Predictive stocks trading using Hidden Markov Model (HMM)

Hieu Trung Le (Eric Le)
Computer Science Department
Stanford University
leric@stanford.edu

Abstract

Hidden Markov Models (HMM) is a statistical signal prediction model, which is widely used to predict economic regimes and stock prices. In this paper, I present the application of HMM to predict stocks price using S&P 500 index, Google Inc., Apple Inc., Qualcomm Inc., and Comcast Corp as examples based on the historical stock prices and future stock price predictions

1. Introduction

All actors in the financial market strive towards one thing: to earn risk-adjusted excess rates of return. One might argue that it is impossible to beat the market, as financial markets are efficient and the price of an asset or security always fully reflects the available information. This hypothesis has been object for harsh questioning. Researchers have shown that financial markets are not always efficient and actors in the market can try to exploit such inefficiencies in order to gain risk-adjusted excess rates of return. One such way is trying to predict trends and future prices of financial assets on the market

Financial markets are one of the most complex systems which are almost impossible to model in terms of dynamical equations. Quantitative traders (or quant trader) often wish to exploit inefficiencies in the financial market and make profit by buying a stock at a low price and sell them at later times at a higher price. However, frequent behaviour modification of financial markets often makes this challenging. This modification often happens abruptly,due to changing periods of government policy, regulatory environment, supply and demand between investor and other macroeconomic effects presents a consistent challenge for quantitative traders (or quant trader) to predict trends and future prices of financial assets on the market. The challenging question they often ask is: "when is the best time to buy or sell a stock".

Stock investments can have a huge return or a significant loss due to the high volatilities of the stock prices. Many models were used to predict stock executions such as the "exponential moving average" (EMA) and the "head and shoulders" methods. However, these forecast models require a stationary of the input time series. In reality, financial time series are often nonstationary, thus nonstationary time series models are needed.

Prediction of stock prices is classical problem of non-stationary pattern recognition in Machine Learning. There has been a lot of research in predicting the behavior of stocks based on their historical performance using Artificial Intelligence and Machine Learning techniques like- Artificial Neural Networks, Fuzzy logic and Support Vector Regression (SVR)

One of the methods which is not as common as the above mentioned that effectively detect and categorize these regimes in order to optimally select trading strategies that can optimize the return of the market portfolio. Such principal method for carrying out regime detection is known as a Hidden Markov Models.

Hidden Markov Models (or HMM) have a strong probabilistic framework for recognizing patterns in stochastic processes. The advantage of HMM can be summarized as:

- strong statistical foundation and probabilistic framework
- able to handle new data robustly
- computationally efficient to develop and evaluate(due to the existence of established training algorithms)
- able to predict similar patterns efficiently

They have been used for analyzing patterns in speech, handwriting and gestures and are still extensively used in those areas. HMM also found success in analyzing wide variety of DNA sequences. The underlying idea behind HMM is that the likelihood of the observations depend on the states of the system which are 'hidden' to the observer. HMM involve inference generative processes on these "hidden" states via "noisy" indirect observations correlated to these processes. To explain it in a different way, the transition from one state to another is modeled as Markov Process in Hidden Markov Models. The next state as result of Markov Process always depends only on the present state, hence the name Hidden Markov Models. States in HMMs are always discrete, while the observations can either be discrete or continuous or both.

Stock markets can be viewed and modeled as Hidden Markov Models. Investor can only observe the stock prices today and the underlying states which are driving the stock prices are unknown. The transition of stock prices today to stock price tomorrow is modeled as the Markov Process. The observations that we are considering are multiple independent variables: daily prices open, close, high and low. In this project, we will consider our observations to be distributed as multivariate Gaussian distribution. In this project, HMM will be used to apply to various assets to detect regimes and is used to come up with a set of quantitative trading strategies to trade S&P 500 index in the market.

2. Related Works

The mathematical foundations of HMMs were developed by Baum and Petrie [2] in 1966. In that paper, they proved the maximum likelihood estimate converge to the correct value. Four years latter, in 1970, Baum et. al. [8] published a maximization method in which the parameters of HMMs are calibrated using a single observation.

In 1983, Levinson, Rabiner and Sondhi [11] introduced a maximum likelihood estimation method for HMMs with multiple observation training, assuming that all the observations are independent. In 2000, Li et. al. [13] presented an HMM training for multiple observations without the assumption of independence of observations.

In 2005, Hassan and Nath [7] applied HMM to forecast some of the airline stocks. The trained HMM is used to search for the variable of interest behavioural data pattern from the past data set. The pattern from past datasets that match with today's stock price behaviour are located. These two datasets are then interpolated with appropriate neighbouring price elements to prepare forecasts and and predict the stocks of the airlines

Nguyen[9] extend the work of Hassan and Nath [7] and used statistical signal prediction model to predict economic regimes and statistical prices. Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Hannan Quinn Information Criterion (HQIC) and Bozdogen Consistent Akaike Information (BCAI) are used to test the performances of HMM with two to six states in order to determine an optimal number of states for HMM. Nguyen[9] then use the selected HMM model and multiple observations (open, close, high, low prices) to predict the closing price of the S&P 500. Many statistic methods are also used to evaluate the HMM out-of-sample predictions over the results obtained by the benchmark the HAR model. Nguyen[9] went further by using the models to trade the S&P 500 using different training and predicting period. The analysis clearly proves that HMM outperforms from traditional method in predicting and trading stocks. I will use AIC, BIC, HQIC, BCAI in this paper to also determine an optimal number of states for HMM.

Price	Min	Max	Mean	Std.
Open	16.66	2173.15	504.06	582.20
High	17.09	2213.35	520.46	599.28
Low	16.66	2147.56	486.91	562.96
Adj Close	17.05	2213.35	506.73	584

Table 1. S&P 500 Dataset

Gupta and Bhuwan [6] used fixed state HMM based Mean Absolute Percent (MAP) estimator to maximize the likelihood of observation of all probable sequences. They applied HMM to forecast and predict the stock market prices for one day by providing the past data. Maximum posterior HMM is used for the prediction. Later Artificial Neural network is also used for the stock market prediction. Then the obtained results of both the models are evaluated using Mean Absolute Percent Error (MAPE)

3. Dataset

I will use one of the common benchmarks for the U.S. stock market, the Standard & Poor's 500 index (S&P 500) as input to my model. Additonally, I will also use hirtorical data of Google Inc., Apple Inc., Qualcomm Inc., and Comcast Corp. Each has monthly data from January 2000 to December 2019 and was taken from `finance.yahoo.com`. There is about 5,031 rows of data over the course of 19 years which include daily open price, daily high price, daily low price and adjusted close price. The summary of the data is presented in Table 1

- Daily Open Price: the price of the stock at the beginning of the trading day (it need not be the closing price of the previous trading day).
- Daily High Price: the highest price of the stock on that trading day.
- Daily Low Price: the lowest price of the stock on that trading day, and close the price of the stock at closing time.
- Adjusted close Price : the closing price of the stock that adjusts the price of the stock for corporate actions.

4. Baseline

I will use the SVR which is similar model to SVM [5] as the baseline. SVM stimates the regression using a set of linear functions that are defined in a high-dimensional feature space.

SVM carries out the regression estimation by risk minimization, where the risk is measured using loss function. Given a set of data points $(x_1, y_1), (x_2, y_2), ..., (x_l, y_l)$, l is the total number of training samples) randomly and independently generated from an unknown function, SVM approximates the function using the following form:

$$f(x) = w\phi(x) + b$$

where $\phi(x)$ represents the high-dimensional feature spaces which is nonlinearly mapped from the input space x. w and b are estimated by minimizing:

$$\text{minimize } \frac{1}{2}\|w\|^2 + C\frac{1}{l}\sum_{i=1}^{l} L_\varepsilon(y_i, f(x_i))$$

$$L_\varepsilon(y_i, f(x_i)) = \begin{cases} |y_i - f(x_i)| - \varepsilon, & |y_i - f(x_i)| \geq \varepsilon \\ 0, & \text{otherwise.} \end{cases}$$

Unlike SVM for classification where y_i is a label, in SVR y_i is a response variable. Using SVR, I tried to find a function which takes in the current day's prices and predicts the next day's close prices. Training and testing is exactly same as I did for HMM. I start predictions with 100th day and use its true observation to re-train the SVR to predict for 99th day and so on. To predict the next day's prices, I trained the SVR with the data from the past till last day. I then passed the current day's prices to get the predictions for the next day. I calculated the MAPE and plotted the predictions on the same plots of HMM to compare the results from SVR and HMM.

Besise SVR, I will also implement additional baseline models using time series forecasting methods: Auto Regressive Integrated Moving Average (or ARIMA) models and Error Trend and Seasonality, or exponential smoothing (or ETS) models. These technique are often used for predicting future aspects of data, in which we translate past data into estimates of future data. These technique are commonly used in business, as companies need to account for the uncertainty of the future, and being able to forecast data over time offers them a way to prepare for this.

There are four general components of time series forecasting model:

- Trend component: the direction in which the data is trending over time
- Cyclical component: the observance of the data deviating over time due to fluctuation
- Seasonal component: captures the variability of the data over time, also due to fluctuation
- Irregular component: captures the random variation in data that cannot be predicted in advance and are generally caused by short-term, unanticipated and nonrecurring factors that occur over time

This can be described as:

$$Y(t) = S(t) + T(t) + C(t) + R(t)$$

where S(t) is the seasonal component, T(t) is the trend-cycle component, C(t) is the cyclical component and R(t) is the remainder component. There exists several techniques to estimate such a decomposition. The most basic one is called classical decomposition that requires

1. Estimating trend T(t) through a rolling mean
2. Computing S(t) as the average detrended series Y(t)-T(t) for each season (e.g. for each month)
3. Computing C(t) as the cyclical component for each season
4. Computing the remainder series as R(t)=Y(t)-T(t)-S(t)

There are several differences between ARIMA and ETS models. ARIMA models are 1)stationary 2) do not have exponential smoothing counterparts and 3) suitable if there is autocorrelation in the data (ie the past data explains the present data well). ETS models, on the other hand, are 1) are not stationary 2) use exponential smoothing and 3) suitable if there is a trend and/or seasonality in the data. To test the effectiveness of our model, we will implement both ARIMA and ETS and compare their performance with our model's performance

5. Main approach

The Hidden Markov Model is a generative probabilistic model in which the system is considered to be transitioning in certain number of states. The state transition is a Markov Process and hence can be defined by a matrix of state transition probabilities. It was introduced by Baum and Petrie [2]. The model has the following main assumptions:

- an observation at t was generated by a hidden state (or regime),
- the hidden states are finite and satisfy the first-order Markov property
- the matrix of transition probabilities between these states is constant
- the observation at time t of an HMM has a certain probability distribution corresponding with a possible hidden state

There are two main categories of the hidden Markov model: a discrete HMM and a continuous HMM. In this paper I will use the discrete HMM to model my prediction. The summary of discrete HMM is summarized in table 2 in [9]

The parameters of an HMM are the constant matrix A, the observation probability matrix B and the vector p, which is summarized in a compact notation $\lambda = \{A, B, p\}$

The main questions when applying HMM are:

1. Given the observation data $O = \{O_t, t = 1, 2, ..., T\}$ and the model parameters $\lambda = \{A, B, p\}$, calculate the probability of observations $P(O|\lambda)$
2. Given the observation data $O = \{O_t, t = 1, 2, ..., T\}$ and the model parameters $\lambda = \{A, B, p\}$, find the best fit state sequence $\mathcal{Q} = \{q_1, q_2, ..., q_T\}$ of the observation sequence
3. Given the observation sequence $O = \{O_t, t = 1, 2, ..., T\}$, calibrate HMM's parameters $\lambda = \{A, B, p\}$

These questions can be solved by using the algorithm:

1. Find the probability of observations: Forward or backward algorithm [1] [4]
2. Find the "best fit" hidden states of observations: Viterbi algorithm [12]
3. Calibrate parameters for the model: Baum–Welch algorithm by [3]

The full algorithm is found in [10].

Hidden Markov Models have been a powerful tool for analyzing non-stationary systems. Stock Markets are non-stationary systems and the observations are continuous in nature. Consider O_t be a vector of historical observed data of a fixed length T, $O = \{O_t^{(1)}, O_t^{(2)}, O_t^{(3)}, O_t^{(4)}\}$ where $O^{(i)}$ with i = 1, 2, 3, or 4 represents the open, low, high or adjusted closing price of a stock. Let S_t to be the state on day t. Since the vector O_t takes real values, observations can be modelled as Multivariate Gaussian distributed. Let the state S_t be the state on day t. The observations are assumed to independent whereas the elements of an observation may be correlated. The state S_t can take only discrete values since HMM is a finite state machine. Figure 1 shows a typical Hidden Markov Process

Let's demonstrate an example. I will use HMMs to predict the daily stock prices. I will predict one year daily stock prices by fixing the length of the "moving window" to 252 days. First, HMM's parameters are calibrated using training data and the probability of the observation of the data

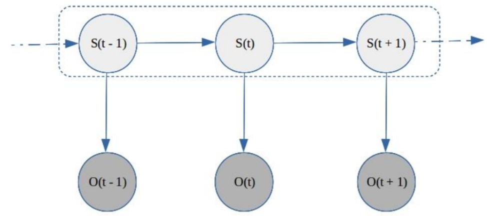


Figure 1. HMM architecture

set is calculated. Then to answer question: "How to use the Hidden Markov Model?", we need to answer these three questions:

- Given the model, how likely it is to observe the given sequence of stock data in a given day?
- Given the model and observations, what is the best hidden state sequence to model the observations of the stock data?
- Given the observations, what are the optimal model parameters for our predictive stock model?

For the first question, we will use Forward algorithm. The second question can be solved by Viterbi algorithm. The third problem can be answered by Baum-Welch algorithm .

The main idea for predicting the next day's stock price is to calculate the log-likelihood of K previous observations and comparing it with the log-likelihood of all the previous sub-sequences of same size by shifting the window by one day in the direction of past data. We then identify a day in the past whose log- likelihood of its K previous observations is the closest to the sub-sequence whose next day's price is to be predicted.

$$j = argmin_i(|P(O_t, O_{t-1}, O_{t-2}, ..., O_{t-K}|\lambda) \\ -P(O_{t-i}, O_{t-i-1}, O_{t-i-2}, ..., O_{t-i-K}|\lambda)|)$$

where $i = 1, 2, ..., T/K$. We then calculate the differential price change from the identified day to its next day. This change is then added to the current day's price to get our next day's prediction.

$$O_t = O_t + (O_{t-j+1} - O_{t-j}$$

Subsequently, after we get the true observation, we include it to our dataset and retune our model parameters in order to ensure that our model doesn't diverge. In short, we fix the size of our sub-sequence and locate another subsequence from the past data which exhibits a similar pattern.

We then map the behavior of the identified sub-sequence to the sub- sequence being used for prediction.

In order to select a model with optimal number of states, we train a set of models by varying the number of states (N) from the state space G. We have considered the values of number of states in G ranging from [2,25]. We then calculate the negative log- likelihood of the training data used for each of the models and chose the model which has the lowest value. However, this approach tends to prefer a complex model implying that the number of states chosen may tend to a higher value and might result in overfitting. In order avoid this problem, we add a penalty term to the negative log likelihood. Depending on the penalty term chosen, we impose restrictions on the model at varying degree.

6. Evaluation Metric

I will use four common criteria: the Akaike information criterion (AIC), the Bayesian information criterion (BIC), the Hannan– Quinn information criterion (HQC), and the Bozdogan Consistent Akaike Information Criterion (CAIC) to evaluate the performances of HMM with different numbers of states. These criteria are suitable for HMM because the EM method was used to maximize the log-likelihood of the model in the Baum–Welch Algorithm.

- $AIC = -2ln(L) + 2k$
- $BIC = -2ln(L) + kln(M)$
- $HQC = -2ln(L) + kln(ln(M))$
- $CAIC = -2ln(L) + k(ln(M) + 1)$,

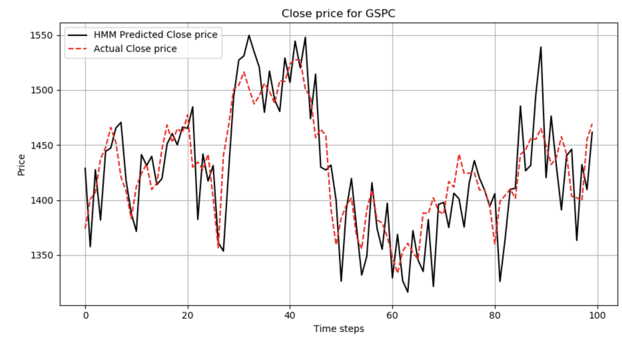
L is the likelihood function for the model, M is the number of observation points, and k is the number of estimated parameters in the model.

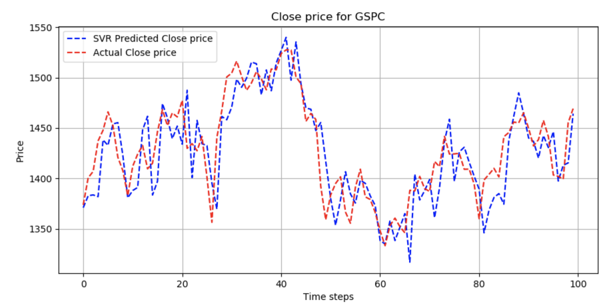
The performance metric that I used in this project is Mean Absolute Percentage Error (MAPE) which is defined as

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} \frac{|Predicted(i) - True(i)|}{True(i)}$$

7. Results & Analysis

In this project, the main objective was to determine the efficiency of HMMs in predicting the stock prices. We used hmmlearn, an open source python library to train the model and calculate the likelihood of the observations. The stocks that we selected are stock index S & P 500 (GSPC), Google Inc., Apple Inc., Qualcomm Inc., and Comcast Corp. We used opening price, closing price, high and low as features for the past 5040 working days (approximately 20 years) when the market was open. We kept aside the recent 100 observations for testing and used rest of the observations for training the model. We predicted the prices for the past 100 days, starting from the 100th day and then using its true observation to retune the model for predicting the 99th day and so on. Therefore, every time we retune the model, the number of training samples will increase by one. First we implemented using fixed model i.e. by fixing the number of states to four. Figure below shows the estimated price vs true price of S&P 500 during 100 days window using HMM, SVR, ARIMA and ETS. We calculated the MAPE and plotted the predictions and the actual prices to compare the results. We then optimized our model by selecting the model with lowest BIC value which is a function of number of states.

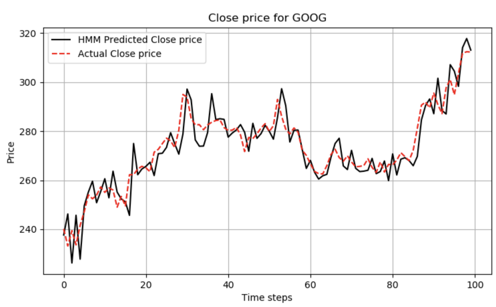


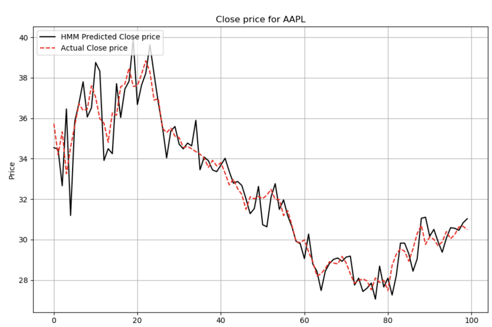


Model	Closing	Open	High	Low
HMM	0.0098	0.0073	0.0067	0.0079
SVR	0.0075	0.0041	0.0053	0.0062
ARIMA	0.0068	0.0038	0.0051	0.0059
ETS	0.0079	0.0047	0.0058	0.0067

Table 2. MAPE for S&P 500

We have observed that the predicted values for Close Price closely follow the trends exhibited by its corresponding true values in both the HMM as well as the SVR, ARIMA and ETS implementations and the MAPE values were found to be similar. The predictions made using the SVR, ETS and ARIMA model was not found to be affected by drastic fluctuations in the stock price. However, the model implemented using HMM was found to be highly sensitive to the fluctuations in stock price. Below is the HMM predicted vs actual stock price for Google Inc., Apple Inc., Qualcomm Inc., and Comcast Corp.





Model	Closing	Open	High	Low
HMM	0.0077	0.0041	0.0052	0.0065
SVR	0.0057	0.0052	0.0061	0.0058
ARIMA	0.0053	0.0057	0.0056	0.0055
ETS	0.0059	0.0047	0.0058	0.0053

Table 3. MAPE for Google

Model	Closing	Open	High	Low
HMM	0.0053	0.0057	0.0044	0.0052
SVR	0.0052	0.0041	0.0053	0.0062
ARIMA	0.0053	0.0044	0.0057	0.0065
ETS	0.0055	0.0047	0.0058	0.0067

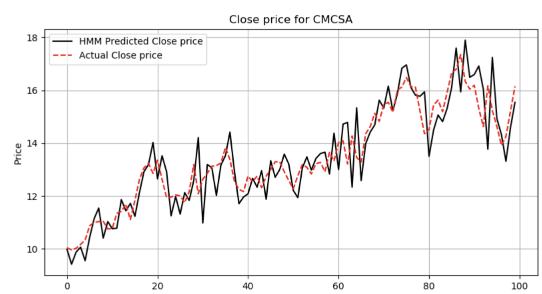
Table 4. MAPE for Apple

Model	Closing	Open	High	Low
HMM	0.0059	0.0070	0.0054	0.00586
SVR	0.0045	0.0041	0.0045	0.0052
ARIMA	0.0041	0.0044	0.0047	0.0055
ETS	0.0043	0.0047	0.0051	0.0060

Table 5. MAPE for Comcast

Model	Closing	Open	High	Low
HMM	0.0124	0.0132	0.0133	0.0122
SVR	0.0098	0.0078	0.0082	0.0106
ARIMA	0.0087	0.0082	0.0084	0.0112
ETS	0.0089	0.0085	0.0088	0.0115

Table 6. MAPE for Qualcomm



8. Conclusion & Future Work

Though in general, the observations will be greatly affected by the choice of the model i.e. the number of states in Hidden Markov Models, it did not make significant difference when we tried to find the optimal states using BIC. Both HMM and SVR give similar accuracy when the next one day prices are predicted. HMM captures the volatility of the stock prices whereas ARIMA, ETS and SVR gives more stable predictions. Therefore, HMM can work better for the stocks with high volatility and ARIMA, ETS and

SVR can work better for stocks which are more stable.

For this project, I wish have more time to analyze larger historical data set (more than 20 years). I also wish I have time to investigate why MAPE number of HHM is generally higher than most baselie methods. Some ideas for future work are:

- Study how HMM can be improved in order to decrease MAPE
- Other observations, indicator that might be helpful in forecasting stock price

9. Code

The code for this project can be found at `https://github.com/leeric92/cs221_finalproj`

References

[1] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov process and to a model for ecology. In *The Annals of Mathematical Statistics 37: 1554–63. Bulletin of the American Mathematical Society 73: 360–63.*, 1967.

[2] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. In *The Annals of Mathematical Statistics 37: 1554–63*, 1966.

[3] L. E. Baum and T. Petrie. Error bounds for convolutional codes and an asymptotically pptimal decoding algorithm. In *IEEE transactions on Information Theory 13: 260–69.*, 1967.

[4] L. E. Baum and G. Sell. Growth functions for transformations on manifolds. In *Pacific Journal of Mathematics 27: 211–27*, 1968.

[5] L. J. Cao and F. E. H. Tay. Support vector machine with adaptive parameters in financial time series forecasting. In *IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 14, NO. 6, NOVEMBER 2003*, 2003.

[6] A. Gupta and B. Dhingra. Stock market prediction using hidden markov models. In *2012 IEEE*, 2012.

[7] N. R. Hassan and B. Nath. Stockmarket forecasting using hidden markov model: A new approach. In *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications*, 2005.

[8] G. S. L.E. Baum, T. Petrie and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. In *The Annals of Mathematical Statistics, 41(1):164–171*, 1970.

[9] N. Nguyen. Hidden markov model for stock trading. In *Int. J. Financial Stud. 2018,*, 2018.

[10] N. T. Nguyen. Probabilistic methods in estimation and prediction of financial models. In *PhD thesis, Florida State University, Tallahassee, FL, USA*, 2014.

[11] L. R. S.E. Levinson and M. Sondhi. An introduction to the application of the theory of probabilistic functions of markov process to automatic speech recognition. In *Bell System Technical Journal, 62(4):1035–1074*, 1983.

[12] A. Viterbi. Error bounds for convolutional codes and an asymptotically pptimal decoding algorithm. In *IEEE transactions on Information Theory 13: 260–69.*, 1967.

[13] M. P. X. Li and R. Plamondon. Training hidden markov models with multiple observations a combinatorial method. In *IEEE Transactions on PAMI, 22(4):371–377*, 2000.