Stock Market Prediction Using Hidden Markov Models

Aditya Gupta, *Non-Student Member, IEEE* and Bhuwan Dhingra, *Non-Student member, IEEE*

***Abstract*-- Stock market prediction is a classic problem which has been analyzed extensively using tools and techniques of Machine Learning. Interesting properties which make this modeling non-trivial is the time dependence, volatility and other similar complex dependencies of this problem. To incorporate these, Hidden Markov Models (HMM's) have recently been applied to forecast and predict the stock market. We present the Maximum a Posteriori HMM approach for forecasting stock values for the next day given historical data. In our approach, we consider the fractional change in Stock value and the intra-day high and low values of the stock to train the continuous HMM. This HMM is then used to make a Maximum a Posteriori decision over all the possible stock values for the next day. We test our approach on several stocks, and compare the performance to some of the existing methods using HMMs and Artificial Neural Networks using Mean Absolute Percentage Error (MAPE).**

***Index Terms*--Forecasting, Hidden Markov models, Maximum a posteriori estimation, Stock markets.**

I. Introduction

The stock market is a network which provides a platform for almost all major economic transactions in the world at a dynamic rate called the stock value which is based on market equilibrium. Predicting this stock value offers enormous arbitrage profit opportunities which are a huge motivation for research in this area. Knowledge of a stock value beforehand by even a fraction of a second can result in high profits. Similarly, a probabilistically correct prediction can be extremely profitable in the amortized case. This attractiveness of finding a solution has prompted researchers, in both industry and academia to find a way past the problems like volatility, seasonality and dependence on time, economies and rest of the market. Previously, techniques of Artificial Intelligence and Machine Learning - like Artificial Neural Networks, Fuzzy Logic and Support Vector Machines, have been used to solve these problems.

Recently, the Hidden Markov Model (HMM) approach was applied to this problem in [9]. The reason for using this approach is fairly intuitive. HMM's have been successful in analyzing and predicting time depending phenomena, or time series. They have been used extensively in the past in speech recognition, ECG analysis etc. The stock market prediction problem is similar in its inherent relation with time. Hidden Markov Models are based on a set of unobserved underlying states amongst which transitions can occur and each state is associated with a set of possible observations. The stock market can also be seen in a similar manner. The underlying states, which determine the behavior of the stock value, are usually invisible to the investor. The transitions between these underlying states are based on company policy, decisions and economic conditions etc. The visible effect which reflects these is the value of the stock. Clearly, the HMM conforms well to this real life scenario.

The choice of attributes, or feature selection is significant in this approach. In the past various attempts have been made using the volume of trade, the momentum of the stock, correlation with the market, the volatility of the stock etc. In our model we use the daily fractional change in the stock value, and the fractional deviation of intra-day high and low. The fractional change is necessary in order to make the required prediction. Measuring the fractional deviation of both the intra-day high and low value is a good measure as it gives the direction of the volatility as well.

We use four different stocks for evaluating the approach - TATA Steel, Apple Inc., IBM Corporation and Dell Inc. A separate HMM is trained for each stock. The one constraint that the training set needs to have is suitable variability in the data. This is taken care of by taking appropriately large periods of time in which the stock value changes steadily yet significantly.

The remaining paper is organized as follows. In Section II we review some of the existing techniques for stock market prediction, especially the ones using HMMs. In Section III we give details of our approach with mathematical justifications. In Section IV we describe the data-sets and provide the experimental results. Finally, in Section V we discuss the results and conclude the paper.

II. Previous Work

Stock Market prediction has been one of the more active research areas in the past, given the obvious interest of a lot of major companies. In this research several machine learning techniques have been applied to varying degrees of success. However, stock forecasting is still severely limited due to its non-stationary, seasonal and in general unpredictable nature.

A. Gupta is with the Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur-208016, India (e-mail: gaditya@iitk.ac.in).

B. Dhingra is with the Department of Electrical Engineering, Indian Institute of Technology, Kanpur-208016, India (e-mail: bhuwand@iitk.ac.in).

978-1-4673-0455-9/12/$31.00 ©2012 IEEE

Predicting forecasts from just the previous stock data is an even more challenging task since it ignores several outlying factors (such as the state of the company, economic conditions ownership etc.).

Machine learning techniques which have been widely applied to forecasting stock market data include Artificial Neural Networks (ANNs) [3], Fuzzy Logic (FL) [2], and Support Vector Machines (SVMs) [7]. Out of these ANNs have been the most successful, however even their performance is quite limited, and not reliable enough [4]. Wang and Leu trained a system based on a recurrent neural network which used features extracted using the Autoregressive Integrated Moving Average (ARIMA) analysis, which showed reasonable accuracy, [8].

HMMs have only been rarely applied to the given problem in the past, which is surprising given the time-dependent nature of the market. HMMs have been successfully applied in the areas of Speech Recognition, DNA sequencing, and ECG analysis (see [6] for a full review). Shi and Weigend, [9], used HMMs to predict changes in the trajectories of financial time series data. Recently Hassal combined HMMs and fuzzy logic rules to improve the prediction accuracy on non-stationary stock data sets, [4]. His performance was significantly better than that of past approaches. The basic idea there was to combine HMM's data pattern identification method to partition the data-space with the generation of fuzzy logic for the prediction of multivariate financial time series data.

Our approach is similar to the one taken by Nobakht et al in [5]. They model the daily opening, closing, high and low indices as continuous observations from underlying hidden states. The main difference between our approach and theirs lies in the features used (we use fractional changes in the above quantities) and in the manner of forecasting. While they look for similar data patterns in the past data, we maximize the likelihood of a sequence of observations over all possible forecasted future values.

III. Approach

We use a continuous Hidden Markov Model (CHMM) to model the stock data as a time series. An HMM (denoted by λ) can be written as

$$\lambda = (\pi, A, B) \tag{1}$$

Where A is the transition matrix whose elements give the probability of a transition from one state to another, B is the emission matrix giving $b_j(O_t)$ the probability of observing O_t when in state j, and π gives the initial probabilities of the states at $t = 1$. Further, for a continuous HMM the emission probabilities are modelled as Gaussian Mixture Models (GMMs):

$$b_j(\overrightarrow{O_t}) = \sum_{m=1}^{M} c_{jm}\, N(\overrightarrow{O_t}, \overrightarrow{\mu_{jm}}, \Sigma_{jm}) \tag{2}$$

where:

- M is the number of Gaussian Mixture components.
- c_{jm} is the weight of the mth mixture component in state j.
- $\overrightarrow{\mu_{jm}}$ is the mean vector for the mth component in the jth state.
- $N(\overrightarrow{O_t}, \overrightarrow{\mu_{jm}}, \Sigma_{jm})$ is the probability of observing $\overrightarrow{O_t}$ in the multi-dimensional Gaussian distribution.

Training of the above HMM from given sequences of observations is done using the Baum-Welch algorithm which uses Expectation-Maximization (EM) to arrive at the optimal parameters for the HMM, [1].

In our model the observations are the daily stock data in the form of the 4-dimensional vector,

$$O_t = \left(\frac{close - open}{open}, \frac{high - open}{open}, \frac{open - low}{open}\right) \tag{3}$$
$$:= (fracChange, fracHigh, fracLow)$$

Here open is the day opening value, close is the day closing value, high is the day high, and low is the day low. We use fractional changes along to model the variation in stock data which remains constant over the years.

Once the model is trained, testing is done using an approximate Maximum a Posteriori (MAP) approach. We assume a latency of d days while forecasting future stock values. Hence, the problem becomes as follows - given the HMM model λ and the stock values for d days $(O_1, O_2, \dots, O_d)$ along with the stock open value for the $(d+1)^{st}$ day, we need to compute the close value for the $(d+1)^{st}$ day. This is equivalent to estimating the fractional change $\frac{close-open}{open}$ for the $(d+1)^{st}$ day. For this, we compute the MAP estimate of the observation vector O_{d+1}.

Let $\hat{O}_{d+1}$ be the MAP estimate of the observation on the $d+1^{st}$ day, given the values of the first d days. Then,

$$\hat{O}_{d+1} = arg \max_{O_{d+1}} P(O_{d+1} | O_1, O_2, \dots, O_d, \lambda) \tag{4}$$

$$= arg \max_{O_{d+1}} \frac{P(O_1, O_2, \dots, O_d, O_{d+1} | \lambda)}{P(O_1, O_2, \dots, O_d, \lambda)} \tag{5}$$

The observation vector O_{d+1} is varied over all possible values. Since the denominator is constant with respect to O_{d+1}, the MAP estimate becomes,

$$= arg \max_{O_{d+1}} P(O_1, O_2, \dots, O_d, O_{d+1} | \lambda) \tag{6}$$

The joint probability value $P(O_1, O_2, \dots, O_d, O_{d+1} | \lambda)$ can be computed using the forward-backward algorithm for HMMs (see [6]). In practice, we compute the probability over a discrete set of possible values of O_{d+1}, see Table II, and find the maximum, hence the name MAP HMM model.

The computational complexity of the forward-backward algorithm for finding the likelihood of a given observation is $O(n^2 d)$, where n is the number of states in the HMM and d is the latency. This procedure is repeated over the discrete set of possible values of O_{d+1}. In our case $n = 4$, $d = 10$ (see IV-B) and there are $50 \times 10 \times 10$ possible values of O_{d+1} (see Table II).

The closing value of a particular day can be computed by using the day opening value and the predicted fractional change for that day.

TABLE I
TRAINING AND TEST DATASETS

Observation	Training Data		Test Data	
	From	Till	From	Till
Tata Steel	12 August 2002	4 November 2009	5 November 2009	23 September 2011
Apple Inc.	10 February 2003	10 September 2004	13 September 2004	21 January 2005
IBM Corp.	10 February 2003	10 September 2004	13 September 2004	21 January 2005
Dell Inc.	10 February 2003	10 September 2004	13 September 2004	21 January 2005

IV. SIMULATION AND RESULTS

A. Datasets

The suggested algorithm was tested on four different stock indices - TATA steel, Apple Inc., IBM Corporation and Dell Inc. Details of the training and testing periods is given in Table I.

B. Implementation

We use the open-source HMM toolbox developed by Kevin Murphy, see [10], to implement the various HMM functions.

1) HMM Parameters: The HMM parameters are set to the following values:

- Number of underlying Hidden States (n), $n = 4$
- Number of mixture components for each state (m), $m = 5$
- Dimension of observations (D), $D = 3$
- Ergodic HMM (all transitions are possible).
- Latency (d), d = 10days.

These were obtained by varying the parameters over suitable ranges and choosing the values which give the minimum error between forecasted and actual stock values. Further, [4] suggests the use of four underlying states since the data dimensionality is also four.

2) Initialization: For initialization of the model parameters (see eqn. (1)) the prior probabilities π and transition probabilities A are assumed to be uniform across all states. To initialize the mean, variance and weights of the Gaussian mixture components we use a *k-means* algorithm. Each cluster found from *k-means* is assumed to be a separate mixture component from which the mean and variance are computed. Weights of the components are assumed to be weights of the clusters, which are divided equally between the states to obtain the initial emission probabilities.

3) Prediction: To compute the MAP estimate $\hat{O}_{d+1}$, we compute the probability values over a range of possible values of the tuple $(\frac{close-open}{open}, \frac{high-open}{open}, \frac{open-low}{open})$ and find the maximum. A higher precision is used for the fractional change values since these are ultimately used for the stock prediction. The range of values is listed in Table II.

4) Results: The metric used to evaluate the performance of the algorithm is Mean Absolute Percentage Error (MAPE) in accuracy. MAPE is the average absolute error

TABLE II
RANGE OF VALUES FOR MAP ESTIMATION

Observation	Range		Number of Points
	Min.	Max.	
$\frac{close-open}{open}$	-0.1	0.1	50
$\frac{high-open}{open}$	0	0.1	10
$\frac{open-low}{open}$	0	0.1	10

TABLE III
MEAN ABSOLUTE PERCENTAGE ERROR FOR DIFFERENT APPROACHES

Stock Name	MAP HMM Model	Combination of HMM Fuzzy Model	ARIMA	ANN
Tata Steel	**1.560**	--	--	--
Apple Inc.	1.510	1.769	1.801	1.801
IBM Corp.	0.611	0.779	0.972	0.972
Dell Inc.	0.824	0.405	0.660	0.660

between the actual stock values and the predicted stock values in percentage.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|p_i - a_i|}{|a_i|}\times 100\% \qquad (7)$$

Where a_i is the actual stock value, p_i is the predicted stock value on day i and n is the number of days for which the data is tested. Table III lists the MAPE values for the four stocks using our developed algorithm. For comparison, the corresponding MAPE values using the HMM-Fuzzy logic model, ARIMA and ANN are also listed (obtained from [4]).

Figures 1 and 2 show the actual stock values along with the forecasted values using the discrete MAP-HMM model for TATA Steel and Apple Inc. respectively.

V. CONCLUSIONS

We present an HMM based MAP estimator for stock prediction. The model uses a latency of d days to predict the stock value for the $(d + 1)^{st}$ day. A MAP decision is made over all the possible values of the stock using a previously trained continuous-HMM. We assume four underlying hidden states which emit the visible observations (fractional change, fractional high, fractional low). Emission probabilities

conditioned on a given state are modeled as Gaussian Mixture Models (GMMs). The model can be easily extended to predict stock values for more than one day in the future; however the accuracy of such predictions would decrease as we look further into the future.

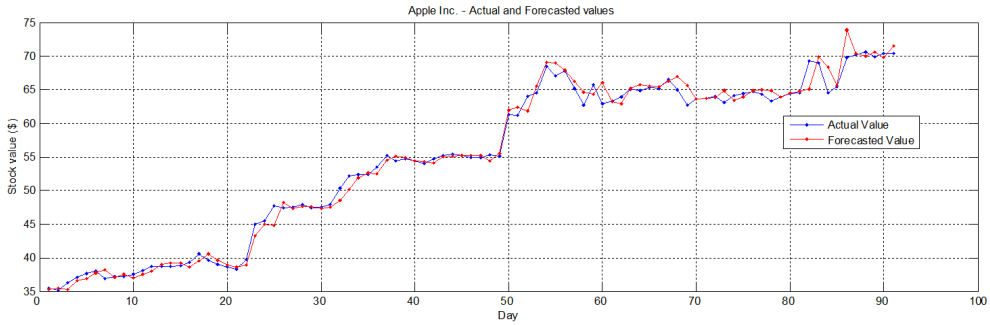

Fig. 1. Actual Value v/s Forecasted Value for Apple Inc. from 13 September 2004 to 21 January 2005.

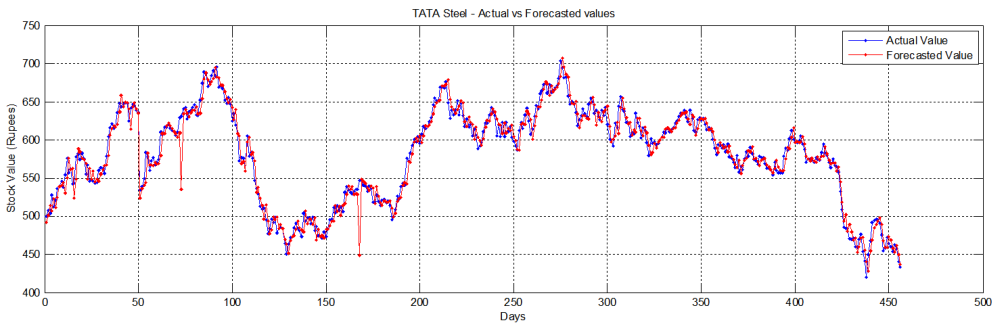

Fig. 2. Actual Value v/s Forecasted Value for Tata Steel from 5 November 2009 to 23 September 2011.

Performance of the suggested algorithm was tested by training HMMs on four different stocks over varying periods of time. The model for one stock was assumed to be independent of other stocks. Our approach was compared to HMM-fuzzy model, ARIMA and ANN for stock forecasting. Table III lists the MAPE values for the four stocks using the various approaches. The MAP-HMM model outperforms others for Apple Inc. and IBM Corporation, and has comparable performance for Dell Inc. Interesting to note is the simplicity of the approach in comparison to other existing methods.

In the current approach, we assume that the model for one particular stock is independent of the other stocks in the market, however in reality these stocks are heavily correlated to each other and, to some extent, to stocks in other markets too. As a future work, it might be intuitive to try and build a model which takes into consideration these correlations. Also, currently the data is quantized to form observation vectors for a full day. A performance improvement might be achieved by removing this quantization and instead taking the full range of minute-by-minute or hour-by-hour stock values.

VI. Acknowledgment

The authors would like to thank Dr. Krithika Venkataramani for her guidance and assistance throughout the project.

VII. References

[1] "Hidden markov models and the baum-welch algorithm". *IEEE Information theory society newsletter*, Dec 2003.

[2] L. Cao and F.E.H. Tay. "Financial forecasting using support vector machines". *Neural Computation and Application*, pages 184–192, 2007.

[3] J.H. Choi, M.K. Lee, and M.W. Rhee. "Trading s&p500 stock index futures using a neural network". *Proc of the Third Annual International Conference on Artificial Intelligence Applications on Wall Street*, pages 63–72, 1995.

[4] M.R. Hassan. "A combination of hidden markov model and fuzzy model for stock market forecasting". *Journal of Neurocomputing*, pages 3439–3446, 2009.

[5] B. Nobakht, C.E.J. Dippel, and B. Loni. "Stock market analysis and prediction using hidden markov models", unpublished.

[6] L.R. Rabiner. "A tutorial on hidden markov models and selected applications in speech recognition". *Proceedings of the IEEE*, pages 257–286, 1989.

[7] E. Vercher, J.D. Bermudez, and J.V. Segura. "Fuzzy portfolio optimization under downside risk measures". *Fuzzy Sets and Systems*, pages 769–782, 2007.

[8] J.H. Wang and J.Y. Leu. "Stock market trend prediction using arima-based neural networks". *IEEE Int. Conf. on Neural Networks*, 4:2160–2165, 1996.

[9] A.S. Weigend and S. Shi. "Predicting daily probability distributions of s&p500 returns". *Journal of Forecasting*, pages 375–392, 2000.

[10] K. Murphy. "HMM Toolbox for MATLAB". Internet: http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html, [Oct. 29, 2011].