is the residual, which is basically random variation.

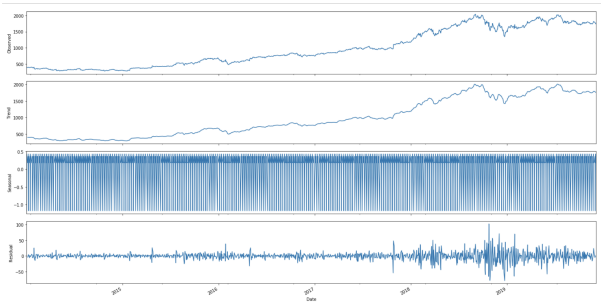


Figure 1: Time Series Component Decomposition

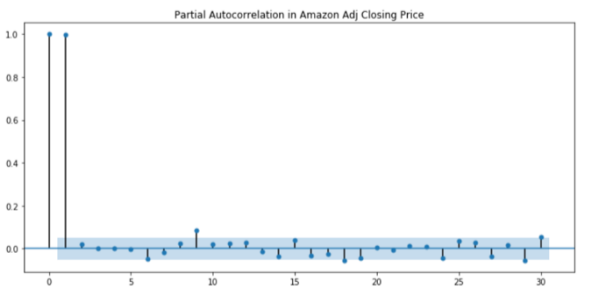


Figure 2: Partial AutoCorrelation

In time series data analysis, estimation of number of past values which can be helpful in predicting the current value is calculated. A Partial autocorrelation plot reveals correlation between different time lags with current value. From (Fig. 2) it can be seen that previous day's value has huge correlation with current day's estimation compared to other past values. So, the lag value should be one.

To validate a lag of one in closing price estimation, Dickey Fuller test is performed on daily percent change of the closing price. If the p value of the test is less than 0.05, then the percentage change time series is stationary. Therefore, it can be inferences that majority of the time varying component in the series is captured within one time lag and the residual is independent of time.

Model Building

Structural time series (STS) are a family of probability models that include many standard time-series modeling ideas, such as autoregression, moving averages, local linear trends, seasonality, and regression. The STS model built in this project expresses an observed time series as sum of following simpler components:

$$f(t) = f_{autoregression}(t) + f_{lineartrend}(t) + f_{seasonality}(t) + \epsilon; \epsilon \sim N(0, \sigma^2)$$

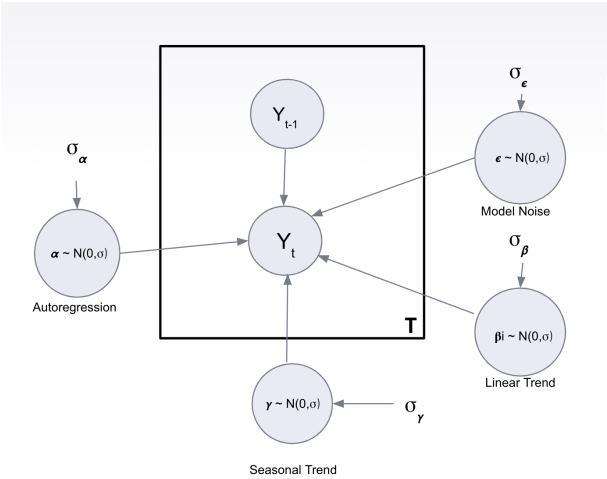


Figure 3: Graphical representation of the model

parameters
α - coefficients for lagged features in Auto Regression Model
β - $slope\_scale(\beta_1)$, $level\_scale(\beta_2)$ are Linear Trend parameters
γ - $drift\_scale$ is Seasonality parameter
ϵ - $observation\_noise\_scale$ is model's noise variance parameter

Table 1: Parameters

The individual components are time series governed by a particular structural assumption. One component encodes a day-of-week seasonal effect, another a local linear trend, and the third one is an autoregressive component to model any unexplained residual effects. A simple random walk could have been used, but an autoregressive component was chosen because it maintains bounded variance over time. After deciding the model components, joint distribution of the data and parameters is optimized using variational inference with mean field approximation.

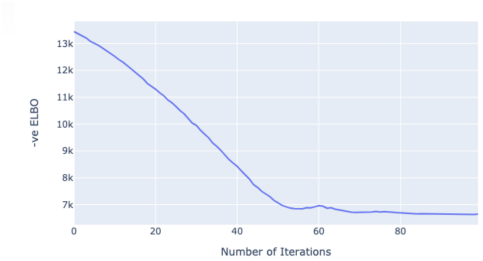
The joint distribution of the parameters is decomposed into product of the distribution of individual components.

$$q(z) = q(\beta_1) \cdot q(\beta_2) \cdot q(\alpha) \cdot q(\gamma) \cdot q(\epsilon))$$

The optimal solution is given as:

$$\log q_j^*(\mathcal{Z}_|) = \mathbb{E}_{i \neq j}[\log p(\mathcal{D}, \mathcal{Z})] + constant$$

The distribution of the individual parameters is assumed to be normal with zero mean. Also the priors used on the parameters are normal and the value of the priors is decided by the tensorflow probability library[4]. A couple of different priors were tried based on heuristics and prior knowledge of stock market but the default priors performed best for prediction. A loss function was defined as negative of ELBO which was optimized using stochastic gradient optimization technique, Adam optimizer. After 50 iterations the algorithm converged and the distribution of the parameters were obtained.



$$ELBO = E_q[\log P(\mathcal{D}, \mathcal{Z})] - E_q[\log q(\mathcal{Z})]$$

$$= \int_z q(\mathcal{Z}) \log P(\mathcal{D}, \mathcal{Z}) d\mathcal{Z} - \int_z q(\mathcal{Z}) \log q(\mathcal{Z}) d\mathcal{Z}$$

Figure 4: Loss Function(-ELBO) optimization

Results

Using mean-field variational approximation we obtain the following posterior distribution for each of the parameters:

parameters	point_estimate	uncertainity
observation_noise_scale	2.75188	0.732038
dayofweek_season/_drift_scale	1.82073	0.0907304
LocalLinearTrend/_level_scale	7.5679	0.535529
LocalLinearTrend/_slope_scale	0.587628	0.0699878
Autoregressive/_coefficients	[0.9311705631955051]	[0.041593167040223394]
Autoregressive/_level_scale	18.4062	0.474753

Table 2: Posterior Estimates of Model Parameters

1000 samples are generated from the posterior distribution of the parameters and further were used to get the predictive distribution of the closing price of stocks over the next 20 days.

This forecasting strategy models the observed time series as a Gaussian state space model and uses filtering to predict the stock price on day T, given the observations until $T-1$. Since we want the predictive distribution over 20 days, the posterior distribution for day $T-1$ acts as the prior distribution for day T[4].

As yesterday's posterior is considered today's prior, from Fig. 5 it is observed that moving further in time, the uncertainty is propagated in the forecasts. Hence, while the initial few estimates are relatively accurate with low uncertainty, the estimates obtained towards the end become more and more inaccurate with very high uncertainty.

To improve the forecast, one-step-ahead predictive strategy is employed. In this approach, given the samples from the posterior over parameters, a predictive distribution is generated over observations at each day T, given the actual observations up till day $T-1$ [4]. From Fig. 6, it can be inferred that the one-step-ahead predictive distribution offers better estimates and all the actual values are within the 95% credible interval.

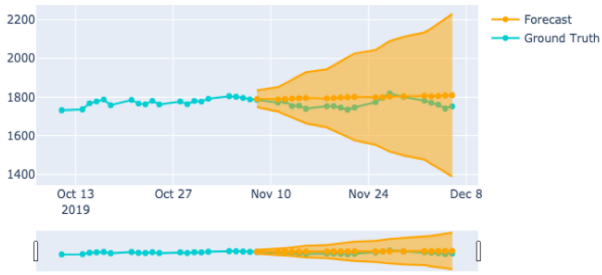


Figure 5: N-step Prediction

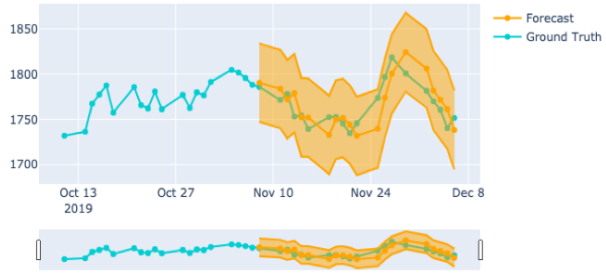
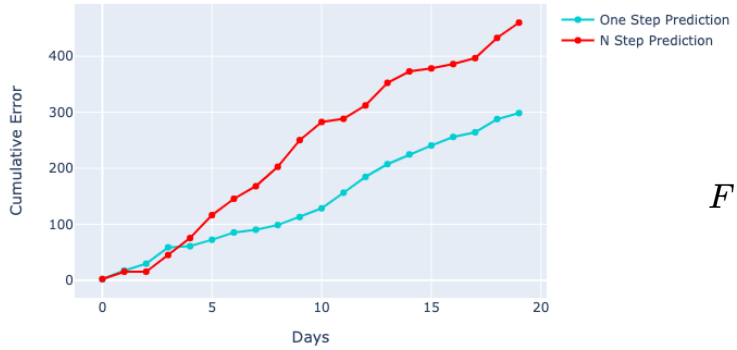


Figure 6: One-step-ahead Prediction

While both the strategies model the series as a guassian state space model and use filtering, the difference is that in the earlier forecasting strategy stock prices for the next 20 days are forecasted only using the samples and training data (01/02/2014 - 11/07/2019), however in one-step-ahead strategy apart from the samples and training data, additional prices are sent up until the day that has to be forecasted. For instance, to get the forecast for November 15th, stock prices between 11/08/2019 - 11/14/2019 will also be used.

Conclusion

Using cumulative absolute one-step-ahead prediction error inferred from S.L. Scott et al.[2] and J. Qio et al.[3], the prediction accuracy from the two forecast strategies are compared. As expected, the cumulative prediction error from the one-step-ahead prediction is lesser compared to the prediction errors using the N-step forecasting (refer Fig. 7).



$$F(Y = T) = \sum_{t=1}^{T} |\hat{Y}_t - Y_t|$$

Figure 7: One Step Cumulative Error Analysis

While the obtained estimates for the stock closing prices are relatively accurate, there are a few instances, wherein the price is forecasted to increase but there is actually a decrease in the stock price. Knowing the relative change of the price is a very important factor in making investment decisions and hence the model forecasts should be improved further. Possible approaches that can be undertaken are:

- Retraining the model everyday to accurately predict the stock prices for tomorrow.
- Using Moving Average component along with the AutoRegressive component will additionally include the effect of the previous residual noise on today's price
- Considering other features like price-earning (P/E) ratio in forecasting stock closing prices.

References

1. About stock market prediction retrieved from https://en.wikipedia.org/wiki/Stock_market_prediction
2. Steven L. Scott, Hal Varian, "Predicting the Present with Bayesian Structural Time Series", International Journal of Mathematical Modeling and Optimization, 2013.
3. Jinwen Qiu, S. Rao Jammalamadaka, Ning Ning, "Multivariate Bayesian Structural Time Series Model", Journal of Machine Learning Research, 2018
4. Tensorflow probability documentation retrieved from https://www.tensorflow.org/probability