SOCIETY OF ACTUARIES

Hidden Markov Model for Portfolio Management with Mortgage-Backed Securities Exchange-Traded Fund

April 2017

Hidden Markov Model for Portfolio Management with Mortgage-Backed Securities Exchange-Traded Fund

SPONSOR Committee on Finance Research AUTHOR Nguyet (Moon) Nguyen

Caveat and Disclaimer

CONTENTS

Hidden Markov Model for Portfolio Management with Mortgage-Backed Securities Exchange-Traded Fund

ACKNOWLEDGMENTS

The author is grateful to the members of the Project Oversight Group (POG) for their expertise, time and assistance with this project. The POG members included Robert Reitano (chair), Pak (Alan) Chan, Jean-Sebastian Chbani, Steve Craighead, Brian Grossmiller, Doug Norris, Kailan Shang, Pierre Tournier and Steve Siegel (SOA Research Actuary).

ABSTRACT

The hidden Markov model (HMM) is a regime-shift model that assumes observation data were driven by hidden regimes (or states). The model has been used in many fields, such as speech recognition, handwriting recognition, biomathematics and financial economics. In this paper, we describe HMM and its application in finance and actuarial areas. We then develop a new application of HMM in mortgage-backed securities exchange-traded funds (MBS ETFs).

We begin with a primer on the hidden Markov model, covering main concepts, the model's algorithms and examples to demonstrate the concepts. Next, we introduce some applications of the model in actuarial and financial areas. We then present applications of HMM on MBS ETFs. Finally, we establish a new use of HMM for a portfolio management with MBS ETFs: predicting prices and trading some MBS ETFs. Data, algorithms and codes generated in this paper can be used for future research in actuarial science and finance.

SECTION 1: A BRIEF HISTORY OF THE HIDDEN MARKOV MODEL

The hidden Markov model is a signal detection model that assumes observation sequences were derived from a hidden state sequence. This sequence is discrete and satisfies the first order of a Markov process. Baum and Petrie (1966) developed the mathematical foundations of HMM. In their paper, the authors assumed that observations were generated by a hidden sequence, which is simulated by a Markov process. In HMM, these states are invisible, while observations (the inputs of the model), which depend on the states, are visible. An observation at time t of an HMM has a particular probability distribution corresponding to a possible state. The probability was called an observation probability. Baum and Petrie assumed that the transition probability matrix of the S-states Markov process and the observation probability matrix are unknown and proved that we could use the maximum likelihood estimate (MLE) to uncover these parameters of the model. Therefore, the history of HMM is the history of calibrating its parameters.

Since its introduction in 1966, HMM has undergone some developments. The innovations involved solving the third problem above: calibrating the model's parameters. HMM was developed from a model for a single observation sequence to a model for multiple observation sequences. In 1970, Baum and his colleagues published a maximization method in which the parameters of HMM are calibrated using a single observation sequence (Baum L. E., Petrie, Soules, & Weiss, 1970). In the paper, the authors suggested using the model for weather prediction and stock market behavior.

More than a decade later, Levinson, Rabiner and Sondhi (1983) introduced an MLE method for training HMM with multiple observation sequences, assuming that all the observation sequences are independent (Levinson, Rabiner, & Sondhi, 1983). The authors pointed out three issues of the training algorithm, the Baum-Welch algorithm, and presented a modified version of it. Then, they applied a particular class of HMM model, called "left-to-right HMM," for speech recognition.

In 2000, Li, Parizeau and Plamondon presented an HMM training for multiple observation sequences without the assumption of independence of these sequences (Li, Parizeau, & Plamondon, 2000). Li and colleagues also indicated two special cases: independence observation sequences and uniform dependence observation sequences. (Baggenstoss, 2001) introduced a modification of the Baum-Welch algorithm for multiple observation spaces. That same year Ghahramani published a complete tutorial of hidden Markov models and Bayesian Networks (Ghahramani, 2001). The new overview of connections between HMM and Bayesian Networks made the model applicable to multiple-state sequences, multiscale representations and a mixture of discrete and continuous variables. Great details about the history and theory of HMM and its applications in a variety of fields can be found in
(Dymarski, 2011).

The remainder of this paper is organized as follows: We present key concepts, examples and algorithms of the hidden Markov model in Section 2. Section 3 contains the applications of HMM for a specific MBS ETF, the MBB, to clarify the model's algorithms. Section 4 summarizes some applications of the HMM in finance and actuarial sciences. Section 5 displays applications of the HMM in predicting prices and trading some MBS ETFs. The last section wraps up the discussion.

SECTION 2: MAIN CONCEPTS AND EXAMPLES OF HMM

In this section, we will present the main concepts of the hidden Markov model and an example of a discrete HMM.

2.1 Main Concepts of HMM

HMM is a stochastic signal model based on the following assumptions:

1. An observation at t was generated by a hidden state.
2. The hidden states are finite and satisfy the first-order Markov property.
3. The matrix of transition probabilities between these states is constant.
4. The observation at time t of an HMM has a certain probability distribution corresponding with a possible hidden state.

There are two main types of HMM: a discrete HMM and a continuous HMM. The two versions have minor differences, so we will first present key concepts of a discrete HMM. Then, we will add details about a continuous HMM.

Basic elements of a discrete HMM are:

- Length of observation data, T
- Number of states, N
- Number of symbols per state, M
- Observation sequence, $O = \{O_t, 1 \le t \le T\}$
- Hidden state sequence, $Q = \{q_t, 1 \le t \le T\}$
- Possible values of each state, $S = \{S_i, 1 \le i \le N\}$
- Possible symbols per state, $V = \{v_{k,}, 1 \le k \le M\}$
- Transition matrix, $A = (a_{ij})$, where $a_{ij}$is the probability of being in state S_i at time t given that the observation is in state S_i at time $t-1$,
$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i), 1 \le i,j \le N.$$
- Vector of initial probability, $p = (p_i), 1 \le i \le N$, where p_i is the probability of being in state S_i at time $t = 1$,
$$p_i = P(q_1 = S_i), 1 \le i \le N.$$
- Observation probability matrix, $B = (b_{ik})$, where b_{ik} is the probability of being in symbol $v_k$of an observation $O_t$given that the observation is in state S_i,
$$b_{ik} = b_i(k) = P(O_t = v_k | q_t = S_i), 1 \le i \le N, 1 \le k \le M.$$
- Probability of observation $P(O|\lambda)$. This probability is called the likelihood of the HMM.

For convenience, in the algorithms presented in this paper, we use the notation $b_i(O_t)$ to present the probability $b_i(k)$.

Transition matrix$A$, vector of initial probability p and observation probability B must satisfy the probability constraints:

$$a_{ij} > 0, 1 \leq i, j \leq N, \text{ and } \sum_{j=1}^{N} a_{ij} = 1, 1 \leq i \leq N.$$

$$p_i > 0, 1 \leq i \leq N, \text{ and } \sum_{i=1}^{N} p_i = 1.$$

$$b_i(k) > 0, 1 \leq i \leq N, 1 \leq k \leq M, \text{ and} \sum_{k=1}^{M} b_i\,(k) = 1, 1 \leq i \leq N.$$

The parameters of an HMM are the transition probability matrix A, the observation probability matrix B and the initial probability vector p. For convenience, we use a compact notation for the parameters:
(Akaike, 1974)

$$\lambda = \{A, B, p\}.$$

If we have infinite symbols for each hidden state, the symbol v_k will be omitted from the model, and the conditional observation probability $b_i(k)$ is written as

$$b_i(k) = P(O_t | q_t = S_i) = b_i(O_t\,), 1 \leq i \leq N.$$

If the probabilities are continuously distributed, we have a continuous HMM.

If we assume that the conditional observation probability is the Gaussian distribution, then we have

$$b_i(O_t\,) = \mathbf{N}(O_t\,, \mu_i, \sigma_i),$$

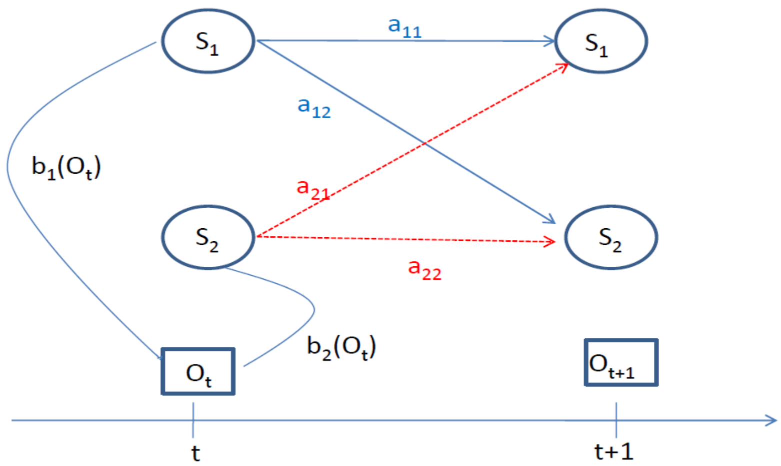
where μ_i andσ_i are the mean and variance, respectively, of the normal distribution corresponding to the state S_i, and $\mathbf{N}(.,.,.)$ is a Gaussian density function.

In this case, the parameters of HMM are

$$\lambda = \{A, \mu, \sigma, p\},$$

where μ and σ are vectors of means and variances, respectively, of the Gaussian distributions. A simple HMM is presented in Figure 1.

Figure 1
SIMPLE HMM



In summary, to define an HMM, first we have to choose a number of states, N, and a number of symbols, M, per state, and specify the symbols. Then, with an observation sequence, $O = \{O_{t,}, 1 \leq t \leq T\}$, we can find the model's parameters $\lambda = \{A, B, p\}$, and use the settings to find the hidden state sequence, $Q = \{q_{t,}, 1 \leq t \leq T\}$ of the observation sequence O.

2.2 Example of a Discrete HMM

We present a simple example of a discrete HMM with two states, $N = 2$, and two symbols for each state, $M = 2$. Suppose that we have two boxes, which represent the two states S_1 and S_2.

$$S = \{S_1, S_2\} = \{Box\ one, Box\ two\}.$$

Box one has three red balls and one blue ball. Box two has two red balls and three blue balls. The *Red* ball and *Blue* ball are the two symbols of each state (or box):

$$V = \{v_1, v_2\} = \{Red, Blue\}.$$

A player will first choose one of the two boxes by using a random process (e.g., flipping a coin) and then select a ball from the chosen box with replacement. The player repeats the process T times and records the selected balls. The sequence of selected balls is the observation data, for example, $O = \{Red, Blue, Blue, Red, \dots\}$. The sequence of selected boxes is the hidden state, for example,

$$Q = \{q_1, q_2, q_3, q_4, \dots\} = \{S_1, S_2, S_2, S_1, \dots\}.$$

We assume that the hidden states of HMM satisfy the first-order Markov chain, which means given a state sequence $\{q_1, q_2, \dots, q_T\}$, the probability of being in state S_i at time $t + 1$ depends only on q_t:

$$P(q_{t+1} = S_i | q_1, q_2, \dots, q_t) = (P(q_{t+1} = S_i | q_t).$$

If the player chooses the boxes (states) by flipping a pair of coins, the transition probability matrix A will be

$$A = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}.$$

Also, the initial probability of choosing the first box (state) is $p = (0.5, 0.5)$. Because we assumed that box one has three red balls and one blue ball, and box two has two red balls and three blue balls, we can calculate the probability of observation matrix B.

If the state S_1 (or box one) was selected at time t:

- The probability of observing a red ball is

$$b_1(1) = P(v_1 | S_1) = P(Red | S_1) = \frac{3}{4} = 0.75$$

- The probability of observing a blue ball is

$$b_1(2) = P(v_2 | S_1) = P(Blue | S_1) = \frac{1}{4} = 0.25$$

If the state S_2 (or box two) was selected at time t:

- The probability of getting a red ball is

$$b_2(1) = P(v_1 | S_2) = P(Red | S_2) = \frac{2}{5} = 0.4$$

- The probability of getting a blue ball is

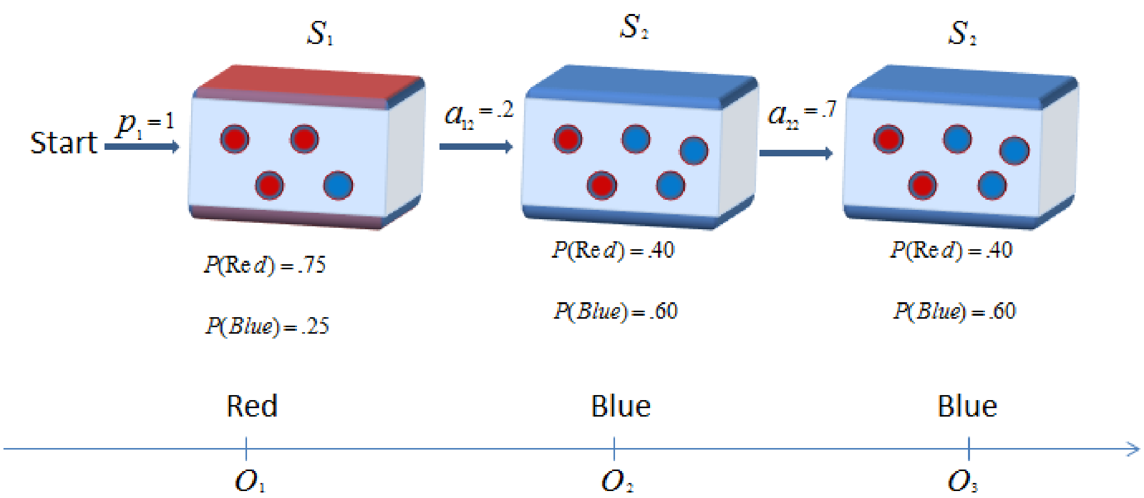
$$b_2(2) = P(v_2 | S_2) = P(Blue | S_2) = \frac{3}{5} = 0.6$$

Thus, we have the observation probability matrix

$$B = \begin{bmatrix} 0.75 & 0.25 \\ 0.40 & 0.60 \end{bmatrix}.$$

Figure 2 is the summary of the simple HMM.

Figure 2
A SIMPLE EXAMPLE OF HMM



Next, we present three main questions that we can ask players when they play the ball game. To make it simple, let's assume that we play the game three times, $T = 3$, and have the observation results $O = \{Red, Blue, Blue\}$. Assume that we choose the state (box) sequence that satisfies the first order of the Markov chain process; box one was picked the first time, $p = (1,0)$, the transition matrix of the state sequence is $A = \begin{bmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{bmatrix}$ and the observation probability matrix is $B = \begin{bmatrix} 0.75 & 0.25 \\ 0.40 & 0.60 \end{bmatrix}$.

Question 1: Suppose that you are given an observation sequence O and the model's parameters $\lambda = \{A, B, p\}$. Can you calculate the probability of observing $P(O|\lambda)$?

We can solve this problem by listing all possible hidden state sequences Q of the observation sequence $O = \{Red, Blue, Blue\}$, calculating the probability $P(O, Q|\lambda)$, and having

$$P(O|\lambda) = \sum_{all\ Q} P(O, Q|\lambda) P(Q\ \ |\lambda).$$

Since the state S_1 is chosen first, we will have four possible state sequences:

$Q_1 = \{S_1, S_1, S_1\}, Q_2 = \{S_1, S_1, S_2\}, Q_3 = \{S_1, S_2, S_1\}, Q_4 = \{S_1, S_2, S_2\}$. (1)

Then,

$$P(O|\lambda) = \sum_{j=1}^{4} P(O|Q_j, \lambda) P(Q_j|\lambda).$$

We have:

$$P(Q_1|\lambda) = P(\{S_1, S_1, S_1\}|\lambda) = p_1 a_{11} a_{11} = 1(0.7)(0.7) = 0.49$$

$$P(Q_2|\lambda) = P(\{S_1, S_1, S_2\}|\lambda) = p_1 a_{11} a_{12} = 1(0.7)(0.3) = 0.21$$

$$P(Q_3|\lambda) = P(\{S_1, S_2, S_1\}|\lambda) = p_1 a_{12} a_{21} = 1(0.3)(0.2) = 0.06$$

$$P(Q_4|\lambda) = P(\{S_1, S_2, S_2\}|\lambda) = p_1 a_{12} a_{22} = 1(0.3)(0.8) = 0.24$$

and

$$P(O, Q_1|\lambda) = P(Red|S_1)P(Blue|S_1)\,P(Blue|S_1) = 0.75(0.25)(0.25) = 0.046875$$

$$P(O, Q_2|\lambda) = P(Red|S_1)P(Blue|S_1)\,P(Blue|S_2) = 0.75(0.25)(0.6) = 0.1125$$

$$P(O, Q_3|\lambda) = P(Red|S_1)P(Blue|S_2)\,P(Blue|S_1) = 0.75(0.6)(0.25) = 0.1125$$

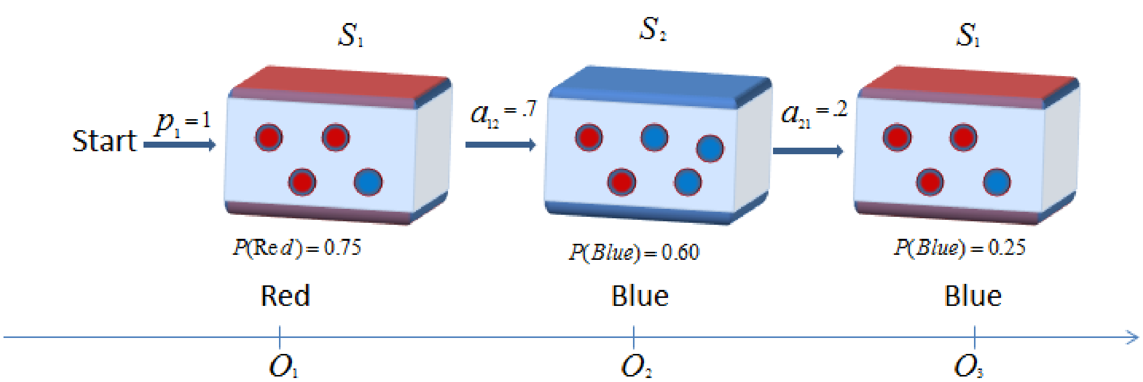
$$P(O, Q_4|\lambda) = P(Red|S_1)P(Blue|S_2)\,P(Blue|S_2) = 0.75(0.6)(0.6) = 0.27.$$

Thus,

$$P(O|\lambda) = 0.046875(0.49) + 0.1125(0.21) + 0.1125(0.06) + 0.27(0.24) = 0.118144.$$

Figure 3 shows the observations with the state sequence $Q_3 = \{S_1, S_2, S_1\}$.

Figure 3
A STATE SEQUENCE AND OBSERVATIONS OF AN HMM



Question 2: Given observation O and the model's parameter λ, find the "best fit" hidden state sequence Q.

Given the observation O and λ, there are many possible state sequences. We will need to find the sequence that maximizes the observation probability. In the ball game example, with the observations $O = \{Red, Blue, Blue\}$ and the initial probability $p = (1,0)$, we have four possible state sequences, listed in (1). We will choose the state sequence Q_k , which maximizes the likelihood of the observations,

$$k = argmax_{1 \le j \le 4}\{P(O, Q_j|\lambda)P(Q_j|\lambda)\}.$$

We can see from the earlier calculations that $Q_4 = \{S_1, S_2, S_2\}$ is the optimal state sequence, since it maximizes the probability of the observation O:

$$P(O|Q_4, \lambda) = P(O, Q_4|\lambda)P(Q_4|\lambda) = 0.27(0.24) = 0.0648.$$

Question 3: Suppose that you do not know the model's parameter λ. All you know is an observation sequence $O = \{Red, Blue, Blue\}$. Can you find λ ?

This is the hardest question, and to answer it we have to find the set of model parameters, λ, that maximizes the probability of observation $P(O|\lambda)$. Like other optimization strategies, we will start by guessing a set of parameters, λ, calculate $P(O|\lambda)$ then modify λ in each step until we locate the optimizer λ^* that gives the targeted observation probability.

The process is too complicated to do by hand. Therefore, we will not present a complete solution for the last question of the HMM here. Instead, we will describe the mathematical algorithm to solve this problem and demonstrate a full iteration of the training process Section 3.

SECTION 3: THREE MAIN PROBLEMS OF HMM AND ITS ALGORITHMS

3.1 Three Main Problems That an HMM Can Solve

In the ball example of the previous section, we mentioned the three questions or problems that we can solve when working with a hidden Markov model. In this section, we will present these three most significant problems of HMM in a general case. Applying HMM to solve a real-world problem, we face three critical questions:

1. Given the observation data $O = \{O_{t,}, 1 \le t \le T\}$ and the model parameters $\lambda = \{A, B, p\}$, compute the probability of observations, $P(O|\lambda)$.

2. Given the observation data $O = \{O_{t,}, 1 \le t \le T\}$ and the model parameters $\lambda = \{A, B, p\}$, find the best corresponding state sequence, $Q = \{q_{t,}, 1 \le t \le T\}$.

3. Given the observation data $O = \{O_{t,}, 1 \le t \le T\}$, calibrate HMM parameters $\lambda = \{A, B, p\}$ to maximize the probability of observation.

Researchers have found mathematical algorithms to solve each of these problems. The first were (Baum & Egon, 1967) and (Baum & Sell, 1968), who introduced the forward and backward algorithms to solve the first problem. Viterbi (1967) and Forney (1973) developed the Viterbi algorithm to solve the second problem. Baum, Petrie, Soules, and Weiss (1970) presented the Baum-Welch algorithm to solve the third problem for a single observation sequence. Over the next three decades, Levinson, Rabiner, and Sondhi (1983), and Li, Parizeau, and Plamondon (2000) implemented algorithms based on the Baum-Welch algorithm to calibrate HMM's parameters for multiple observation sequences.

In the following sections, we introduce the forward algorithm, the backward algorithm, the Viterbi algorithm and the Baum-Welch algorithm. Either the forward or backward algorithm can be used to calculate the probability of observation in problem 1, while both of these algorithms are needed in the Baum-Welch algorithm. For convenience, we use the notation $b_i(O_t)$ for conditional probability $b_i(k)$.

3.2 Forward Algorithm

The forward algorithm is an algorithm to find the probability of observation, $P(O|\lambda)$, given the HMM's parameters. In the ball example in Section 2.2, we solve this problem by listing all possible outcomes of the hidden states and calculating the probability of observations given each state sequence and summing them up to have $P(O|\lambda)$. However, the method is not efficient if we have a longer observation sequence, since the process requires too many calculations. To solve this issue, we have a better way to calculate the probability of observation, the forward algorithm.

3.2.1 Algorithm

In the forward algorithm, we calculate the likelihood of the model by considering the probability of observation if the last state is S_i:

$$P(O_1, O_2, \dots, O_T\,,\ q_T = \ S_i\,|\lambda).$$

Then, the observation probability $P(O|\lambda)$ is the sum of these conditional probabilities:

$$P(O|\lambda) = \sum_{i=1}^{N} P(O_1, O_2, \dots, O_T\,,\ q_T = \ S_i\,|\lambda).$$

The probability $P(O_1, O_2, \dots, O_T\,,\ q_T = \ S_i\,|\lambda)$ can be calculated recursively by defining the forward probability function $\alpha_t(i)$:

$$\alpha_t(i) = P(O_1, O_2, \dots, O_t\,,\ q_t = \ S_i\,|\lambda), 1 \le t \le T, 1 \le i \le N.$$

Recursively, we have

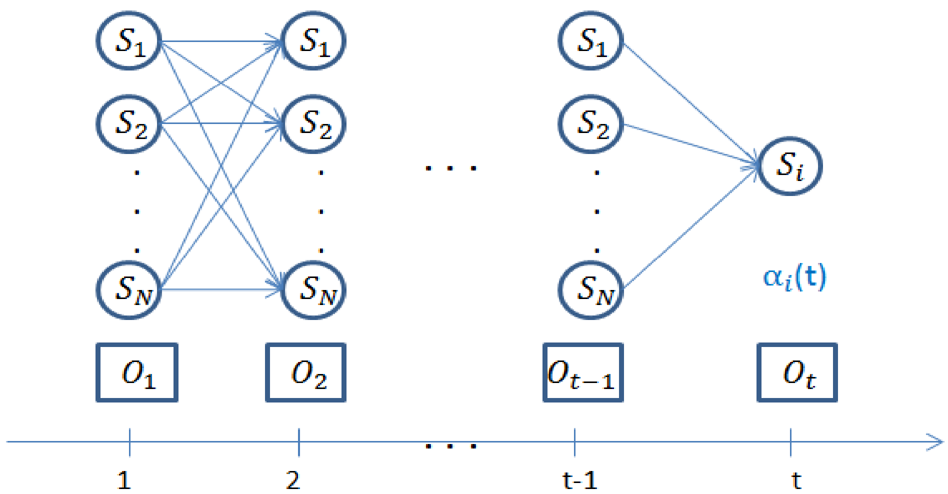
$$\alpha_{t+1}(j) = \left(\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right) b_j(O_{t+1}), 1 \le t < T, 1 \le j \le N.$$

The probability of observation $P(O|\lambda)$ is just the sum of all $\alpha_T(i), 1 \le i \le N$.

Figure 4 is the diagram of the joint probability function $\alpha_t(i)$.

Figure 4

JOINT PROBABILITY FUNCTION $\alpha_t(i)$



Algorithm 1 presents the forward algorithm for an HMM.

Algorithm 1
FORWARD ALGORITHM FOR ONE OBSERVATION SEQUENCE

1: Initialization: for i = 1, 2, ..., N

$$\alpha_{t=1}(i) = p_i b_i(O_1).$$

2: Recursion: for $t = 2, 3, \ldots, T$, and for $j = 1, 2, \ldots, N$, compute

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} \right] b_j(O_t).$$

3: Output:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i).$$

3.2.2 Example

We can apply the forward algorithm to calculate $P(O|\lambda)$ in the ball example in Section 2.2 with two states and two symbols, $v_1 = Red$ and $v_2 = Blue$, for each state.

Recall that the observations are

$$O = \{Red, Blue, Blue\},$$

$$A = \begin{bmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{bmatrix}, B = \begin{bmatrix} 0.75 & 0.25 \\ 0.40 & 0.60 \end{bmatrix}, \text{and } p = (1,0).$$

We have

$$P(O|\lambda) = \alpha_3(1) + \alpha_3(2) = P(O_1, O_2, O_3 | q_3 = S_1) + P(O_1, O_2, O_3 | q_3 = S_2).$$

Since $p = (p_1, p_2) = (1,0)$ at time $t = 1$, we have

$$\alpha_1(1) = P(O_1 | q_1 = S_1) = P(Red | q_1 = S_1) = p_1 b_1(1) = (1)(0.75) = 0.75$$

$$\alpha_1(2) = P(O_1 | q_1 = S_2) = P(Red | q_1 = S_2) = p_2 b_2(1) = 0.$$

Using the recursive formula, at time $t = 2$ we have

$$\alpha_2(1) = [\alpha_1(1)a_{11}+\alpha_1(2)a_{21}]b_1(2) = 0.75(0.7)(0.25) = 0.13125$$

$$\alpha_2(2) = [\alpha_1(1)a_{12}+\alpha_1(2)a_{22}]b_2(2) = 0.75(0.3)(0.6) = 0.135.$$

At time $t = 3$ we have

$$\alpha_3(1) = [\alpha_2(1)a_{11}+\alpha_2(2)a_{21}]b_1(2) = [0.13125(0.7) + 0.135(0.2)](0.25) = 0.029719$$

$$\alpha_3(2) = [\alpha_2(1)a_{12}+\alpha_2(2)a_{22}]b_2(2) = [0.13125(0.3) + 0.135(0.8)](0.6) = 0.088425.$$

Thus, the probability of observation is

$$P(O|\lambda) = \alpha_3(1) + \alpha_3(2) = 0.029719 + 0.088425 = 0.118144.$$

This result is consistent with the result that we found in Section 2.2. However, using the recursive formula significantly reduces the number of calculations needed to find $P(O|\lambda)$. In general, with T observations and N states, the traditional method (presented in Section 2.2) will need $2T(N^T)$ calculations, since we will have N^T possible state sequences and each sequence needs $2T$ calculations to calculate $P(Q_j|\lambda)$ and $P(O, Q_j|\lambda)$. In contrast, using the forward algorithm we will only need N^2T calculations.

3.3 Backward Algorithm

Similar to the forward algorithm, we can use a backward algorithm to calculate the probability $P(O|\lambda)$.

3.3.1 Algorithm

Let's denote the backward probability function

$$\beta_T(i) = 1, 1 \leq i \leq N$$

and

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_T | q_t = S_i, \lambda), 1 \leq t < T, 1 \leq i \leq N.$$

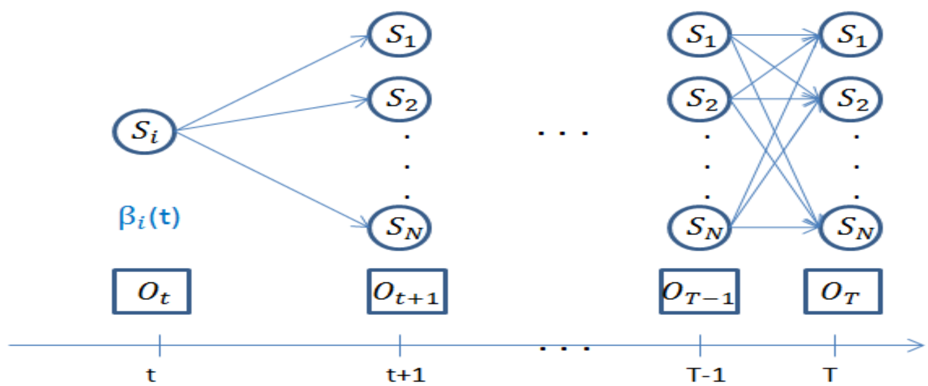
Then, we can calculate $\beta_t(i)$ by using the backward recursive formula:

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j)\, a_{ij} b_j(O_{t+1}), 1 \leq t < T, 1 \leq i \leq N.$$

The probability of observation $P(O|\lambda)$ is just the sum of all $p_i b_i(O_1)\beta_1(i), 1 \leq i \leq N$.

The conditional probability $\beta_t(i)$ is displayed in Figure 5.

Figure 5
CONDITIONAL PROBABILITY FUNCTION $\beta_t(i)$



Algorithm 2
BACKWARD ALGORITHM

1. Initialization: for $i = 1, ..., N$
$$\beta_T(i) = 1.$$
2. Recursion: for $t = T-1, T-2, ..., 1$, for $i = 1, ..., N$
$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j).$$
3. Output:
$$P(O|\lambda) = \sum_{i=1}^{N} p_i b_i(O_1) \beta_1(i).$$

3.3.2 Example

We apply the backward algorithm to calculate $P(O|\lambda)$ in the ball example in Section 2.2 with two states (S_1, S_2) and two symbols ($v_1 = Red$ and $v_2 = Blue$) for each state. Recall that we have

$$\{O_1, O_2, O_3\} = \{Red, Bue, Blue\},$$

$$A = \begin{bmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{bmatrix}, B = \begin{bmatrix} 0.75 & 0.25 \\ 0.40 & 0.60 \end{bmatrix}, \text{and } p = (1,0).$$

First, we calculate the forward function at time t.

At time $t = 3$:

$$\beta_3(1) = 1, \text{and } \beta_3(2) = 1.$$

At time $t = 2$:

$$\beta_2(1) = \beta_3(1)a_{11}b_1(O_3) + \beta_3(2)a_{12}b_2(O_3) = 1(0.7)(0.25) + 1(0.3)(0.6) = 0.335$$

$$\beta_2(2) = \beta_3(1)a_{21}b_1(O_3) + \beta_3(2)a_{22}b_2(O_3) = 1(0.2)(0.25) + 1(0.8)(0.6) = 0.53$$

At time $t = 1$:

$$\beta_1(1) = \beta_2(1)a_{11}b_1(O_2) + \beta_2(2)a_{12}b_2(O_2) = 0.335(0.7)(0.25) + 0.53(0.3)(0.8) = 0.157525$$

$$\beta_1(2) = \beta_2(1)a_{21}b_1(O_2) + \beta_2(2)a_{22}b_2(O_2) = 0.335(0.2)(0.25) + 0.53(0.8)(0.6) = 0.27215$$

The observation probability is

$$P(O|\lambda)=p_1b_1(O_1)\beta_1(1) + p_2b_2(O_1)\beta_1(2) = 1(0.75)(0.157525) + 0 = 0.118144.$$

The result is identical with the result using the forward algorithm. We also notice that

$$P(O, q_t = S_i|\lambda) = \alpha_t(i)\beta_t(i).$$

Thus, we can use the forward and backward functions to calculate the model's likelihood:

$$P(O|\lambda) = \sum_{i=1}^{N} P(O, q_t = S_i|\lambda)=\sum_{i=1}^{N} \alpha_t(i)\beta_t(i), 1 \le t \le T.$$

For example, with $t = 2$, we have

$$P(O|\lambda) = \alpha_2(1)\beta_2(1)+\alpha_2(2)\beta_2(2) = 0.13125(0.355) + 0.135(0.53) = 0.118144.$$

3.4 Viterbi Algorithm

The Viterbi algorithm is used to solve the second problem of HMM: find the "best fit" hidden states of the observations. The goal here is to find the best sequence of states Q when (O, λ) are given to maximize the probability of observation $P(O, Q|\lambda)$. There are many possible state sequences Q for an observation sequence O. Among these state sequences, we need to find the "best fit" sequence Q^* that satisfies

$$P(O, Q^*|\lambda) = max_{\text{all Q}}\{P(O, Q |\lambda)\}.$$

The process of selecting the hidden states is explained in the next section.

3.4.1 Algorithm

We define the function $\delta_t(j)$ with $1 \le t \le T$ and $1 \le j \le N$

$$\delta_t(j) = max_{1\le j\le N}\{P(q_1, q_2, \ldots, q_t = S_j, O_1, O_2, \ldots, O_t|\lambda\}.$$

Then the maximum of observation probability with an optimal state sequence Q^* is the maximum of function $\delta_T(j)$ on all possible values j,

$$P(O, Q^*|\lambda) = max_{1\le j\le N}\{\delta_T(j)\}.$$

The function $\delta_t(j)$ can be calculated recursively by initializing

$$\delta_1(j) = p_i\, b(O_1), 1 \le j \le N,$$

and using the recursive formula

$$\delta_{t+1}(j) = b_j\,(O_{t+1}) max_{1\le i\le N}\{[\delta_1(j) a_{ij}]\}\,.$$

Using $\delta_t(j)$ we can locate the most likely state q_t as

$$q_t = argmax_{1\le j\le N}\{\delta_t(j)\}.$$

The Viterbi algorithm is presented in Algorithm 3.

Algorithm 3

VITERBI ALGORITHM FOR ONE OBSERVATION SEQUENCE

1: Initialization:

$$\delta_1(j) = p_j b_j(O_1), \quad j = 1, 2, ..., N;$$

$$\phi_1(j) = 0.$$

2: Recursion: for $2 \le t \le T$, and $1 \le j \le N$

$$\delta_t(j) \quad = max_i[\delta_{t-1}(i)a_{ij}]b_j(O_{t+1})]$$

$$\phi_t(j) \quad = argmax_i[\delta_{t-1}(i)a_{ij}]$$

3: Output:

$$q_T^* \quad = argmax_i[\delta_T(i)]$$

$$q_t^* \quad = \phi_{t+1}(q_{t+1}^*),\ t = T-1, \ldots, 1$$

3.4.2 Example

We now apply the Viterbi algorithm to find the "best fit" hidden states Q^* (or the sequence of boxes) of the observation $O = \{Red, Blue, Blue\}$ in the ball example to maximize the probability of observation $P(O, Q^*|\lambda)$. Recall that

$$A = \begin{bmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{bmatrix}, B = \begin{bmatrix} 0.75 & 0.25 \\ 0.40 & 0.60 \end{bmatrix}, \text{and } p = (1,0).$$

We will calculate values of the probability function $\delta_t(i)$ and the index function $\Phi_t(i)$.

At time $t = 1$:

$$\delta_1(1) = p_1 b_1(O_1) = p_1 b_1(Red) = p_1 b_1(v_1) = p_1 b_1(1) = 0.75,$$

$$\delta_1(2) = p_2 b_2(O_1) = p_2 b_2(Red) = 0,$$

$$\Phi_1(1) = \Phi_1(2) = 0.$$

At time $t = 2$:

$$\delta_2(1) = max\{\delta_1(i)a_{i1}\}b_1(O_2) = \delta_1(1)a_{11}b_1(O_2) = 0.75(0.7)(0.25) = 0.13125.$$

$$\delta_2(2) = max\{\delta_1(i)a_{i2}\}b_2(O_2) = \delta_1(1)a_{12}b_1(O_2) = 0.75(0.3)(0.25) = 0.05625$$

$$\Phi_2(1) = argmax_{1 \le i \le N}\delta_1(i)a_{i1}\} = 1$$

$$\Phi_2(2) = argmax_{1 \le i \le N}\delta_1(i)a_{i2}\} = 1.$$

At time $t = 3$:

$$\delta_3(1) = max\{\delta_2(i)a_{i1}\}b_1(O_3) = \delta_2(1)a_{11}b_1(O_3) = 0.13125(0.7)(0.25) = 0.02297$$

$$\delta_3(2) = max\{\delta_2(i)a_{i2}\}b_2(O_3) = \delta_2(1)a_{12}b_2(O_3) = 0.13125(0.3)(0.6) = 0.02363$$

$$\Phi_3(1) = argmax_{1 \le i \le N}\{\delta_2(i)a_{i1}\} = 1$$

$$\Phi_3(2) = argmax_{1 \le i \le N}\{\delta_2(i)a_{i2}\} = 2$$

Thus, the "best fit" hidden states are

$$q_3 = argmax_{1 \le i \le N}\{\delta_3(i)\} = 2$$

$$q_2 = \Phi_3(q_3) = \Phi_3(2) = 2$$

$$q_1 = \Phi_2(q_2) = \Phi_2(2) = 1$$

In summary, the "best fit" state sequence is

$$Q^* = \{q_1, q_2, q_3\} = \{1, 2, 2\} = \{S_1, S_2, S_2\}.$$

This result agrees with the result that we found in Section 2.2.

With minor changes, the forward algorithm, backward algorithm, and Viterbi algorithm can be used for multiple observation sequences as well. Next we present the most important algorithm, the Baum-Welch algorithm, for two cases: single observation and multiple independent observations.

3.5 Baum-Welch Algorithm

We turn now to the solution for the third problem, which is the most difficult problem of HMMs. Here we have to find the parameters $\lambda = \{A, B, p\}$ given a finite observation sequence O. A fundamental method is finding a set of parameters λ that maximizes the probability of observation $P(O|\lambda)$. Unfortunately, given observation data, there is no way to find a global maximizer of $P(O|\lambda)$. However, we can find a local maximizer of $P(O|\lambda)$ using the Baum-Welch algorithm (Baum L. E., Petrie, Soules, & Weiss, 1970).

The process of the Baum-Welch algorithm is as follows: With an initial set of parameters, λ, the algorithm estimates a mew parameter λ^* using an optimization method to maximize the probability of observation $P(O|\lambda)$; thus we have $P(O|\lambda^*) > P(O|\lambda)$. Weterminate the process when we reach a desired error or number of interactions. The re-estimate parameters' process is explained in the next section.

3.5.1 Baum-Welch Algorithm for a Single Observation Sequence

The observation probability $P(O|\lambda)$ can be calculated by using both forward and backward functions, $\alpha_t(i)$ and $\beta_t(i)$.

$$P(O|\lambda) = \sum_{i=1}^{N} P(q_t = S_i|O, \lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i).$$

To describe the re-estimate parameters' procedure, we introduce a probability function, $\gamma_t(i)$, the probability of being in state S_i at time t given the observation sequence, as:

$$\gamma_t(i) = P(q_t = S_i|O, \lambda).$$

By applying Bayes' rule, we have

$$\gamma_t(i) = \frac{P(q_t = S_i, O|\lambda)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}$$

The probability of being in state S_i at time t and state S_j at time $t+1$, $\xi_t(i,j)$, is defined as

$$\xi_t(i,j) = P\left(q_t = S_i, q_{t+1} = S_j \middle| O, \lambda\right).$$

Similarly, using Bayes' rule, we have

$$\xi_t(i,j) = \frac{P\left(q_t = S_i, q_{t+1} = S_j, O|\lambda\right)}{P(O|\lambda)}$$

$$= \frac{P\left(O_1, O_2, \ldots, O_t, q_t = S_i|\lambda\right) P\left(O_{t+1}, O_{t+2}, \ldots, O_T, q_{t+1} = S_j|\ q_t = S_i, \lambda\right)}{P(O|\lambda)}$$

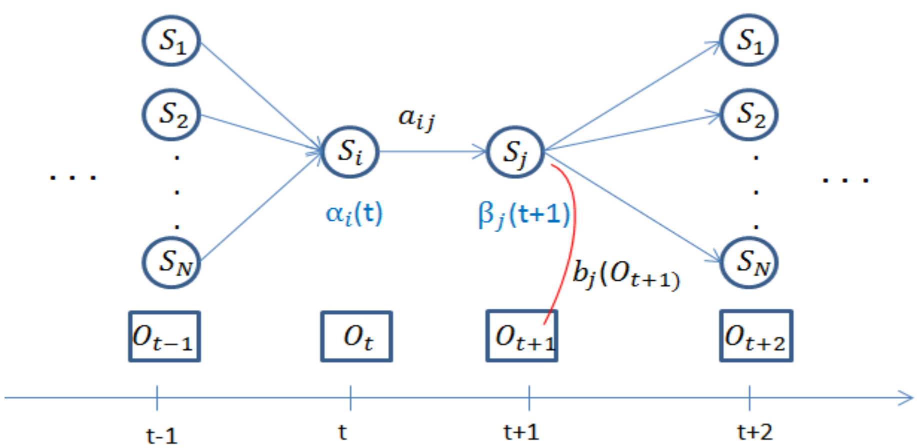
$$= \frac{P\left(O_1, O_2, \ldots, O_t, q_t = S_i|\lambda\right) P(O_{t+1}, q_{t+1} = S_j|\ q_t = S_i, \lambda)\ P\left(O_{t+1}, O_{t+2}, \ldots, O_T, q_{t+1} = S_j|\ \lambda\right)}{P(O|\lambda)}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}$$

The calculation of $\xi_t(i,j)$ is displayed in Figure 6.

Figure 6

PROBABILITY FUNCTION $\xi_t(i,j)$



Clearly, we have

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j).$$

Thus, we can consider $\gamma_t(i)$ is the expected number of times that state S_i is visited. Then, we have

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{the expected number of transitions made from state } S_i,$$

and

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{the expected number of transitions from state } S_i \text{ to } S_j.$$

We can use the following formulas to update parameter $\lambda^*$of a discrete HMM:

$$p_i^* = \text{expected number of times in state } S_i \text{ at time } 1 = \gamma_1(i), 1 \le i \le N$$

$$a_{ij}^* = \frac{\text{expected number of transitions from state } S_i \text{to } S_j}{\text{expected number of transitions made from state } S_i} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$b_j^*(k) = \frac{\text{expected number of times in state } S_j \text{and observing symbol} v_k}{\text{expected number of times in state } S_j} = \frac{\sum_{\substack{t=1 \\ |O_t=v_k}}^{T-1} \gamma_t(j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

The constraints of the model parameters $A, B,$ and p:

$$a_{ij} > 0, 1 \le i, j \le N \text{ and } \sum_{j=1}^{N} a_{ij} = 1, 1 \le i \le N \quad (2)$$

$$p_i > 0, 1 \le i \le N \text{ and } \sum_{i=1}^{N} p_i = 1 \quad (3)$$

$$b_i(k) > 0, 1 \le i \le N, 1 \le k \le M \text{ and } \sum_{k=1}^{M} b_i(k) = 1, 1 \le i \le N,$$

are automatically satisfied because of the definition of $\gamma_t(i)$ and we have $\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)$.

Baum, Petrie, Soules and Weiss (1970) proved that with an initial parameter λ, each time we update the model's parameters λ^* using the foregoing formulas, the probability of observation increases, $P(O|\lambda^*) > P(O|\lambda)$. The complete Baum-Welch algorithm for a discrete HMM with one sequence of observations is given here.

Algorithm 4

BAUM-WELCH ALGORITHM FOR A DISCRETE HMM WITH A SINGLE OBSERVATION SEQUENCE

---

1. Initialization: input parameters λ, the tolerance tol, and a real number $\triangle$
2. Repeat until $\triangle < tol$
   - Calculate $P(O, \lambda)$ using forward algorithm
   - Calculate new parameters λ^*: for $1 \le i \le N$

$$p_i^* = \gamma_1(i)$$

$$a_{ij}^* = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \ 1 \le j \le N$$

$$b_i^*(k) = \frac{\sum_{t=1}^{T} |_{O_t=v_k} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}, \ 1 \le k \le M$$

   - Calculate $\triangle = |P(O, \lambda^*) - P(O, \lambda)|$
   - Update $\lambda = \lambda^*$
3. Output: parameters λ.

For a continuous HMM, we present the case when the probability of observation at time t, $b_i(k) = b_i(O_t)$, is a Gaussian distribution. In this case, $b_i(O_t)$ is the density of a normal distribution at the value O_t, as mentioned inSection 2.1:

$$b_i\,(O_t) = \mathbf{N}(O_t\,, \mu_i, \sigma_i).$$

And, the updated parameters are

$$\mu_i^* = \frac{\sum_{t=1}^{T-1} \gamma_t(i) O_t}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

and

$$\sigma_i^* = \frac{\sum_{t=1}^{T-1} \gamma_t(i)(O_t - \mu_i)(O_t - \mu_i)'}{\sum_{t=1}^{T-1} \gamma_t(i)},$$

where $(O_t - \mu_i)'$ is the transpose of $(O_t - \mu_i)$, and $1 \le i \le N$.

3.5.2 Example

In this example, we use the Baum-Welch algorithm to re-estimate the model's parameters one time to solve the third question of the ball game described in Section 2.2. First, we restate the ball problem.

Suppose that we have two boxes of balls, and each box contains two possible ball colors: red or blue. A player will first choose a box by using a process that satisfies the first order of a Markov chain with a constant transition probability matrix A. Then, the player will pick a ball from the selected box. Given the observations of three successive tries $O = \{Red, Blue, Blue\}$, can we find the matrix A, the probability of observing a red or blue ball in his first try (vector p), and the numbers of red and blue balls in each box (or the matrix B: probability of getting a red or blue ball from each box)? The answer is yes. We can use the Baum-Welch algorithm to find the answer.

Recall that the discrete HMM has two states ($S_1 = box\ one, S_2 = box\ two$) and two symbols for each state ($v_1 = Red, v_2 = Blue$). We choose the initial parameters, which are the ones that we used in the examples in Sections 3.2.2, 3.3.2 and 3.4.2, $\lambda = \{A, B, p\}$, to avoid recalculating forward and backward probabilities. The initial parameters are

$$A = \begin{bmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{bmatrix}, B = \begin{bmatrix} 0.75 & 0.25 \\ 0.40 & 0.60 \end{bmatrix}, \text{and } p = (1,0).$$

We know from the previous examples that with the initial parameters, the probability of observations is $P(O|\lambda) = 0.11814$. To update the model's parameters, we need to calculate the expected probabilities $\gamma_t(i)$ and $\xi_t(i,j)$and use these values to re-estimate the parameters, in the hope that the new parameter will increase the likelihood of the model. Using the forward and backward results from the examples in Sections 3.2.2 and 3.3.2 we calculate $\gamma_t(i)$ and $\xi_t(i,j)$ at time t.

- $\gamma_t(i)$, the probability of being in state S_i at time t.

At time $t = 1$:

$$\gamma_1(1) = 1, \gamma_1(2) = 0$$

At time $t = 2$:

$$\gamma_2(1) = 0.39438, \gamma_1(2) = 0.60562$$

At time $t = 3$:

$$\gamma_3(1) = 0.25155, \gamma_1(2) = 0.74845$$

- $\xi_t(i,j)$, the probability of being in state S_i at time t and state S_j at time $t + 1$.

At time $t = 1$:

$$\xi_1(1,1) = 0.39438, \xi_1(1,2) = 0.60562, \xi_1(2,1) = 0, \xi_1(2,2) = 0.$$

At time $t = 2$:

$$\xi_2(1,1) = 0.19441, \xi_2(1,2) = 0.19997, \xi_2(2,1) = 0.05713, \xi_2(2,2) = 0.54848$$

Now, we re-estimate the model's parameters.

- The probability of being in state S_i at time $t = 1$

$$p^* = \left(\gamma_1(1), \gamma_1(2)\right) = (1,0).$$

- The state transition probability A^*

$$a_{11}^* = \frac{\xi_1(1,1) + \xi_2(1,1)}{\gamma_1(1) + \gamma_2(1)} = \frac{0.39438 + 0.19441}{1 + 0.39438} = 0.42226$$

$$a_{12}^* = \frac{\xi_1(1,2) + \xi_2(1,2)}{\gamma_1(1) + \gamma_2(1)} = \frac{0.60562 + 0.19997}{1 + 0.39438} = 0.57774$$

$$a_{21}^* = \frac{\xi_1(2,1) + \xi_2(2,1)}{\gamma_1(2) + \gamma_2(2)} = \frac{0 + 0.05713}{0 + 0.60562} = 0.09434$$

$$a_{22}^* = \frac{\xi_1(2,2) + \xi_2(2,2)}{\gamma_1(2) + \gamma_2(2)} = \frac{0 + 0.54848}{0 + 0.60562} = 0.90566$$

- The matrix of observation probability B^*

$$b_1^*(1) = \frac{I_{\{O_1=Red\}}\gamma_1(1)+I_{\{O_2=Red\}}\gamma_2(1)+I_{\{O_3=Red\}}\gamma_3(1)}{\gamma_1(1)+\gamma_2(1)+\gamma_3(1)} = \frac{\gamma_1(1)}{\gamma_1(1)+\gamma_2(1)+\gamma_3(1)} = 0.60756$$

$$b_1^*(2) = \frac{I_{\{O_1=Blue\}}\gamma_1(1)+I_{\{O_2=Blue\}}\gamma_2(1)+I_{\{O_3=Blue\}}\gamma_3(1)}{\gamma_1(1)+\gamma_2(1)+\gamma_3(1)} = \frac{\gamma_2(1)+\gamma_3(1)}{\gamma_1(1)+\gamma_2(1)+\gamma_3(1)} = 0.39244$$

$$b_2^*(1) = \frac{I_{\{O_1=Red\}}\gamma_1(2)+I_{\{O_2=Red\}}\gamma_2(2)+I_{\{O_3=Red\}}\gamma_3(2)}{\gamma_1(2)+\gamma_2(2)+\gamma_3(2)} = \frac{\gamma_1(2)}{\gamma_1(2)+\gamma_2(2)+\gamma_3(2)} = 0$$

$$b_2^*(2) = \frac{I_{\{O_1=Blue\}}\gamma_1(2)+I_{\{O_2=Blue\}}\gamma_2(2)+I_{\{O_3=Blue\}}\gamma_3(2)}{\gamma_1(2)+\gamma_2(2)+\gamma_3(2)} = \frac{\gamma_2(2)+\gamma_3(2)}{\gamma_1(2)+\gamma_2(2)+\gamma_3(2)} = 1$$

In summary, we have new parameters:

$$p^* = (1,0), A^* = \begin{bmatrix} 0.42226 & 0.57774 \\ 0.09434 & 0.90566 \end{bmatrix}, B^* = \begin{bmatrix} 0.60756 & 0.39244 \\ 0.0000 & 1.0000 \end{bmatrix}.$$

With the fresh set of parameters, $\lambda^* = \{A^*, B^*, p^*\}$, using the forward algorithm we calculate the new likelihood of the model and have the result $P(O|\lambda^*) = 0.40574$. We can see that after only one iteration, with the re-estimated parameters λ^*, the probability of observations increases from 0.11814 to 0.40574. If we continue the re-estimation process, we will have an optimal solution for the problem.

3.5.3 Baum-Welch Algorithm for Multiple Observation Sequences

Algorithms of HMM for multiple observation sequences were presented by (Li, Parizeau, & Plamondon, 2000). In this paper, we introduce HMM's algorithms for various observation sequences with an assumption that the observations are independent.

Suppose that we have $L-$ observation sequence $O = \{O_t^{(1)}, O_t^{(2)}, \ldots, O_t^{(L)}, 1 \le t \le T\}$, where $O^{(l)}, 1 \le l \le L$ are independent. The probability of observation $P(O|\lambda)$ can be calculated by $P(O|\lambda) = \prod_{l=1}^{L} P(O^{(l)}|\lambda)$, where $P(O^{(l)}|\lambda)$ is probability of observation $O^{(l)}$ given λ. Using the same definitions as in the previous sections, we denote $\alpha_t^{(l)}(i), \beta_t^{(l)}(i), \gamma_t^{(l)}(i)$, and $\xi_t^{(l)}(i,j)$ probability functions of observation sequence $\{O_t^{(l)}, 1 \le t \le T\}$. The Baum-Welch algorithm for multiple observation sequences is presented here.

Algorithm 5

BAUM-WELCH ALGORITHM FOR A DISCRETE HMM WITH MULTIPLE OBSERVATION SEQUENCES

1. Initialization: input parameters λ, the tolerance tol, and a real number $\triangle$
2. Repeat until $\triangle < tol$
   - Calculate $P(O, \lambda) = \Pi_{l=1}^{L} P(O^{(l)}|\lambda)$ using the forward algorithm
   - Calculate new parameters $\lambda^* = \{A^*, B^*, p^*\}$, for $1 \le i \le N$

$$p_i^* = \frac{1}{L} \sum_{l=1}^{L} \gamma_1^{(l)}(i)$$

$$a_{ij}^* = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T^{(l)}-1} \xi_t^{(l)}(i,j)}{\sum_{l=1}^{L} \sum_{t=1}^{T^{(l)}-1} \gamma_t^{(l)}(i)}, \ 1 \le j \le N$$

$$b_i^*(k) = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T^{(l)}} |_{O_t^{(l)}=v_k^{(l)}} \gamma_t^{(l)}(i)}{\sum_{l=1}^{L} \sum_{t=1}^{T^{(l)}} \gamma_t^{(l)}(i)}, \ 1 \le k \le M$$

   - Calculate $\triangle = |P(O, \lambda^*) - P(O, \lambda)|$
   - Update

$$\lambda = \lambda^*.$$

3. Output: parameters λ.

If the observation probability $b_i(k)$, defined in Section 2.1, is Gaussian, $b_i(k) = b_i(O_t) = \mathbf{N}(O_t, \mu_i, \sigma_i)$, we will have a continuous HMM and the updated parameters are

$$\mu_i^* = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T^{(l)}-1} \gamma_t^{(l)}(i) O_t^{(l)}}{\sum_{l=1}^{L} \sum_{t=1}^{T-1} \gamma_t^{(l)}(i)}$$

and

$$\sigma_i^* = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T^{(l)}-1} \gamma_t^{(l)}(i) \left(O_t^{(l)} - \mu_i\right) (O_t^{(l)} - \mu_i)'}{\sum_{l=1}^{L} \sum_{t=1}^{T-1} \gamma_t^{(l)}(i)},$$

where $\left(O_t^{(l)} - \mu_i\right)'$ is the transpose vector of $(O_t^{(l)} - \mu_i)$, $1 \le i \le N$, and $1 \le l \le L$.

SECTION 4: APPLICATIONS OF HMM IN FINANCE

4.1 Applications of HMM in Finance

In the previous sections, we presented the basic concepts and algorithms of the original version of the hidden Markov model introduced by (Baum & Petrie, 1966). In this section, we give a brief overview of its applications in finance and actuarial fields.

Recently, researchers have applied HMM for forecasting stock prices. HMM was used to forecast stock price for interrelated markets in (Hassan & Nath, 2005). HMM with two states was used to predict regimes in market turbulence, inflation and industrial production index by (Kritzman, Page, & Turkington, 2012). Nguyen used HMM with both single and multiple observations to forecast economic regimes and stock prices (Nguyen N. , 2014). The following year, Nguyen used HMM for single observation data to predict regimes of some economic indicators and made stock selections based on the performances of these stocks during the predicted regimes (Nguyen & Nguyen, 2015). HMM was also used to analyze stock market trends in (Kavitha, Udhayakumar, & Nagarajan, 2013). Previously, Lajos (2011) applied the HMM to predict S&P daily prices and developed a strategy to trade stock. And before that, Idvall and Jonsson (2008) implemented the HMM algorithms on foreign exchange data.

4.2 Applications of Regime-Switching Models in Finance

Applications of HMM for finance were presented in (Mamon & Elliott, 2004). However, many of the papers it included used the regime-switching model introduced in (Hamilton, 1989), which is not identical to the HMM introduced in (Baum & Petrie, 1966). Based on the principal concepts of state sequences, Hamilton developed a regime-switching model for nonstationary time series and the business cycle. In the regime-switching model (RSM), observation variables were generated by an autoregression model, whose parameters were optimized by a discrete Markov chain. Many researchers have used the RSM and claimed that the model is the same as HMM. Although the RSM and HMM are both associated with regimes or hidden states, the RSM should be viewed as a regression model with regime-shift variables. Furthermore, HMM is abroad model that allows a more flexible relationship between observation data and its hidden state sequence. We summarize some applications of RSM in finance next.

Investigators have been using RSM to detect financial crises for decades. Bonnie (1998) investigated the dynamic impact of macroeconomic aggregates on housing prices and housing stocks. Elliott and Wilson (1995) used RSM to model short-term interest rates, which assumed that the mean-reverting level follows a finite-state and continuous-time Markov chain. Garcia and Parron (1996) had used three-state RSM and a regression method to estimate the U.S. real interest rate and inflation rate. In the early 2000s Ang and Bekaert (2002) applied a regime-shift model for international asset allocation. Guidolin and Zimmermann (2005) used a four-state RSM to study asset allocation decisions on asset returns. And in 2007, Erlwein and Mamon implemented HMM for a financial data set of 30-day Canadian Treasury bill yields. More recently, in 2013, Zhu and Cheng used RSM to examine macroeconomic risk level. Also in that year, Nneji, Brooks and Ward used the RSM for investigating the impact of the macroeconomy on the dynamics of the residential real estate market in the United States.

4.3 Available HMM Software Packages

Many packages of HMM's codes are available as research tools for researchers. We list the two most common packages that were written in R and MatLab.

HMM package in R written by Harte (2016). In this package, readers can find manual instructions to download and use the package to solve these three main tasks of HMM: calibrate model parameters, calculate the probability of observations and find the hidden state sequence. Some main function codes provided in Appendix B were based on the package. The package can be downloaded from this link: https://cran.r-project.org/web/packages/HiddenMarkov/index.html.

Hidden Markov Model Toolbox (HMM) in MatLab by Chen (2016). This package includes Viterbi, HMM filter, HMM smoother, EM algorithm for calibrating the parameters of HMM, etc. Users can download the whole package from this link: http://www.mathworks.com/matlabcentral/fileexchange/55826-pattern-recognition-and-machine-learning-toolbox.

SECTION 5: HMM FOR MORTGAGE-BACKED SECURITIES EXCHANGE-TRADED FUNDS

HMM has been used as a powerful tool to predict economic trends and stock prices. However, it has not been used by researchers for mortgage-backed securities. In this section, we give a brief overview of mortgage-backed securities and apply HMM to analyze mortgage-backed securities exchange-traded funds (MBS ETFs).

In the HMM application portion, first we use the Baum-Welch algorithms to calibrate the model's parameters and the Viterbi algorithm to locate the hidden states (or regimes) of observation data. Second, we describe a method of using HMM to predict the probability of recession states and develop a strategy to predict some MBS ETFs. Finally, we use HMM to select MBS ETF portfolios.

5.1 Overview of Mortgage-Backed Securities

Mortgage-backed securities have been used as investment instruments for investors and lenders since the 1980s. The MBS industry provides banks with more cash to make more mortgage loans and keeps mortgage rates competitive and mortgages readily available. Many exchange-traded funds that focus exclusively on mortgage securities, called MBS ETFs, were formed to make mortgage investments more feasible for individual investors.

The U.S. housing market reached its peak in mid-2006, when it began declining. According to a 2007 report in the *Economist* (2007), from 1997 to 2006, the price of a typical American house increased by 124 percent. The U.S. housing market experienced the subprime mortgage crisis during the economic recession of December 2007 to June 2009. After seven years of declining rates, from 6.825 percent in June 2006 to 3.682 percent in May 2013, the 30-year fixed-rate mortgage increased to 4.004 percent in August 2015. The changes in interest rates affect MBS in opposite directions. If interest rates increase, MBS prices will decrease. Conversely, when interest rates drop, MBS securities will rise.

MBS and mortgage rates are driven by many economic factors, such as gross domestic product, inflation rate and interest rate. Researchers have investigated the performance of some economic indicators to evaluate the MBS. Zenios (1993) considered both the future interest rate and the prepayment activity of the MBSs to simulate an MBS portfolio. Calhoun and Deng (2002) analyzed the results of using different loan-level statistical models for fixed and adjustable rate mortgages. Kau, Keenan, Muller and Epperson (1992) investigated many intricacies of a fixed-rate mortgage model and concluded that "default always lowers the value of the mortgage to the lender, whereas none financial prepayment always raises it."

In the following sections, we select some MBS ETFs and use HMM to investigate their behaviors.

5.2 Data Selection

As mentioned in Section 5.1, MBS have been known to investors since the 1980s. However, individual investors had limited access to MBS until the mortgage crisis in 2007. Several ETFs that focus exclusively on MBS were provided for trading by many brokers. In this research, we collect historical data of MBS ETFs that are actively traded on the market, which can be downloaded from https://finance.yahoo.com. These MBS ETFs are listed in Table 1.

The data were gathered in three frequencies: daily, weekly and monthly. For each of these ETFs we have four observation sequences: "close," "open," "low" and "high" prices. We assume throughout this paper that the four observation sequences are independent.

You can see from Table 1 that most of the MBS ETFs were incepted recently. Therefore, to show applications of HMM to MBS ETFs in the following sections (5.3, 5.4 and 5.5), we will use the iShare MBS ETF known as MBB, because this ETF has long enough historical data to be useful.

Table 1
LIST OF MBSETFs

Broker	Trading Symbol	Issued Date
iShares	MBB	03/13/2007
	GNMA	02/14/2012
	CMBS	02/14/2012
Vanguard	VMBS	11/23/2009
Barclays	MBG	01/30/2009
FlexShares	MBSD	01/04/2016
First Trust	LMBS	11/05/2014

5.3 Using HMM to Find the Hidden States of Observations

In this section, we present the results of using HMM to find hidden states (or regimes) of fund MBB.

The observation price of MBB at time t is a continuous random variable. We assume that the observation probability $b_i(O_t) = P(q_t = S_i, O_t|\lambda)$ is the density of the normal distribution with mean μ_i and standard deviation α_i at point $O_t, 1 \leq i \leq N$. Therefore, the symbol v_k is omitted in the continuous HMM, and the parameter of the model is $\lambda = \{A, \mu, \sigma, p\}$.

To use HMM, we first need to choose observation data and the number of states, N. To make it simple, we just use one observation series, the "close" price of MBB, and we choose two, three or four states. In the two-state HMM, we assume that the observation data have two states (or regimes): growth and regression. In three-state HMM we assume that an observation sequence has three states: growth, moderate, and regression. In four-state HMM, we suppose an observation sequence has four states: strong growth, weak growth, weak regression, and regression. The HMM models that we use in this section are continuous and for a single observation sequence.

Now, let's explore how to use the two-state HMM to find the hidden state sequence for the MBB's close price. Suppose the observation data, O, is the MBB's weekly prices from March 16, 2007, to October 26, 2016.

We choose initial parameters $\lambda = \{A, \mu, \sigma, p\}$ for the Baum-Welch algorithm for N state HMM as:

$$A = (a_{ij}) = \left(\frac{1}{N}\right), \mu_i = m, \sigma_i = s.\varepsilon_i, p = (1,0, \dots ,0), i = 1, \dots , N,$$

where m and s are the mean and standard deviation of the observation sequence O, respectively, and ε is a random number from the standard normal distribution.

Researchers can choose initial parameters for the Baum-Welch in different ways as long as they satisfy the conditions of A in (2) and p in (3) presented earlier.

First, we use the Baum-Welch algorithm with two states, $N = 2$, to calibrate HMM's parameters, $\lambda = \{A, \mu, \sigma, p\}$. The Baum-Welch algorithm gives us the following results:

- Transition matrix A:

$$A = \begin{bmatrix} 0.9932 & 0.0068 \\ 0.0147 & 0.9853 \end{bmatrix}$$

- Means and variances of two observation probability distributions for two states:

$$\mu_1 = 108.54 \quad \sigma_1 = 0.97$$
$$\mu_2 = 103.54 \quad \sigma_2 = 2.40$$

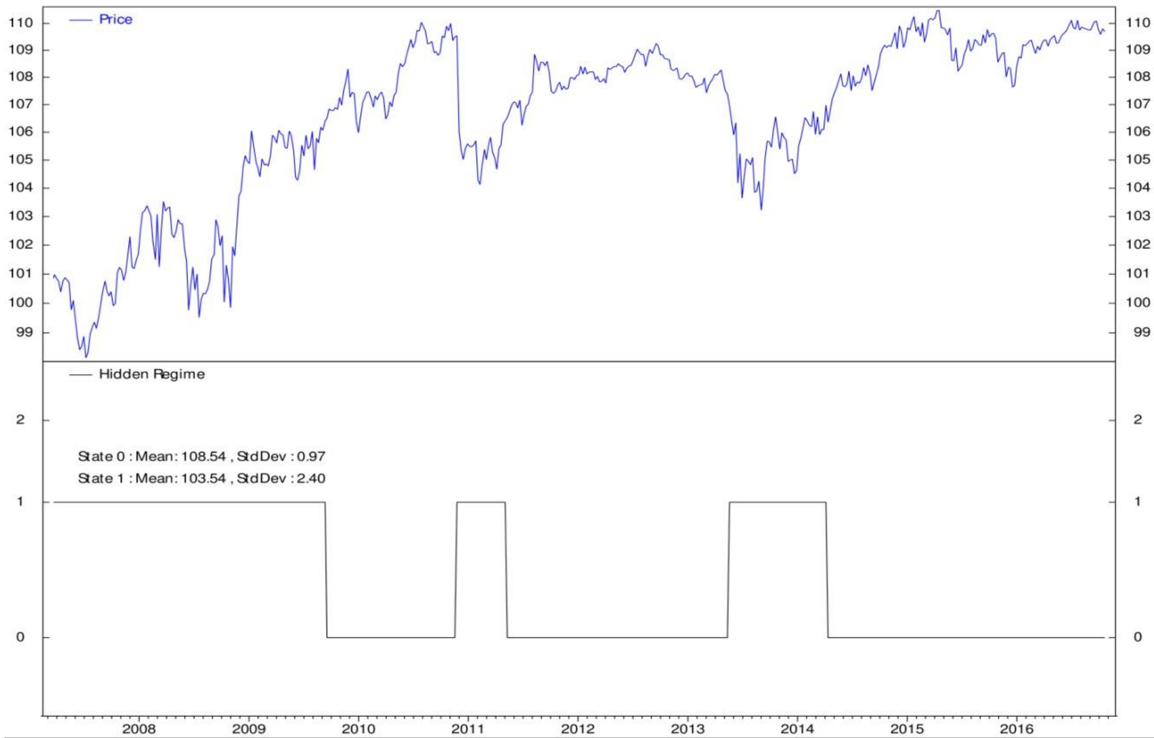
- Initial probability of being in state 0 or 1 at time 0: $p = (0,1)$.

After calibrating the HMM's parameters, we have to define the two states based on the results. In general, stocks during the growth state will have higher returns and lower volatilities than those during the regression state. However, the two conditions may not happen at the same time, so we define state 0and state 1based on the ratio of the means and standard deviations of the two normal distributions.

We define state 1as the state that has a lower ratio $\frac{\mu_i}{\sigma_i}, \boldsymbol{i = 1, 2}$, and state 0 as the state that has a higher ratio. In this case, we have $\mu_1 > \mu_2$ and $\boldsymbol{\sigma_1 < \sigma_2}$**.** Thus, we choose state 0 as the state that corresponds with normal distribution $\mathbf{N(108.54, 0.97)}$, and state 1 as the state that corresponds with normal distribution $\mathbf{N(103.54, 2.4)}$.

Finally**,** we use the Viterbi algorithm with the calibrated parameters to find "the fittest" hidden states of the observation data. The results are shown in Figure 7. Results for weekly and monthly data are presented in Figures 1 and 2 in the Appendix. We can see from Figure 7 that HMM captures well the hidden states of observation data. MBB price was in state 1 (low returns and high volatilities) during the economic recession time 2007-2009, the first quarter of 2011 and from the second quarter of 2013 to the first quarter of 2014.

Figure 7
FINDING TWO STATES OF MBB DAILY PRICE USING THE VITERBI ALGORITHM



Similar to the two-state HMM, we have the following results of the three-state and four-state HMM.

Results of using three-state HMM to calibrate the model's parameters and find hidden states of MBB observation:

- Transition matrix A:

$$A = \begin{bmatrix} 0.9565 & 0.0356 & 0.0079 \\ 0.0070 & 0.9978 & 0.0152 \\ 0.0063 & 0.0258 & 0.9979 \end{bmatrix}$$

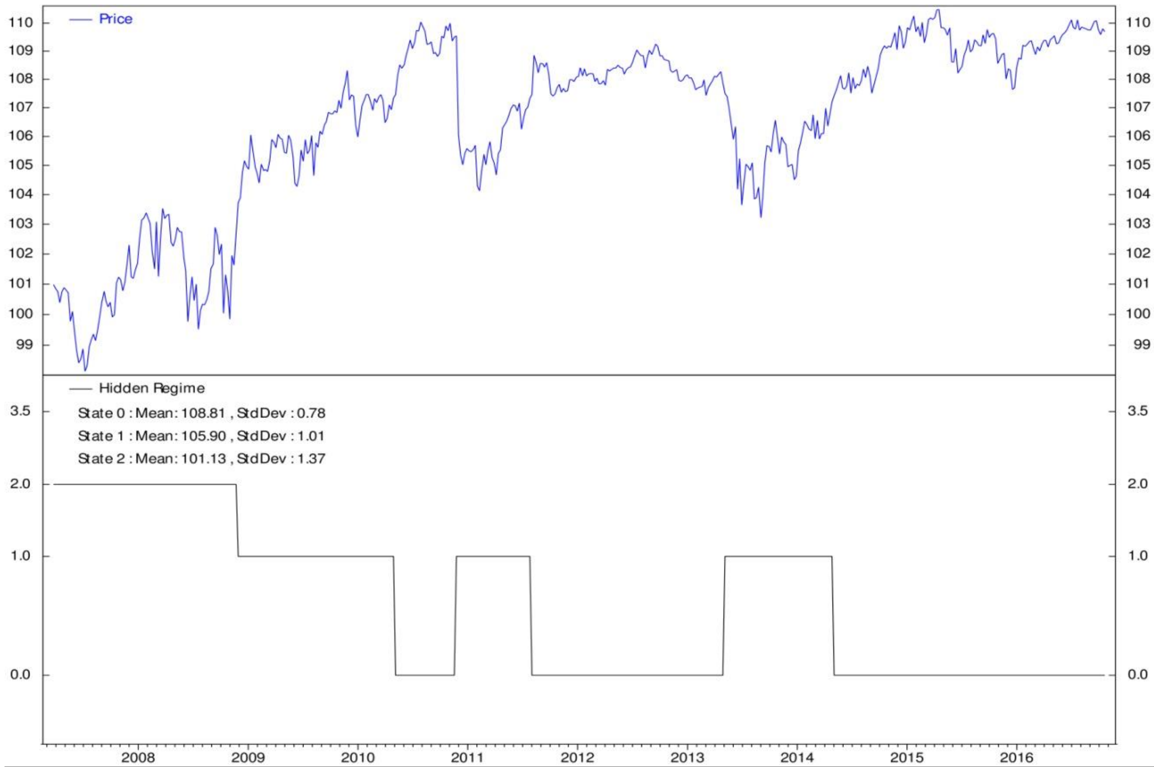
- Means and variances of three observation probability distributions (three normal distributions):

$$\mu_0 = 108.81 \quad \sigma_0 = 0.78$$
$$\mu_1 = 105.90 \quad \sigma_1 = 1.01$$
$$\mu_2 = 101.13 \quad \sigma_2 = 1.73$$

- Initial probability of being in state 0, 1, or 2at time 0: $p = (0,0,1)$.

The hidden sequence of the observations using the three-state HMM is presented in Figure 8.

Figure 8
FINDING THREE STATES OF MBB DAILY PRICES USING THE VITERBI ALGORITHM



Results of using four-state HMM to calibrate the model's parameters and find hidden states of MBB observation:

- Transition matrix A:

$$A = \begin{bmatrix} 0.6885 & 0.2919 & 0 & 0.0196 \\ 0.7251 & 0.2677 & 0.0072 & 0 \\ 0.0087 & 0 & 0.8594 & 0.1119 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

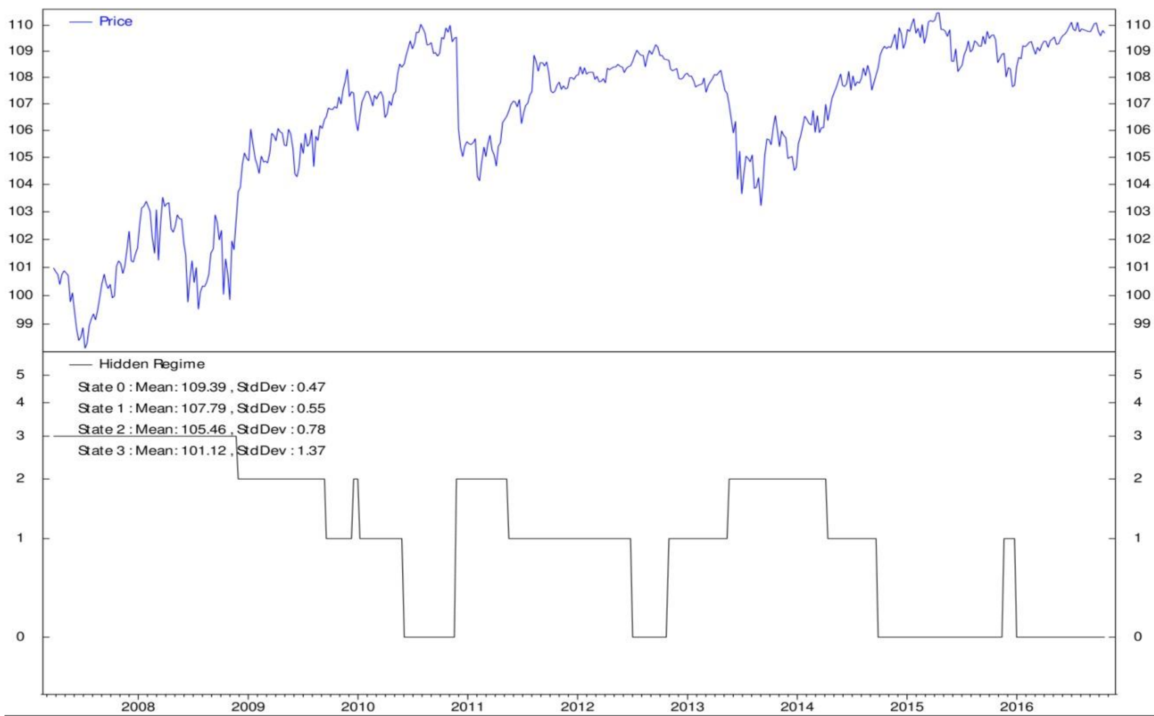
- Means and variances of four observation probability distributions (four normal distributions):

$$\mu_0 = 109.39 \quad \sigma_0 = 0.47$$
$$\mu_1 = 107.79 \quad \sigma_1 = 0.55$$
$$\mu_2 = 105.46 \quad \sigma_2 = 0.78$$
$$\mu_3 = 101.12 \quad \sigma_3 = 1.37$$

- Initial probability of being in state 0 to 3 at time 0: $p = (0,0,0,1)$.

The "best fit" hidden state sequence using the four-state HMM is displayed in Figure 9.

Figure 9
FINDING FOUR STATES OF MBB WEEKLY PRICES USING THE VITERBI ALGORITHM



The results show that we can use HMM to find hidden states of historical observation data. Furthermore, we can use HMM to predict the probability of being in different states in the future. We present the technique in the next section.

5.4 Using HMM to Predict Probability of Being in the Recession State

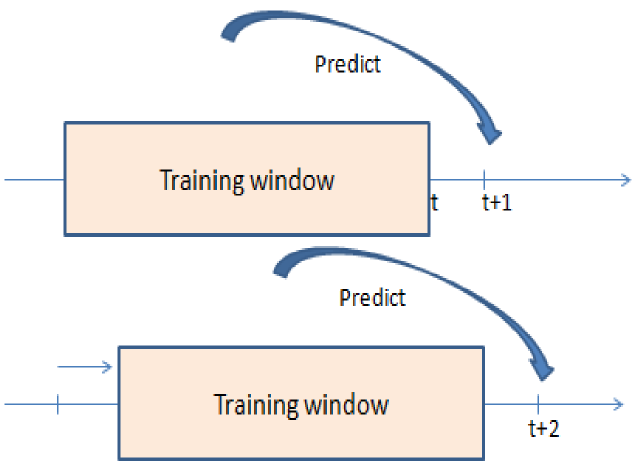
In this section, we use HMM to predict the probability of MBB being in the recession state. We use HMM for both single observation data and multiple observation data in this section. For single observation data we use MBB "close" price, and for multiple observation data we use MBB "close," "open," "low" and "high" prices.

To make the predictions and model validations, we use two-year MBB daily data from October 16, 2014, through October 26, 2016. We use fixed-length windows of one-year data, with each window having 252 consecutive trading days, to predict the probability of MBB being in recession the next day. The first data window, from October 30, 2014, to October 27, 2015, was used to predict the probability of being in the recession state on October 28, 2015.

We first use the Baum-Welch algorithm to calibrate HMM's parameters and then define states 0 and 1 based on the calibrated parameters of the two normal distributions: μ, σ. The recession state would have lower μ/σ compared to those of the "bull" market. Using the defined states and based on the first-order property of the Markov chain of HMM's state sequences, we can predict probability of being in state 1 by using the $\xi_t(i,j)$ function. The observation probability was calculated by using transition matrix A and the normal probability distribution corresponding to the recession state. After the first prediction, we will move the training window up one day and use the new data set to make the second prediction and so on. The moving window technique is presented in Figure 10.

Figure 10

MOVING WINDOW FOR PREDICTION



The predicted probability of being in the recession state for a one-year period is presented in Figures 11 and 12. In Figure 11 we use one observation sequence, the "close" price of MBB, and in Figure 12 we use four observation sequences: "close," "open," "low" and "high" price.

Figure 11

PREDICTED PROBABILITY OF BEING IN A "BEAR" MARKET (STATE 1) USING ONE OBSERVATION SEQUENCE

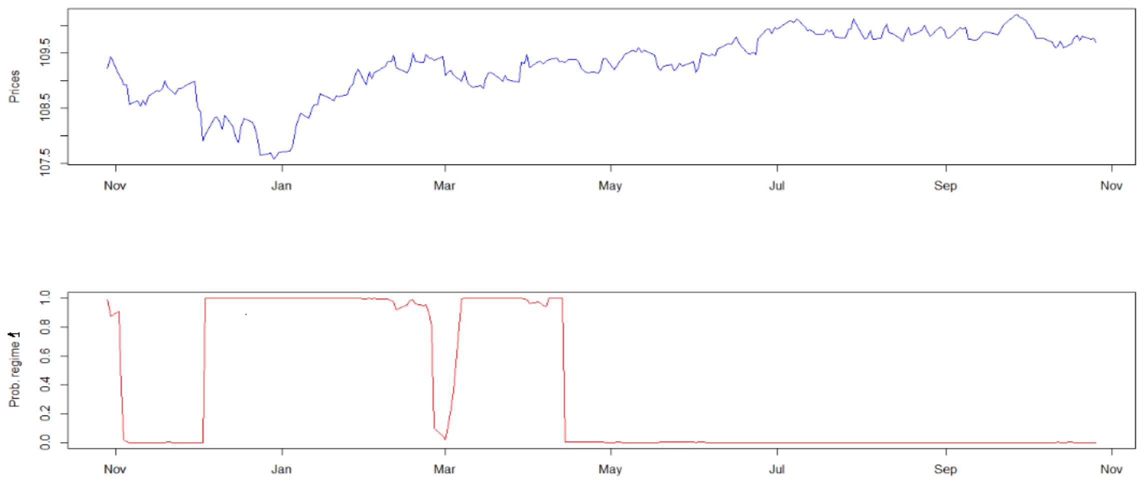
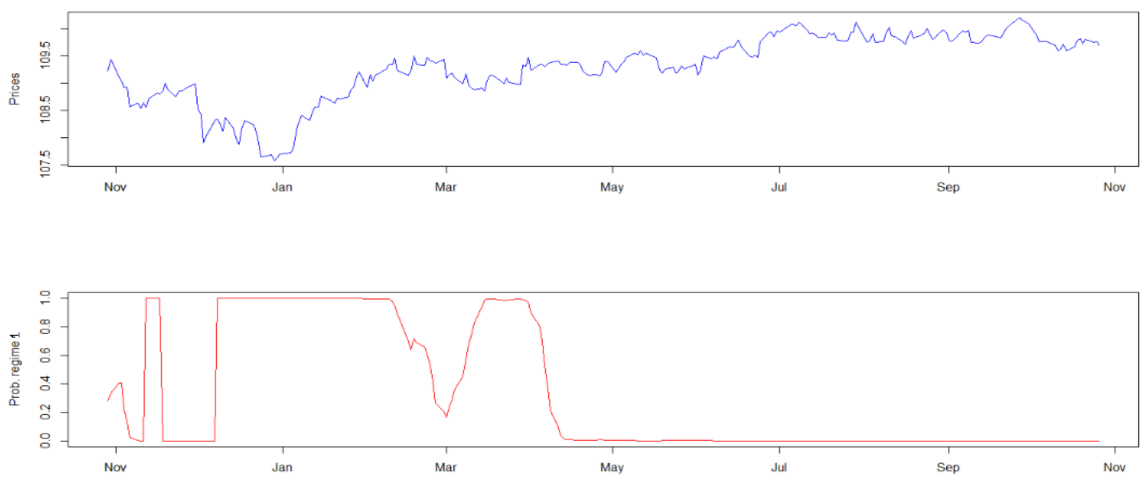


Figure 12

PREDICTED PROBABILITY OF BEING IN A "BEAR" MARKET (STATE 1) USING FOUR OBSERVATION SEQUENCES



5.5 Using HMM to Predict Prices and Trade MBSETFs

The accuracy of a prediction model is the most important factor. HMM will perform differently based on the number of hidden states. Therefore, to forecast the MBB's prices using HMM, we have to choose the number of states first. In this section, we use the two standard criteria—the Akaike information criterion (AIC) (Akaike, 1974) and the Bayesian information criterion (BIC) (Schwarz, 1978)—to choose the best model among the HMM with two, three or four states. Then we will use HMM with two, three or four states to predict MBB's prices.

5.5.1 Model Selection

Choosing a number of hidden states for HMM is a critical task. We use two common criteria, the AIC and the BIC, to evaluate the performances of HMM with different numbers of states. The criteria measure how well a model fits with given data by evaluating the log likelihood of the model. The likelihood, L, of an HMM is the probability of observations given the model's parameters

$$L = P(O|\lambda).$$

The two criteria are suitable for HMM because in the model training algorithm, the Baum-Welch algorithm, the expectation–maximization (EM) method was used to maximize the log likelihood, $\ln(L)$, of the model. The AIC and BIC are calculated using the following formulas:

$$\text{AIC} = -2\ln(L) + 2k$$
$$\text{BIC} = -2\ln(L) + k\ln(M)$$

where L is the likelihood of the model, $M$is the number of observation points, and k is the number of estimated parameters in the model. In this paper, we assume that the distribution corresponding to each hidden state is a Gaussian distribution. Therefore, the number of parameters, k, is formulated as $k = N^2 + 2N - 1$, where $N$is the number of states used in the HMM. The model with lower AIC or BIC will perform better. In the AIC if more parameters are added to the model, the second term is bigger and the log likelihood of the model increases, which makes the first term smaller. However, when the sample size is large, the log likelihood dominates the penalty term, $2k$. Thus, the AIC will prefer the model with more parameters.

In the BIC, the penalty term (the second term) includes the log of sample size and number of parameters. Thus, similar to the AIC, selecting the "best fit" model using BIC is equivalent to choosing the model with the largest log likelihood.

We use the moving training window that was described in Section 5.4 to make calibrations and plot the results in Figures 13 and 14.The results show that HMM with four states is more accurate han HMM with two or three states.

Figure 13
AIC OF HMM WITH DIFFERENT NUMBER OF STATES

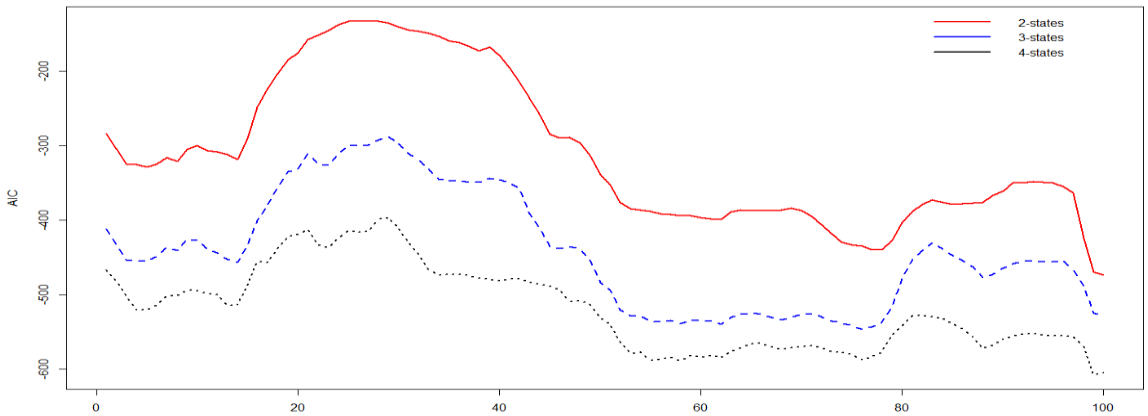
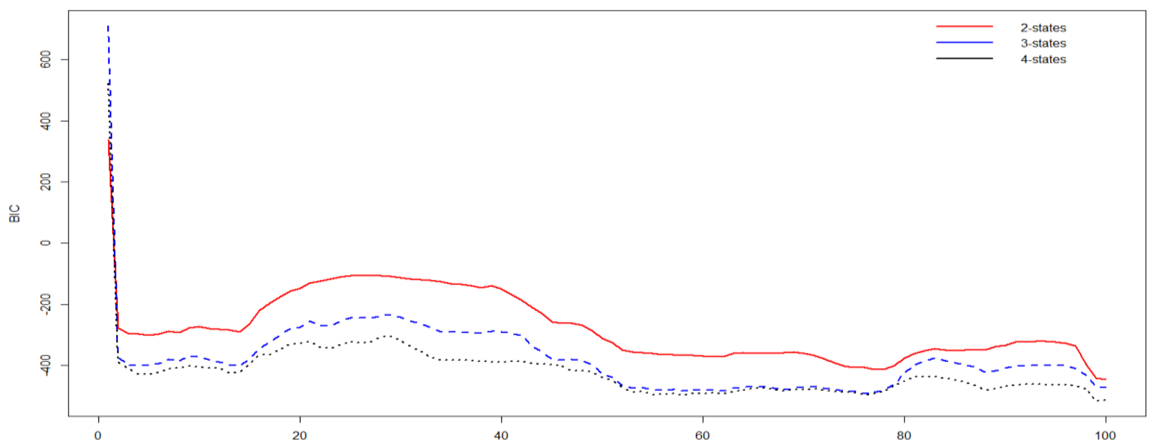


Figure 14
BIC OF HMM WITH DIFFERENT NUMBER OF STATES



In the next section, we will use HMM with two, three and four states to predict MBB price and compare the predicted errors to see if the results are consistent with the AIC or BIC test.

5.5.2 Price Prediction

In this section, we use HMM and historical price data for MBB to predict its future prices and compare the predictions with the market prices. We forecast MBB's prices using HMM with different numbers of states and calculate the mean absolute percentage errors (MAPE) of the estimates,

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

where N is the number of predicted points, M is market price and P is the predicted price of a stock. We use HMM with multiple observation sequences (open, low, high and close price) to predict future close price.

To do this, we divided MBB daily data from March 16, 2007, to October 26, 2016, into three parts: one part for back tracking, one part for model training and one part for model validating. Similar to Section 5.4 we use data from a one-year time period (252 trading days) for a training window, and for convenience, we use one-year data, from October 28, 2015, to October 26, 2016, for model testing.

The prediction process can be divided into three steps:

1. Calibrate HMM's parameters and calculate the likelihood of the model.
2. Find a day in the past that has a similar likelihood to that of the recent day.
3. Use the difference of stock prices on the "similar" day in the past to predict future stock prices.

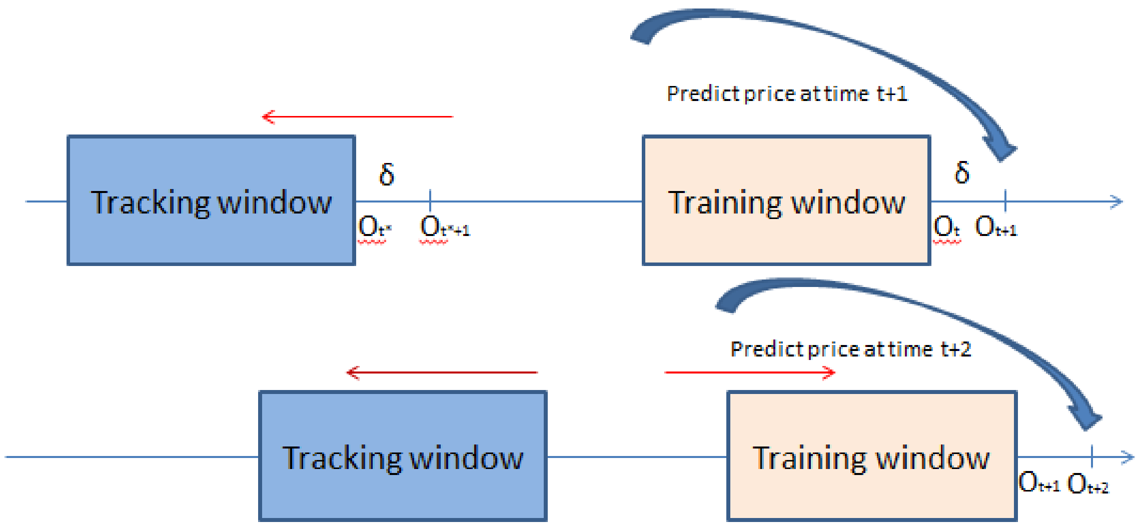
This prediction approach is based on the work of Hassan and Nath (Hassan & Nath, 2005). In order to predict the ETF price for October 28, 2015, we use a one-year data window from October 30, 2014, to October 27, 2015.

In the first step, we use the Baum-Welch algorithm to calibrate the parameters for HMM and use the forward algorithm to calculate the probability of observation for the data set.

In the second step, we move the training data backward by one day, which is called a tracking data window, and use calibrated parameters,λ, to calculate the probability of the tracking date window. We continue to move the tracking window backward until we find the "similar" tracking data window that has very close probability of observation with the training data window (the chosen error is 10^{-6}).

In the last step, we use the difference, δ, between the prices on the end date of the "similar" tracking window and the next consecutive day to predict the price of MBB for October 28, 2015. We assume that the change of MBB's price in October 27, 2016, and October 28, 2016, is equal to δ; thus the predicted price equals the price on October 27, 2016, plus δ. For the second prediction, we move the training window up one day and repeat the process. The prediction process is presented in Figure 15.

Figure 15
PRICE PREDICTION PROCESS



The results of price predictions with two-, three- and four-state HMM are presented in Figures 16-18. We can see from the figures that the HMM with three or four states is more accurate than the HMM with two-states in stock price predictions. We calculate the mean absolute percentage errors, MAPE, of the predictions using these three models and present the results in Table 2.

Figure 16
PRICE PREDICTION FROM OCTOBER 28, 2015, TO OCTOBER 26, 2016, USING TWO-STATE HMM

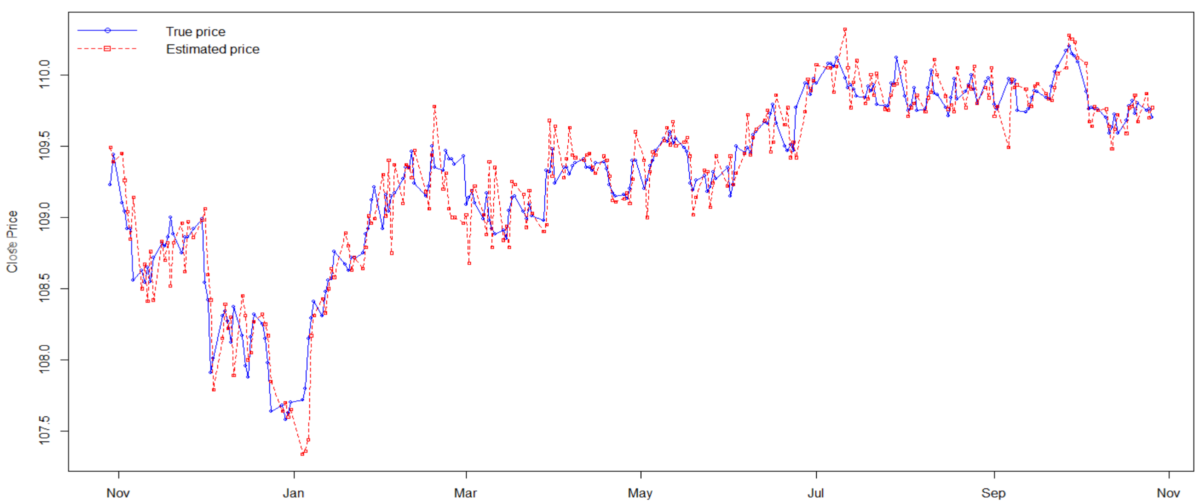


Figure 17

PRICE PREDICTION FROM OCTOBER 28, 2015, TO OCTOBER 26, 2016, USING THREE-STATE HMM

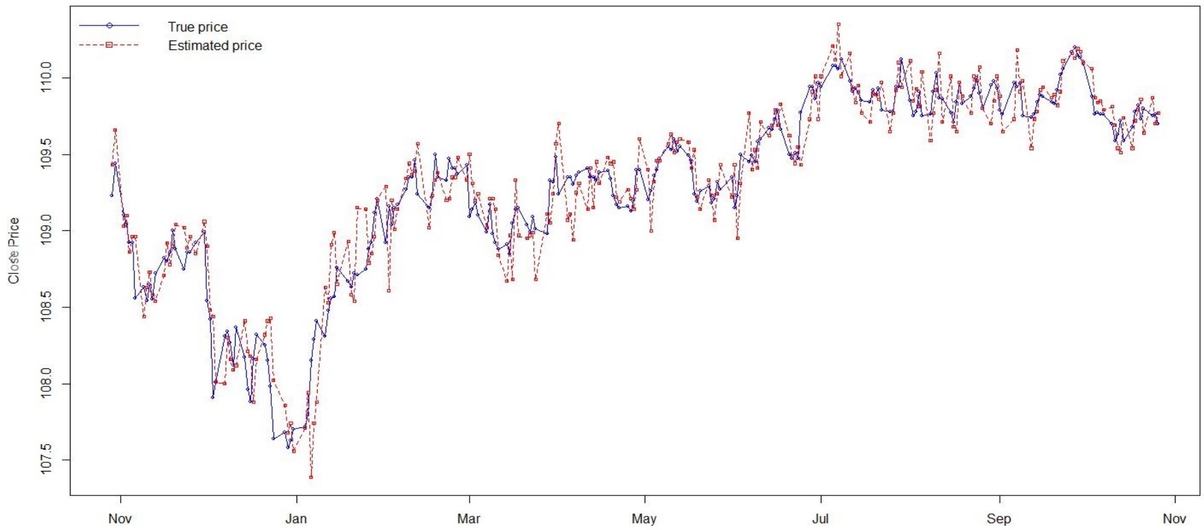


Figure 18

PRICE PREDICTION FROM OCTOBER 28, 2015, TO OCTOBER 26, 2016, USING FOUR-STATE HMM

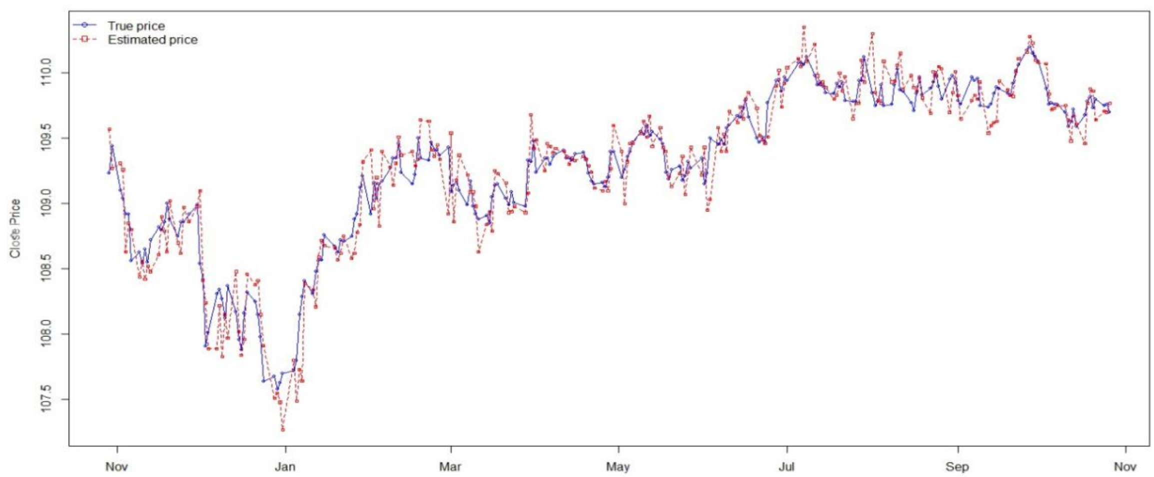


Table 2

PRICE PREDICTION ERRORS USING HMM

Number of states	2	3	4
MAPE	0.001291	0.001316	0.001239

The results in Table 2 show that the four-state HMM gave the smallest error for predicting MBB's prices. The result is consistent with the result in Section 5.5.1: the AIC and BIC results indicated that HMM is most accurate with four states.

5.5.3 Model Validations

We now compare HMM with the standard forecast model, the historical average model for stock return predictions. Among the eight MBSETFs listed in Table 1, we can choose only five that have long enough historical data to implement the HMM. The five MBS ETFs are MBB, GNMA, CMBS, VMBS and MBG. We use daily and weekly prices of the ETFs. Due to the limitations of data, we use a training window of 52 weeks for weekly data (one-year period from November 2, 2015, to October 24, 2016) and training windows of 100 days (from June 7, 2016, to October 26, 2016) and 252 days (from October 28, 2015, to October 26, 2016) for daily data. For convenience, the length of the out-of-sample predictions equals the length of the training window. That means we predict prices for 52 weeks, 100 days and 252 days. For each ETF, we use its four observation sequences—the open, low, high and close prices—to predict the close price in future time.

With a training window size m, the historical average model (HAM) is defined as

$$r_{m+t} = \frac{1}{m+t-1} \sum_{i=1}^{m+t-1} r_i.$$

Since the moving average model works well only for a stationary series, we use r_t is as the total return of stock at time t, $r_t = P_t - P_{t-1}$, where P_t is price at time t of a stock. After using HAM to predict return, we use the formula $P_t = P_{t-1} + r_t$ to obtain the predicted price of the stock.

The MAPE of the estimates, defined in Section 5.5.2, is used to compare the performances of HMM and HAM.

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

Errors of stock price predictions using the two models are presented in Table 3.

Table 3
ERRORSOF OUT-OF-SAMPLE PREDICTED PRICES USING HMM vs. HAM

ETF	Model	100Days (6/7/16-10/26/16)	252 Days (10/28/2015-10/26/2016)	52 Weeks (11/2/2015-10/24/2016)
MBB	HAM	0.0007	0.0009	0.0018
	HMM	0.0005	0.0009	0.0014
GNMA	HAM	0.0013	0.0016	0.0021
	HMM	0.0009	0.0016	0.0020

VMBS MA	0.0009	0.0010	0.0017
HMM	0.0007	0.0011	0.0013
CMBS HAM	0.0021	0.0022	0.0047
HMM	0.0017	0.0023	0.0038
MBG HAM	0.0030	0.0016	0.0027
HMM	0.0027	0.0014	0.0018

Based on the error estimator MAPE, the HMM beats the HAM in predicting the ETFs in most of the cases. In only two cases—forecasting VMBS and CMBS using a 252-day training window—did the HAM outperform the HMM. However, by definition the MAPE measures the mean of absolute percentage errors. It does not tell us the accuracy of the predictions according to the real observation trends.

One of the disadvantages of the moving average method is it is not sensitive to a sudden change of the momentum of the prices. Therefore, it is not the desired candidate to predict trading signal or trends of observations. By comparison, HMM is based on regimes of observation sequences and the probability of jumping from one state to another. Thus, HMM is a good model to capture the regime shifts of observations.

In the next section we investigate the efficiencies of HMM and HAM in predicting trading or trending signals.

5.5.4 Trading Mortgage-Backed Securities

We now apply the two models, HMM and HAM, to trade the MBS ETFs based on the predicted returns. Similar to Section 5.5.3, we will trade these five ETFs—MBB, GNMA, CMBS, VMBS and MBG—in three different periods: 100 days, 252 days for daily trading and 52 weeks for weekly trading. We assume that we buy or sell the ETFs by the close prices without transaction fee, and we buy or sell 100 shares in each trading. The trading strategy is based on the predicted prices in Section 5.5.3. If the model predicts that the price of the ETF goes up, we will buy it and hold until the model predicts that the price goes down. Results of the trading are summarized in Table 4. The table presents the percentage earning in the whole period of trading using HMM and HAM.

Table 4
COMPARISION TRADING MBS ETFs USING HMM AND HAM

ETF	Model	100 Days (6/7/16-10/26/16)	252 Days (10/28/2015-10/26/2016)	52 Weeks (11/2/2015-10/24/2016)
MBB	HAM	0.23%	-0.91%	0.03%
	HMM	1.96%	2.42%	2.12%
GNMA	HAM	0.24%	-0.47%	-0.75%
	HMM	3.98%	4.73%	2.81%
VMBS	HAM	0.28%	-0.86%	-0.02%
	HMM	2.33%	4.39%	2.17%
CMBS	HAM	0.17%	-0.95%	0.29%
	HMM	3.53%	9.62%	5.35%
MBG	HAM	-0.07%	0.00%	0.00%
	HMM	2.18%	7.61%	3.91%

Based on the results in Table 4, we see that HMM captures price changes well. Therefore, it yields a higher return than the moving average model in trading the ETFs. The results show that HMM is a promising model for stock trading.

SECTION 6: CONCLUSIONS

In this paper, we presented an introduction of the hidden Markov model: the historical developments of the model, its basic elements and algorithms, and applications of the model in finance and actuarial fields. We presented simple examples to explain the model's concepts and three main algorithms. We then applied HMM for mortgage-backed securities exchange-traded funds, MBS ETFs. The Baum-Welch algorithm was used to calibrate the model's parameters, and the Viterbi algorithm was used to find the hidden state sequence of the observation data. We used HMM with two, three, and four states to predict the probability of MBB, an iShares MBS ETF, being in the recession state, using one observation sequence and four observation sequences. The results showed that HMM is a good model for forecasting observations' regimes. The more observation sequences we used, the better the predictions were.

Before applying the model for predicting and trading the MBS ETFs, we used the two standard goodness-of-fit tests, the AIC and the BIC, to choose the best HMM (or the number of states for the HMM) for the ETFs. The results showed that HMM with four states is the best model among the two-, three-, or four-state HMM. Based on the model selection results, we used the four-state HMM with multiple observation sequences to predict the ETFs' prices. We compared the HMM with the historical average model, HAM, in predicting the prices. The results showed that the HMM outperformed the HAM for various training windows and out-of-sample time periods. The trading results based on the predicted prices showed that the HAM was worse for trading because it failed to capture the trading signals or the future trends of the underlying access. By comparison, the HMM worked well in trading the MBS ETFs because it predicted precisely the future returns of the funds.

In conclusion, the HMM is a stochastic model that is based on the constant transition matrix of observations' hidden states. It has many applications in finance and actuarial sciences. It is a potential model for predicting extreme regimes of observation sequences and for forecasting prices and trading indexes.

APPENDIX A: RESULTS

Using MBB weekly data and the Baum-Welch algorithm to calibrate parameters for two-state HMM, we get the following results:

- Transition matrix A:

$$A = \begin{bmatrix} 0.9932 & 0.0068 \\ 0.0147 & 0.9853 \end{bmatrix}$$

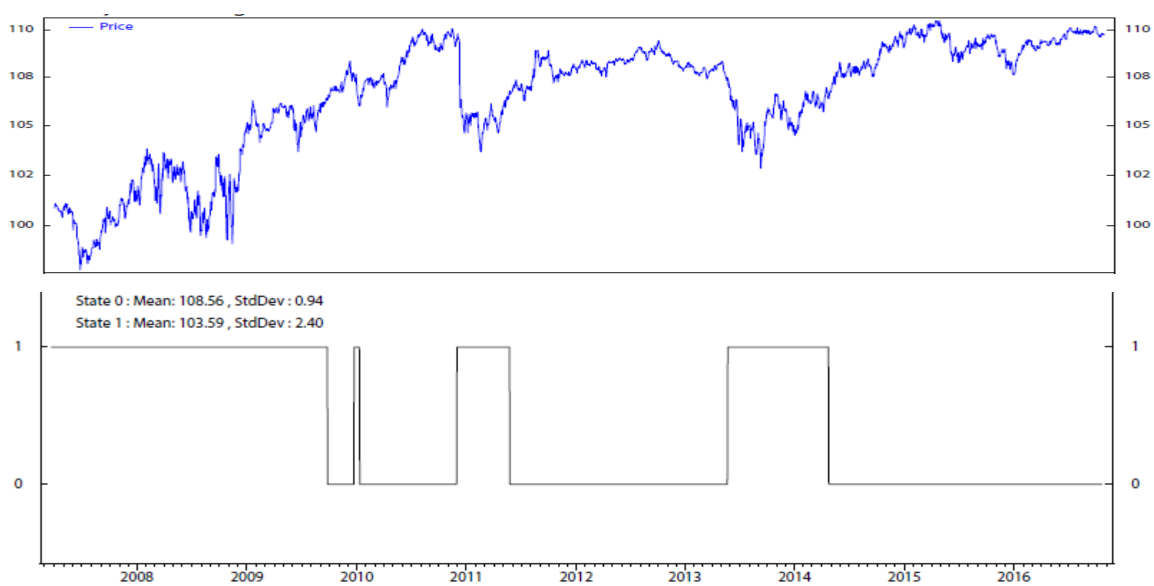
- Means and variances of two observation probability distributions (two normal distributions):

$$\mu_0 = 108.56 \quad \sigma_0 = 0.94$$
$$\mu_1 = 103.59 \quad \sigma_1 = 2.40$$

- Initial probability of being in state 0 or 1 at time 0: $p = (0,1)$.

The simulation of the two-state sequence of the weekly MBB is presented in Figure 1.

Figure 1
FINDING TWO STATES OF MBB WEEKLY DATA USING THE VITERBI ALGORITHM



Using MBB monthly data and the Baum-Welch algorithm to calibrate parameters for two-state HMM, we have the following results:

- Transition matrix A:

$$A = \begin{bmatrix} 0.9687 & 0.0313 \\ 0.0619 & 0.9381 \end{bmatrix}$$

- Means and variances of two observation probability distributions for two states:

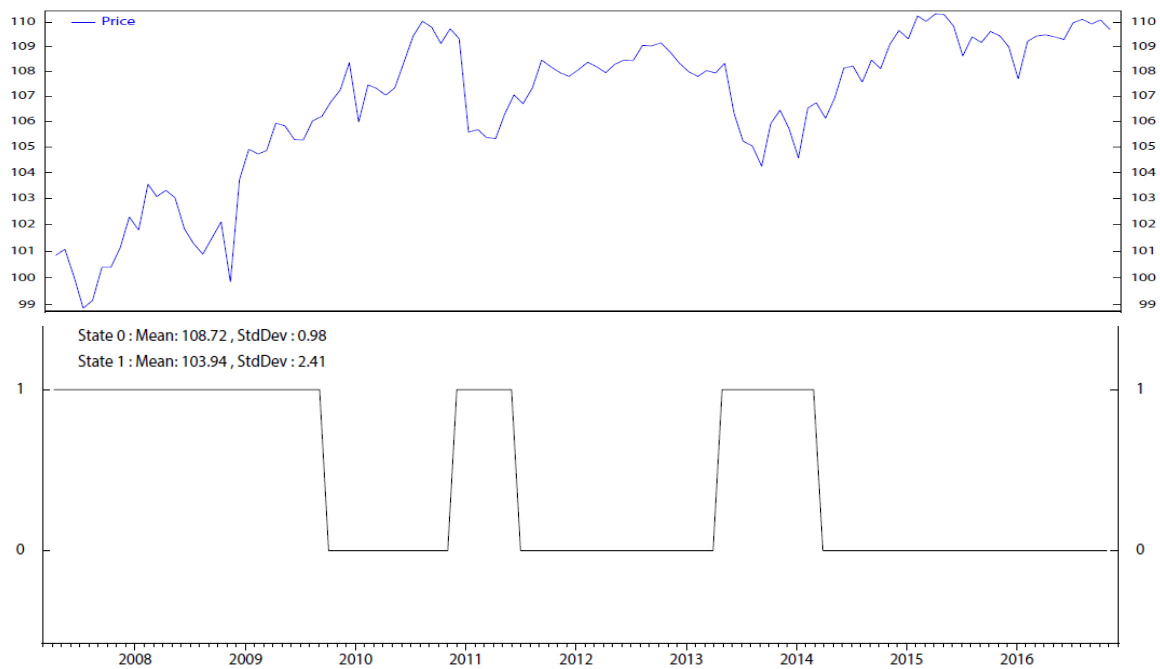
$$\mu_1 = 108.72 \quad \sigma_1 = 0.98$$
$$\mu_2 = 103.94 \quad \sigma_2 = 2.41$$

- Initial probability of being in state 0 or 1 at time 0: $p = (0,1)$.

The simulation of the two-state sequence of the monthly MBB is presented in Figure 2.

Figure 2

FINDING HIDDEN STATES OF MBB MONTHLY DATA USING TWO-STATE HMM



APPENDIX B: R CODE

The R code file is available at this link:

https://www.soa.org/files/research/projects/2017-hidden-markov-model-codes.zip

REFERENCES

The Economist. 2007. “CSI: Credit Crunch." October 18.

Akaike, H. 1974. "A New Look at the Statistical Model Identification." *IEEE Transactions on Automatic Control* 19 (6):716-723.

Ang, A., and G. Bekaert. 2002. "International Asset Allocaion with Regime Shifts." *The Review of Financial Studies* 15 (4): 1137-1187.

Baggenstoss, P. M. 2001. "A Modified Baum-Welch Algorithm for Hidden Markov Models with Multiple Observation Spaces." *IEEE Transactions on Speech and Audio Processing* 9 (4): 411-416.

Baum, L. E., and J. A. Egon. 1967. "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Process and to a Model for Ecology. *Bulletin of the American Mathematical Society* 73 (3): 360-363.

Baum, L. E., and G. R. Sell. 1968. "Growth Functions for Transformations on Manifolds." *Pacific Journal of Mathematics* 27 (2): 211-227.

Baum, L. E., T. Petrie, G. Soules, and N. Weiss. 1970. "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains." *The Annals of Mathematical Statistics* 41 (1): 164-171.

Baum, L., and T. Petrie. 1966. "Statistical Inference for Probabilistic Functions of Finite State Markov Chains." *The Annals of Mathematical Statistics* 37 (6): 1554-1563.

Bonnie, B. 1998. "The Dynamic Impact of Macroeconomic Aggregates on Housing Prices and Stock of Houses: A National and Regional Analysis." *Journal of Real Estate Finance and Economics* 17: 1979-1987.

Chen, M. 2016. Hidden Markov Model Toolbox (HMM). Computer program. MathWorks. https://www.mathworks.com/matlabcentral/fileexchange/55866-hidden-markov-model-toolbox--hmm-.

Calhoun, C. A., and Y. Deng. 2002. "A Dynamic Analysis of Fixed- and Adjustable-Rate Mortgage Terminations." *The Journal of Real Estate Finance and Economics* 24 (1): 9-33.

Dymarski, P., ed. 2011. *Hidden Markov Models, Theory and Applications.* Rijeka, Croatia: InTech.

Elliott, J. R., and C. A. Wilson. 2007. "The Term Structure of Interest Rates in a Hidden Markov Setting." *Hidden Markov Models in Finance.* Springer.

Erlwein, C., and R. Mamon. 2009. "An Online Estimation Scheme for a Hull-White Model with HMM-driven Parameters."*Statistical Methods and Applications* 18 (1): 87-10. Springer.

Forney, G. D. 1973. "The Viterbi Algorithm." *Proceedings of the IEEE* 61 (3): 268-278.

Garda, R., and P. Parron. 1996. "An Analysis of the Real Interest Rate Under Regime Shifts."*The Review of Economics and Statistics* 78 (1): 111-125.

Ghahramani, Z. 2001. "An Introduction to Hidden Markov Models and Bayesian Networks." *International Journal of Pattern and Recognition and Artificial Intelligence* 15 (1): 9-42.

Guidolin, M., and A. Timmermann. 2005. "Asset Allocation Under Multivariate Regime." Federal Reserve Bank of St. Louis Working Paper.

Hamilton, J. D. 1989. "A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle." *Econometrica* 57 (2): 357-384.

Harte, D. 2016. Hidden Markov: Hidden Markov Models, R package version 1.8-7. Statistics Research Associates, Wellington.

Hassan, M. R., and B. Nath. 2005. "Stock Market Forecasting Using Hidden Markov Model: A New Approach." *Proceedings of the 2005 5th International Conference on Intelligent Systems Design and Applications*.

Idvall, P., and C. Jonsson. 2008. "Algorithmic Trading: Hidden Markov Models on Foreign Exchange Data. Master's thesis, Linköping University.

Kau, J. K., D. C. Keenan, W. J. Muller, and J. F. Epperson. 1992. "A Generalized Valuation Model for Fixed-Rate Residential Mortgages." *Journal of Money, Credit and Banking* 24 (3): 279-299.

Kavitha, G., A. Udhayakumar, and D. Nagarajan. 2013. "Stock Market Trend Analysis Using Hidden Markov Models." *International Journal of Computer Science and Information Security* 11 (10): 103.

Kritzman, M., S. Page, and D. Turkington. 2012. "Regime Shifts: Implications for Dynamic Strategies." *Financial Analysts Journal* 68 (3). https://papers.ssrn.com/sol3/papers2.cfm?abstract_id=2066848.

Lajos, J. 2011. "Computer Modeling Using Hidden Markov Model Approach Applied to the Financial Markets." PhD diss., Oklahoma State University.

Levinson, S. E., L. R. Rabiner, and M. M. Sondhi. 1983. "An Introduction to the Application of the Theory of Probabilistic Functions of Markov Process to Automatic Speech Recognition." *Bell System Technical Journal* 62 (4): 1035–1074.

Li, X. P., M. Parizeau, and R. Plamondon. 2000. "Training Hidden Markov Models with Multiple Observations: A Combinatorial Method." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22: 371–377.

Mamon, R., and R. J. Elliott, eds. 2007. *Hidden Markov Models in Finance.* Vol. 104. New York: Springer.

Nguyen, N. A., and D. Nguyen. 2015. "Hidden Markov Model for Stock Selection." *Risks* 3(4): 455-473.

Nguyen, N. T. 2014. "Probabilistic Methods in Estimation and Prediction of Financial Models." PhD diss., Florida State University.

Nneji, O., C. Brooks, and C. W. R. Ward. 2013. "House Price Dynamics and Their Reaction to Macroeconomic Changes." *Economic Modelling* 32 (May): 172-178.

Rabiner, L. R. 1989. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." *Proceedings of the IEEE* 77 (2): 257-286.

Sand, A., C. N. Pedersen, and T. Mailund. (n.d.). HMMlib: A C++ Library for General Hidden Markov Models Exploiting Modern CPUs. Computer program.

Schwarz, G. 1978. "Estimating the Dimension of a Model." *The Annals of Statistics* 6 (2): 461-464.

Viterbi, A. J. 1967. "Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm." *IEEE Transactions on Information Theory* 13 (2): 260-269.

Zenios, S. A. 1993. "A Model for Portfolio Management with Mortgage-Backed Securities." *Annals of Operations Research* 43 (6): 337-356.

Zhang, Y. 2004. "Prediction of Financial Time Series with Hidden Markov Model." Simmon Fraser University.

Zhu, Y. A. and J. Cheng. 2013. "Using Hidden Markov Model to Detect Macro-economic Risk Level." *Review of Integrative Business & Economics Research* 2 (1): 238-249.