Predicting Stock Market Prices with Hidden Markov Models

STATS 201C Final Project

Wesley Cheng

Abstract

This project attempts to use Hidden Markov Models (HMM) to predict the price of Apple (APPL) stock. Past studies provide insight into the optimal number of states to use in a HMM model and for this reason we use a two state model. Our particular prediction model differs from past models in that we use the Bhattacharyya distance on the parameters of our response distributions to predict next day stock price returns. Our results show that this method outperforms both the naive method and alternative prediction approach from past studies. These preliminary results appear to be promising in terms of providing a possible viable stock trading method.

1 Introduction

Predicting prices in the stock market has been a long studied problem as there are potentially very high financial gains to be made. There is a large difficulty in such a prediction because daily and even hourly flucations make stock prices highly volatile. This project explores the use of Hidden Markov Models to model stock price movements. In the past, researchers have used HMMs for similar forecasts. Hassan and Nath use a four state model to model close, open, high, and low prices of various airline stocks [2]. Their prediction is made by looking in the past for a similar day to the current day, and then using day after the similar day to predict the closing price of the future day of interest.

This project is based primarily off Nguyen's implementation [3]. In particular, she uses a two state model and models rolling time windows in the past to use as comparison for current day time windows. Instead of modeling the daily close, open, high, and low prices, which are strongly correlated, she instead uses daily close, open, high, and low returns. Her methodology will be more clearly stated later in the methodology section. This project differs in the way future stock price returns are predicted. As with previous methods, it will use stock price data from the past, however instead of using the probability of a past observation sequence as the criteria to finding a match to the probability of a current observation sequence of interest, we compare the differences in the calibrated normal distribution parameters of our response variables (daily close, open, high, and low returns).

This project is organized as follows. In section 2, we review HMMs to see how it can be applied to our stock price prediction problem. Section 3 provides an overview of the data used, how we use Nguyen's findings to create prediction models in the context of our problem at hand, and how our prediction method differs. Section 4 provides the results of our analysis, and finally, concluding remarks and various potential next steps are discussed in section 5.

2 Hidden Markov Models and Parameter Estimation

Recall from class, we studied HMMs in which our hidden states emitted a singular value. In our problem, the only difference is that our observations are a 4d vector consisting of the daily close, open, high, and low returns. HMMs consist of the following

- Observations, $\boldsymbol{O} = \{O_t^{(l)}, t = 1, 2, \ldots, T, l = 1, 2, \ldots, L\}$ where T is the length of our observed sequence (the 4 returns), and l denotes the number of sequences that we observe in our dataset
- Hidden states sequence, $Z = \{Z_t, t = 1, \ldots, T\}$
- Possible hidden state values, $\{S_i, i = 1, \ldots N\}$
- Transition matrix $A = (a_{ij})$ where each entry in the matrix a_{ij} denotes the probability $P(Z_t = S_j | Z_{t-1} = S_i)$
- Initial probabilites $\pi = (\pi_i)$ where π_i denotes the probability $P(Z_1 = S_i), i = 1, \ldots, N$
- Observation density vector $\boldsymbol{b}_{Z_t}$ which consists of observational densities $b_j^k(O_t^{(l)}) = P(O_t^{(l)} | Z_t = j)$

Our project assumes a Guassian distribution for our each of our response variables. Hence all our b_j^k are Guassian distribution functions. We use the package **depmixS4** [5] to implement our Hidden Markov Model, which provides a Guassian distribution for each of the four responses variables for the two hidden states, for a total of 16 Guassian response parameters to be estimated. This differs from Nguyen's implementation in which only two Gaussians are modeled, one for each hidden state. Our parameters are then given by

$$\boldsymbol{\theta} \equiv \{\pi, A, \mu, \sigma\}$$

We can also let $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3)$ be our parameter vector, with each $\boldsymbol{\theta}_i$ containing the parameters for the prior, transition, and response models. This will be useful when estimating the parameters.

The joint likelihood is given by [5]

$$P(\boldsymbol{O}_{1:T}, \boldsymbol{Z}_{1:T} | \boldsymbol{\theta}) = \pi_i \boldsymbol{b}_{S_t}(\boldsymbol{O}_t) \prod_{t=1}^{T-1} a_{ij} \boldsymbol{b}_{S_t}(\boldsymbol{O}_{t+1})$$

To estimate our parameters, the package **depmixS4** uses the EM alogrithm for unconstrained models.

The joint log-likelihood is given by [5]

$$P(\boldsymbol{O}_{1:T}, \boldsymbol{Z}_{1:T} | \boldsymbol{\theta}) = \log P(Z_1, | \boldsymbol{\theta}_1) + \sum_{t=2}^{T} \log P(Z_t | Z_{t-1}, \boldsymbol{\theta}_2) + \sum_{t=1}^{T} \log P(\boldsymbol{O}_t | Z_t, \boldsymbol{\theta}_3)$$

Since the hidden states are unobserved, we replace them with their expected values given an initial set of parameters $\boldsymbol{\theta}' = (\boldsymbol{\theta}'_1, \boldsymbol{\theta}'_2, \boldsymbol{\theta}'_3)$ and observations. Therefore in our E-step, our expected log likelihood function is [5]

$$
\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \mathbb{E}_{\boldsymbol{\theta}'}(\log P(\boldsymbol{O}_{1:T}, Z_{1:T} | \boldsymbol{\theta}_{1:T}, \theta)) \\
&= \sum_{j=1}^{N} \gamma_1(j) \log P(Z_1 = j | \boldsymbol{\theta}_1) + \sum_{t=2}^{T} \sum_{j=1}^{N} \sum_{k=1}^{N} \xi_t(j,k) \log P(Z_t = k | Z_t = j, \boldsymbol{\theta}_2) \\
&\quad + \sum_{t=1}^{T} \sum_{j=1}^{N} \sum_{l=1}^{L} \gamma_t(j) \log P(O_t^{(l)} | Z_t = j, \boldsymbol{\theta}_3)
\end{aligned}
$$

where $\xi_t(j,k) = P(Z_t = k, Z_{t-1} = j | \boldsymbol{O}_{1:T}, \boldsymbol{\theta}')$ and $\gamma_t(j) = P(Z_t = j)$, both of which can be comuted using the forward-backward algorithm. Since we have seperated our subvectors $\boldsymbol{\theta_1}, \boldsymbol{\theta_2}$, and $\boldsymbol{\theta_3}$ in our Q function, for the M-step in which we maximize for $\boldsymbol{\theta}$, we can seperately maximize for each subvector. **depmixS4** maximizes $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$ using the **nnet** package [4] and $\boldsymbol{\theta_3}$ using `GLM`.

3 Data and Methodology

3.1 Data Collection and Model Assumptions

The **quantmod** makes it very easy to gather historical stock price data in R. For our project we pull close, open, high, and low prices for AAPL stock for the two time periods that Nguyen used. The first time period ranges from January 16, 2015 to October 30, 2015, in which she uses to calibrate the model parameters and calcuate AIC/BIC to determine the optimal number of states (we will also use the time period to generate predictions, Nguyen does not). The second time period ranges from March 22, 2016 to August 10, 2017 so we can make predictions for APPL stock prices from August 15, 2016 to August 11, 2017. This allows us to use the same period of time Nguyen used for her model to compare our results to hers.

3.2 Model Selection

HMMs have three underlying assumptions [6].

- Markov Assumption (memoryless assumption), $a_{ij} = P(Z_{t+1} = j | Z_t = i)$. Next states are only dependent on the current state.
- Stationarity Assumption, $P(Z_{t_1+1} = j | Z_{t_1} = i) = P(Z_{t_2+1} = j | Z_{t_2} = i)$ for any t_1 and t_2. Transition probabilites are independent of the actual time the transition happens.
- Output Independent Assumption, $P(\boldsymbol{O_{1:T}} | \boldsymbol{Z}) = \prod_{t=1}^{T} p(\boldsymbol{O_t} | Z_t, \boldsymbol{\theta})$

In order to satisfy the last assumption, which is often not held valid [6], Nguyen, examines the autocorrelation plots of both the open, low, high, and close prices, and open, low, high, and close returns of AAPL. She shows that ACF plots for the prices exhibit strong persistence, suggesting the stock prices are not independent. When making the ACF plots for the returns, autocorrelations at all lags are found to be small, suggesting that open, low, high, and close returns are independent of each other. Nguyen confirms these results with the Ljung-Box test at the $\alpha = .01$ significance level. Hence from the stock price data we pulled earlier, we compute the daily returns from January 17, 2015 to October 30, 2015 and use these returns to calibrate the model parameters $\boldsymbol{\theta}$.

Model parameters are calibrated by fitting HMMs on $T = 100$ day windows. For example, we first fit a HMM model from January 16, 2015 to June 6, 2015, then January 17, 2015 to June 7, 2015, until our last 100 day block contains October 30, 2015 as the last date, for a total of 100, 100 day blocks. Each fitted HMMs returns eight different normal distributions for a total of 16 parameters, as well as prior and transition probabilties parameters. These gaussian parameters will be used for our prediction method.

Nguyen runs this calibration process for a various number of states and keep track of each window's AIC and BIC values. She finds that the two state model consistently has lower BIC values than three and four state models, while AIC across the various number of states remain almost very close. For this reason, she uses a two state model, which we will assume is the most optimal model as well.

3.3 Prediction Method

The prediction method is where our method differs from past methods. More specifically, our methods differs in the criteria in which we use "match" current time windows with past time windows. Imagine we wish to predict the stock price of AAPL on August 15, 2016. Similar to when we calibrate parameters using the 100 day windows in the historical data, we fit a HMM for the previous 100 day returns, that is, the returns from March 23, 2016 to August 12, 2016 (8/13-14 are non trading days). Nguyen finds the probability of this particular observation sequence $P(\boldsymbol{O}|\boldsymbol{\theta})$ using the calibrated parameters. She then moves the time window back one day (March 22, 2016 to August 11, 2016) and calculates $P(\boldsymbol{O}^{\text{new}}|\boldsymbol{\theta})$. She continutes until she finds a new data set $\boldsymbol{O}^* = \{O_t^{(1)}, O_t^{(2)}, O_t^{(3)}, O_t^{(4)}, t = T^* - 99, \ldots, T^*\}$ where

$$P(\boldsymbol{O}^*|\boldsymbol{\theta}) \simeq P(\boldsymbol{O}|\boldsymbol{\theta})$$

Once $\boldsymbol{O}^*$ and T^* is found, the predicted stock return for August 15, 2016 is

$$O_{T+1}^{(4)} = O_{T^*+1}^{(4)}$$

In other words, we look for a past window that closely resembles the current window and use the following day's return to predict the return of our day of interest. However I found this method slightly unintuitive. Firstly, Nguyen uses $P(\boldsymbol{O}^*|\boldsymbol{\theta}) \simeq P(\boldsymbol{O}|\boldsymbol{\theta})$ as the criteria to finding the past similar window, however she never specifies what the stopping criteria is. How do we know when $P(\boldsymbol{O}^*|\boldsymbol{\theta}) \simeq P(\boldsymbol{O}|\boldsymbol{\theta})$? If we stop too early, it is possible that there exists a $\boldsymbol{O}^{**}$ where $P(\boldsymbol{O}^{**}|\boldsymbol{\theta})$ is closer to $P(\boldsymbol{O}|\boldsymbol{\theta})$. In addition, each preceding time window only differs by one observation point, so the probabilities of consequetive sequences, should be very close to each other, further complicating the issue as to when to stop moving the blocks backward by one day.

I propose an alternative prediction method. Instead of using incrementally moving back one day, I generate a set of training data where each row is a $T = 100$ day block, with associated HMM parameters. Instead of calculating the probability of the particular observation sequences associated with this block, $P(\boldsymbol{O}|\boldsymbol{\theta})$, which should only differ slightly from successive blocks (since observational sequences only differ by one data point), we do not use this probability to match with the current 100 day block of interest. We instead compare the parameters for the two different 100 day blocks to assess "similarity".

More concretely, we use AAPL returns from January 16, 2015 to October 30, 2015 to fit 100, 100 day blocks. For each block, we calculate the sum of Bhattacharyya distances of the 8 response distributions (4 response variables and 2 hidden states) from the 8 response distributions of the current 100 day block of interest. The Bhattacharyya distance between two classes under the normal distribution is computed [1] as

$$D_B(p,q) = \frac{1}{4}\log\left(\frac{1}{4}\left(\frac{\sigma_p^2}{\sigma_q^2} + \frac{\sigma_q^2}{\sigma_p^2}\right)\right) + \frac{1}{4}\left(\frac{(\mu_p - \mu_q)^2}{\sigma_p^2 + \sigma_q^2}\right)$$

We then match our current 100 day block of interest to a historical 100 day block based off the criteria in which the sum of Bhattacharyya distances is at a minimum when compared to other historical 100 day blocks. Our predicted return is made the same way as Nguyen's method by looking at the historical block's next day return. We then multiply the previous day closing price by (1 + predicted return) to predict the closing price.

4 Results

To evaluate our model, we use the same criteria that Nguyen uses in her paper. MAPE, or mean absolute percentage error is calculated as

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|M_i - P_i|}{M_i}$$

where N is the number of predicted points, M is the true AAPL stock closing price, and P is the predicted price of AAPL stock from our model.

Since the EM algorithm converges to a local optimum and our initial parameters contain randomness, the estimated parameters for each 100 day block differs if we rerun the EM

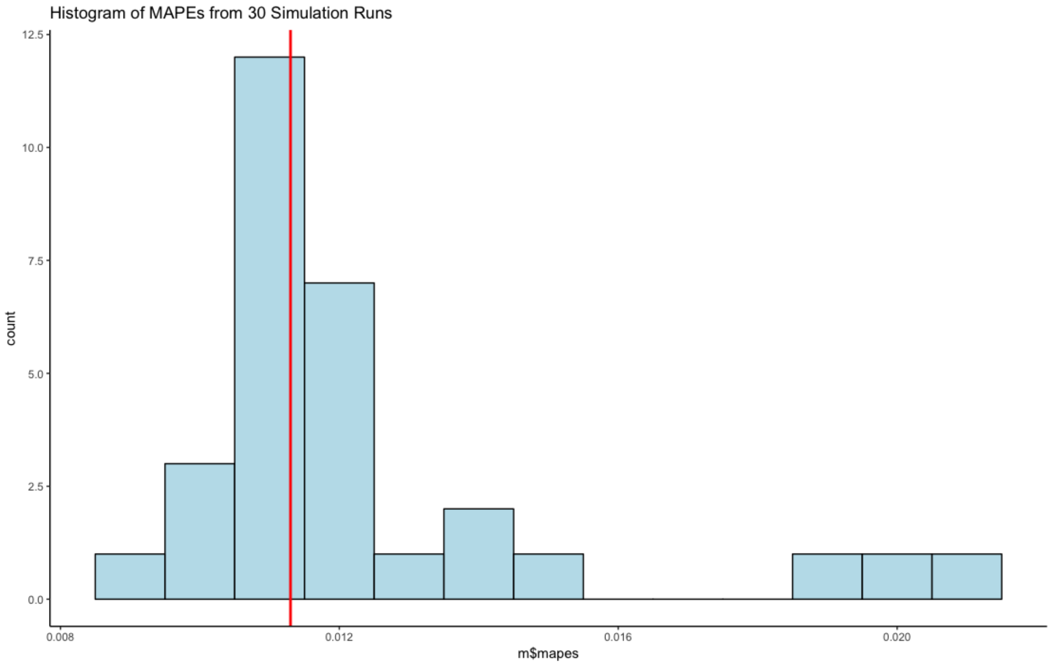

Figure 1: Histogram of MAPEs via 30 simulations with Nguyen's benchmark MAPE

algorithm with a different set of initial parameters. What we found is sometimes our MAPE was very high because consecutive prediction window blocks happened to be matched to the same historical block with an extreme corresponding predicted return. This means certain periods of prediction were using an extreme predicted return to predict the next day price, throwing off the MAPE. One possible solution would be to ignore any matches with a very extreme predicted return and use the next closest match, just because days with say $> 5\%$ or $< -5\%$ are so rare and would be inherently very difficult to predict. Instead of ignoring these matches, we find that running multiple simulations and averaging the predicted results provided superior predictive performance (MAPE = 0.0087), even though the average MAPE across all simulations (0.0124) was higher than Nguyen's method (0.0113). Figure 1 shows that a majority of our MAPEs were near the ballpark of Nguyen's 0.0113 benchmark, which some simulations outright beating Nguyen's method and others providing very poor overall predictions. Table 1 shows a comparison of MAPEs. Here the Naive method simply uses the previous day return as a prediction for the next day return. The asterisk * in the table indicates MAPEs as reported in Nguyen's paper.

Prediction MAPE	Avg MAPE	Median MAPE	Ngyuen MAPE*	Naive MAPE*
0.0087	0.0124	0.0113	0.0113	0.0133

Table 1: Mean Absolute Prediction Errors

From Figure 2 we can see that our averaged predicted price is closely aligns with the true AAPL stock price, esepcially from Aug 2016 through June 2017. After that, there is a

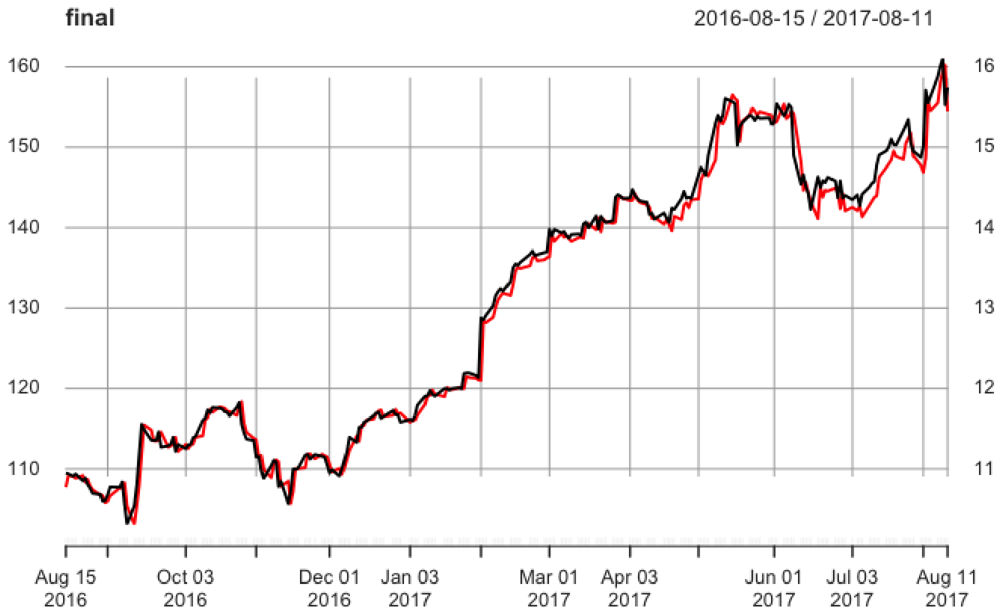

Figure 2: Predicted vs Actual AAPL Stock Price from August 15, 2016 to August 11, 2017

period of time where the predicted price consistently underestimates the true AAPL stock price which is a cause of slight concern. However, when considered as a whole, our method seems to provide promising results in predicting the stock price of AAPL.

5 Concluding Remarks and Next Steps

What we have shown in this project, is an alternative prediction method that uses HMMs to model various time blocks of AAPL stock's open, low, high, and close returns. We use the Bhattacharyya distance as a metric to match current time blocks with historical time blocks. We find that our particular simulations, when averaging the predicted prices, produces superior predictive results when compared to Nguyen's method. While we shown via simulation, that our predictive method provides performance gains, more research needs to be done to show the statistical validity of this method.

Aside from this, there are numerous exciting potential next steps for this project. The historical time range that we used as reference (January 16, 2015 to October 30, 2015) was completely arbritrary aside from the fact that Ngyuen used this time block to calibrate her model parameters in estimating the optimal number of hidden states. If we can come up with a metric that chooses the optimal historical time range to use as reference, we could get significant predictive performance gains. Secondly, Nguyen mentions that using time blocks of $T \geq 80$ worked well with her particular model. More simulation studies to determine the optimal value of T should be done.

Increasing the number of simulations and taking the average of the predicted results is another potential next step. For the sake of the time frame of this project and limitation in computing power, we only run 30 simulations, however we saw that even though some predictions produced bad MAPEs, averaging the predictions provided performance gains. Averaging predictions from various models is a commonly used method in machine learning, and we feel that running more simulations may indeed further our prediction accuracy. Applying this method to a wider range of stocks of varying volatility in different sectors can also be done to test how robust our prediction method is to price volatility and how well it generalizes to common stocks in general.

References

[1] Guy Barrett Coleman and Harry C Andrews. Image segmentation by clustering. *Proceedings of the IEEE*, 67(5):773–785, 1979.

[2] Md Rafiul Hassan and Baikunth Nath. Stock market forecasting using hidden markov model: a new approach. In *Intelligent Systems Design and Applications, 2005. ISDA'05. Proceedings. 5th International Conference on*, pages 192–196. IEEE, 2005.

[3] Nguyet Nguyen. An analysis and implementation of the hidden markov model to technology stock prediction. *Risks*, 5(4):62, 2017.

[4] Brian Ripley, William Venables, and Maintainer Brian Ripley. Package nnet. *R package version*, pages 7–3, 2016.

[5] Ingmar Visser, Maarten Speekenbrink, et al. depmixs4: an r package for hidden markov models. *Journal of Statistical Software*, 36(7):1–21, 2010.

[6] Narada Warakagoda. *Assumptions in the theory of HMMs*, 1996 (accessed June 10, 2018).