An Investigation of Hidden Markov Model with Partially Missing Observations

Erding Liao, supervised by Prof. Ery Arias-Castro
UCSD Department of Mathematics

April, 2021

Contents

List of Symbols

C the (hidden) chain of Markov Model

(C_i, c_i) C_i is the i-th hidden component in C, and c_i represents the exact hidden state occupied by the chain at time $t = i$

X the observation chain of Hidden Markov Model

X^{-t} the observation chain with missing value at time t

D set of time t with missing values

X^{-D} the observation chain with missing value in the set D

(X_i, x_i) X_i is the i-th observed component in X, and x_i represents the exact observation at time $t = i$

A transition matrix of Markov Model

a_{ij} the (i, j) element of matrix A: transition probability from state i to state j

B emission matrix of Hidden Markov Model

$b_j(k)$ the (j, k) element of matrix B: emission probability of observation k under the condition of hidden state j

$B(x_t)$ the diagonal emission matrix

$B_{ii}(x_t)$ the (i, i) elment of matrix $B(x_t)$: emission probability of observation x_t under the condition of hidden state j, which is $b_i(x_t)$

λ the vector of initial probabilities of Markov Model

λ_i the i-th element of λ: probability of starting the Markov Chain with state i

Θ true parameters of Hidden Markov Model (A, B, λ)

$\Theta^{(0)}$ the current set of parameters in E-step of EM algorithm

$\Theta^{(1)}$ the next generation of parameters produced by M-step of EM algorithm

$\hat{\Theta}$ output of EM algorithm as the approximation on true parameters Θ

N number of possible hidden states; number of parameters

M number of possible observations

$\alpha_t(i)$ forward probabilities up to time t with $C_t = i$

$\beta_t(i)$ backward probabilities up to time t with $C_t = i$

$\boldsymbol{\alpha}_t$ row vector of forward probabilities with respect to all i

$\boldsymbol{\beta}_t$ row vector of backward probabilities with respect to all i

T length of observation/hidden chain

$L_T(X)$ likelihood of given observation chain $X = X_1 = x_1, ..., X_T = x_T$

$L_T^{-t^*}(X)$ ignorable likelihood of X^{-t^*}

$L_T^{-D}(X)$ ignorable likelihood of X^{-D}

w_t sum of elements in $\boldsymbol{\alpha}_t$

ϕ_t $\boldsymbol{\alpha}_t$ normalized by w_t

$\delta_t(i)$ probability of the most likely previous transition chain $c_{t-1} \rightarrow$c$_t$ in Viterbi algorithm

$\psi_t(i)$ c_{t-1} that corresponds with $\delta_t(i)$

$\sigma_t(i)$ probability of $c_t = i$ at time t under the condition of given observation chain X

$\epsilon_t(i, j)$ transition probability from $c_t = i$ to $c_{t+1} = j$ under the condition of given observation chain X

d step size of imputation in Viterbi section of EM-Viterbi algorithm

$b(F)$ penalty term in information criterion

M_{BIC} BIC score of the approximated model M

$\mathbf{1}$ row vector of 1

1 Introduction

Hidden Markov Models (also abbreviated as HMMs) are widely used as flexible tools for modeling and approximating time series. During recent decades, HMMs are efficiently applied in artificial intelligence, finance (José G.Dias, 2015 [5]), and biology (Sean R. Eddy 1998 [7]). There have been highly developed algorithms of HMMs with full observations of time series, while it is also important for speech recognition to discuss on missing data caused by failure of recording. This paper aims to extend the investigation of approximation and prediction with missing observations[20] to modeling HMMs.

In this paper, we will investigate HMMs under the situation when one or more observations are missing and modify existing algorithms of full observations to be compatible with the missing data. In Section 2, we discuss the likelihood of the observation when data are missing and provide prediction of hidden states under different conditions, given that true parameters are known. In Section 3, we modify the EM algorithm with Viterbi to be better applied. Section 4 includes some approaches of model selection applied to HMMs. Lastly, Section 5 contains our experiments and conclusion about the performance of the modified algorithm on a real dataset.

1.1 HMM Basics: Formulation

Markov Model represents a memoryless transition, in which the current state only depends on one state before:

$$P(C_t = c_j | C_1 = c_1, ..., C_{t-1} = c_i) = P(C_t = c_j | C_{t-1} = c_i)$$

For any Markov Model with N possible states, we can thus formulate a transition matrix A where $P(C_t = j | C_{t-1} = i) = a_{ij}$. Each element on ith row and jth column marks the probability of transition from state i to state j. One property of Markov Model is that, if the chain of states is long enough, then in the long-term the proportion of occupation time of states i (the time that the chain is in the specific state) is the "stationary probability" λ_i, and for all states we can form the vector λ. Since the Markov chain is stationary in the long term, λ_i also indicates the probability of starting at i in the chain.

Based on Markov Model, a Hidden Markov Model forms by hidden states with Markov property and observation states, which is uniquely determined by current hidden states. Therefore, hidden states naturally have their transition matrix A, and λ_i of Markov Model is used here as the initial probability $\lambda_i = P(C_1 = c_i)$

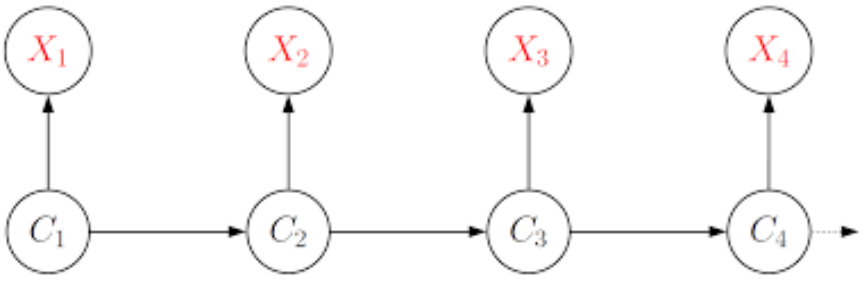


Figure 1: Probability Graph Model of Hidden Markov Distribution

Because the current observation X_t only depends on the current state C_t (C_t is known):

$$P(X_t = k | C_1 = c_1, ..., C_t = j) = P(X_t = k | C_t = j) = b_j(k),$$

a Hidden Markov Model would also have an emission matrix of M possible observations

$$B = \big[b_j(k)\big]_{N\times M}$$

Usually, a completely formulated HMM would start with parameters (A, B, λ), and the complexity is mainly from the number of hidden states N, number of observations M. In practical situations, it is to start with estimating (A, B, λ) through EM-algorithm. This method of approximation is discussed in Section 3.

In this paper we always assume the HMM to be stationary: if the chain of hidden states start at $t = 0$ with an initial probability vector $\lambda = (\lambda_1, ..., \lambda_N)$ that is also its stationary probability vector, where $\lambda_i = P(C_1 = c_i)$, then $\lambda A = \lambda$. This assumption makes sure that the distribution of the HMM is consistent from $t = 1$ to $t = T$, and the vector of initial probabilities does not change when we are starting in the middle of the observation chain. It is essential for implementations in Section 3.2.1, where we apply an iterative methods along the time series.

1.2 Forward/Backward-probability

In a Hidden Markov Model with given parameters (A, B, λ) and observation chain $X = \{X_1 = x_1, X_2 = x_2, ..., X_T = x_T\}$, one fast and efficient approach to find the observation probability $P(X)$ is by defining forward and backward probabilities. The forward probability $\alpha_t(i)$ represents the probability of having observations from x_1 forward to x_t and one hidden state $C_t = i$; and the backward probability $\beta_t(i)$ marks the conditional probability that given the hidden state $C_t = i$, we have observations from tail x_t backward to x_{t+1}:

$$\begin{aligned}\alpha_t(i) &= P(X_1 = x_1, ..., X_t = x_t, C_t = i)\\ &= b_i(x_t)\sum_{j=1}^{N}\alpha_{t-1}(j)a_{ji} \qquad (1)\\ \alpha_1(i) &= \lambda_i b_i(x_1)\end{aligned}$$

$$\begin{aligned}\beta_t(i) &= P(X_{t+1} = x_{t+1}, ..., X_T = x_T|C_t = i)\\ &= \sum_{j=1}^{N} a_{ij}b_j(x_{t+1})\beta_{t+1}(j) \qquad (2)\\ \beta_T(i) &= 1\end{aligned}$$

(Zucchini et al. 2017 [20]) and for all possible hidden states i in $(\alpha_t(i), \beta_t(i))$ at time t, we can construct vectors of forward/backward probabilities, marked by $\boldsymbol{\alpha_t}$ and $\boldsymbol{\beta_t}$:

$$\begin{cases} \boldsymbol{\alpha}_1 = \lambda, \\ \boldsymbol{\alpha}_t = \lambda B(x_1)AB(x_2)...AB(x_t), \\ \boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t AB(x_{t+1}), \end{cases} \tag{3}$$

$$\begin{cases} \boldsymbol{\beta}_T = \mathbf{1}^\top, \\ \boldsymbol{\beta}_t = B(x_{t+1})AB(x_{t+2})...AB(x_T)\mathbf{1}^\top, \\ \boldsymbol{\beta}_{t-1} = B(x_t)A\boldsymbol{\beta}_t, \end{cases} \tag{4}$$

where $B(x_t)$ is the diagonal matrix containing all emission probabilities of X_t given different hidden states, deriving from $b_i(x_t)$ of the original emission matrix B:

$$B_{ii}(x_t) = P(X_t = x_t \mid C_t = i) = b_i(x_t)$$

A simple but helpful lemma of forward/backward probability is that:

$$\alpha_t(i)\beta_t(i) = P(X_1 = x_1, ..., X_T = x_T, C_t = i) \tag{5}$$

which represents the probability with respect to the observation chain and only one hidden state at time t. The lemma is frequently applied in any maximization of probabilities about observations (Zucchini et al. [20]).

1.3 HMM in Speech Recognition

Speech recognition works on the decoding of input audios by fitted parameters. The raw audio files are transformed through digital signal processing algorithms and feature extraction such as Mel-scale Frequency Cepstral Coefficient (Dave et al. [3]) to multivariate acoustic vectors containing information of pronunciation. HMMs in speech recognition will first approximate parameters from training vectors and decode the most possible phonemes (basic components of words) for new inputs, and they are then formed to words and sentences through linguistic models.

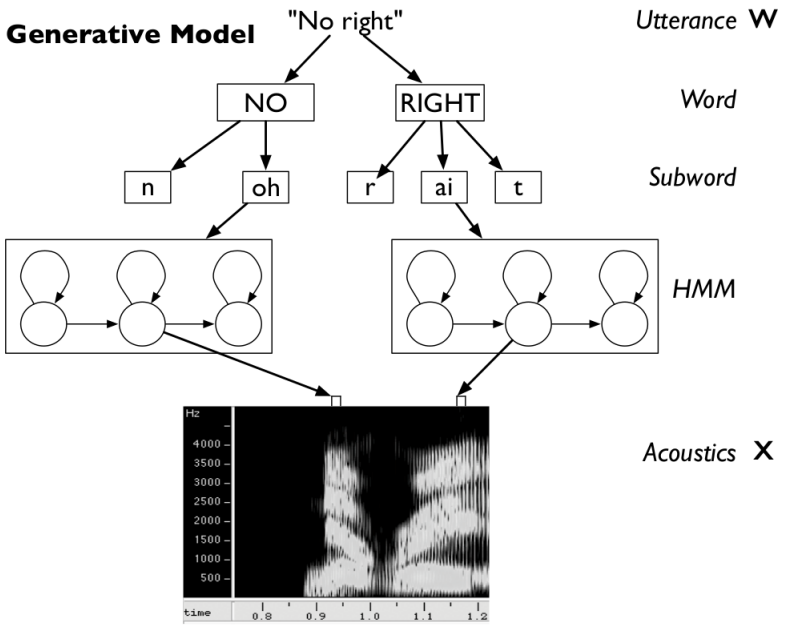


Figure 2: The formation of speech recognition system[11]

Usually in speech recognition, the HMM is composed by a discrete transition matrix of finite possible hidden states and an emission model of continuous multivariate vectors, resulting a complicated HMM-GMM model, where hidden chains (categorical phonemes) are still constructed under Markov Model, but emissions of acoustic data as continuous vectors are determined through Gaussian Mixture Models (Reynolds et al. [16]).

1.4 HMM in Finance

Another important application of HMM is in Market Timing: similarly to speech recognition, the entire system is also constructed to a nested or hierarchical HMM. Hidden states are decided uniquely in every hierarchy of the model: at the first floor, there are usually five hidden states on the general conditions of the market: Strong Bear(SB) when the price is falling or going to fall vastly, Weak Bear(WB), Strong Bull(SU) when the price is rising or going to rise vastly, Weak Bull(WU), and Random Walk(RW) when there is no apparent trend of rising/falling, while at the bottom floor, the hidden states may be detailed to information exactly from previous price, such as interest rate.

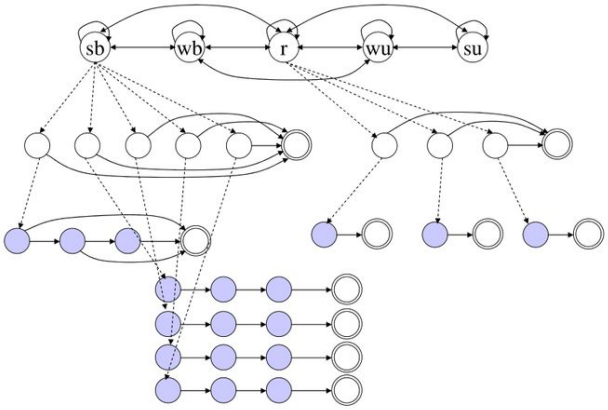


Figure 3: The hierarchial HMM of quantitative market timing[8]

From Figure 3, it is clear that in the system of hierarchical HMM, each pair of floors $\{f-1, f\}$ are constructed in HMM, and states at the lower floors are emissions of states at higher floors. If states at floor $f-1$ are still categorical data (like stock states of either rising, falling, or random walking as we mentioned before) while starting at f states are related to some quantitative data of continuous variables such as price or interest rate, then the model between $\{f-1, f\}$ would be from categorical data to quantitative data and therefore a HMM-GMM model. However, when going forward to $f+1$, continuous variables at f must be transformed into categorical data in order to work as hidden states in HMM of the pair $\{f, f+1\}$, usually by taking floor/ceiling integers.

Since each pair of floors in hierarchical model has the structure of HMM, while the thesis is not focused on single HMM, experiments and discussion in Section 5 will mainly focus on the model of one pair of hidden-observation relation between stock states and price. A very simple example is provided below for better illustration of HMM structure:

Suppose we have 3 stock states: Random Walk(RW), Bear(B), and Bull(U), and probabilities of either rising/falling price for each states are given by:

States	RW	B	U
Rise	0.5	0.25	0.8
Fall	0.5	0.75	0.2

and we know that:

- if the price is randomly walking, the probability to stay is 0.3, the probability of transition to bear market is 0.1, and the probability of transition to bull market is 0.6.

- if there is a bear market, the probability to stay bear is 0.1, probability of transition to random walk is 0.4, and the probability of transition to bull market is 0.5.
- if there is a bull market, the probability to stay is 0.7, the probability of transition to bear market is 0.2, and the probability of transition to random walk is 0.1.

Also, the starting state $\{RW, B, U\}$ is determined by corresponding probabilities $\{0.35, 0.1, 0.55\}$. Then the transition matrix A, emission matrix B, and initial vector λ are given by:

$$A = \begin{bmatrix} 0.3 & 0.1 & 0.6 \\ 0.4 & 0.1 & 0.5 \\ 0.1 & 0.2 & 0.7 \end{bmatrix}, \ B = \begin{bmatrix} 0.5 & 0.5 \\ 0.25 & 0.75 \\ 0.8 & 0.2 \end{bmatrix}, \ \lambda = \begin{bmatrix} 0.35, & 0.1, & 0.55 \end{bmatrix}$$

To simplify the notation, usually the set of hidden states is marked by numeric hidden set $I = \{RW, B, U\} = \{1, 2, 3\}$. For example, the transition probability a_{12} represents $P(c_t = B | c_{t-1} = U)$

2 Inference for Hidden Markov Models

2.1 Likelihood with Full Observation Chain

For a stationary HMM with given true parameters (A, B, λ) and independent observation chain $X = \{X_1 = x_1, ..., X_T = x_T\}$, the likelihood is defined as $L_T(X) = P(X)$. Since this probability only considers about observations, ignoring hidden states at any time t. Therefore, from the lemma in Equation 5 the likelihood is formulated as

$$\begin{aligned} L_T(X) &= \lambda AB(x_1)AB(x_2)...AB(x_T)\mathbf{1}^\top \\ &= \sum_{i=1}^{N} P(X_1 = x_1, ..., X_T = x_T, C_t = i) \\ &= \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) = \boldsymbol{\alpha_t}\boldsymbol{\beta_t}^\top \end{aligned} \tag{6}$$

Notice that the formulation of the likelihood applied the same structure of dynamic processing as vector-structured forward probability in Equation 3, and thus we can reconstruct the likelihood function L_T in the form of $\boldsymbol{\alpha_T}$:

$$\boldsymbol{\alpha_1} = \lambda AB(x_1), \ \boldsymbol{\alpha_t} = \boldsymbol{\alpha_{t-1}} AB(x_t), \ L_T(X) = \boldsymbol{\alpha_T}\mathbf{1}^\top,$$

and thus similarly to forward probabilities, the likelihood works recursively from the head of the observation chain.

2.2 Prediction of Hidden States

One application of $L_T(X)$ is to find the most possible hidden chains with maximized probability given true parameters (A, B, λ) and observations. Instead of naively solving all possible combinations of hidden chain C, which is highly time-consuming even with short chains, we can apply the forward/backward probabilities in the formula of likelihood.

For each time t, the most likely hidden state c_t^* given the condition of observation chain X is:

$$c_t^* = \arg\max_{1 \geq i \geq N} \left[\frac{P(c_t = i,\ X)}{P(X)} \right] \tag{7}$$

By the definition of forward/backward probabilities:

$$\begin{aligned} c_t^* &= \arg\max_{1 \geq i \geq N} \left[\frac{\alpha_t(i)\beta_t(i)}{\sum_{k=1}^{N} \alpha_t(k)\beta_t(k)} \right] \\ &= \arg\max_{1 \geq i \geq N} \left[\frac{\alpha_t(i)\beta_t(i)}{L_T(X)} \right] \end{aligned} \tag{8}$$

Therefore, iterative calculation on $\alpha_t(i)$ and $\beta_t(i)$ would also simplify the time complexity to $O(TN^2)$. One shortcoming is that, this formulation is confined on optimizing the hidden state c_t with respect to the single time point t, but the local maximizer may not be global, and sometimes the transition probability a_{ij} of two consecutive local maximizer $(c_t = i, c_{t+1} = j)$ may be 0, resulting in some impossible approximation in larger time scales. On the contrary, Viterbi algorithm provide with the solution considering on the entire hidden sequence. More information is mentioned in Section 2.4

Moreover, the estimation by directly maximization is problematic with the likelihood itself due to underflow from consecutive multiplication of $\boldsymbol{\alpha_t}$: since forward probabilities are always between 0 and 1, the likelihood of long observation chain would ultimately fall into 0 for discrete HMMs and infinite for continuous HMMs (Leroux et al. 1992 [10]).

A possible solution raised by Zucchini [20] is to scale the vector of forward probability $\boldsymbol{\alpha_t}$ by the sum of its element, so that scaled elements add to 1:

$$\begin{aligned} \phi_t &= \frac{\boldsymbol{\alpha}_t}{w_t}, \phi_0 = \lambda \\ w_t &= \sum_{i=1}^{N} \alpha_t(i), w_0 = 1 \end{aligned}$$

Therefore,

$$
\begin{aligned}
L_T(X) = w_T &= \prod_{t=1}^{T} \left[\frac{w_t}{w_{t-1}} \right] \\
&= \prod_{t=1}^{T} \phi_{t-1} AB(x_t) \mathbf{1}^\top,
\end{aligned}
\tag{9}
$$

There may be more than one scaling method: one can also apply to other form of scaling that is more practical for the situation, such as the Z-score or dividing by the maximum value of $\boldsymbol{\alpha_t}$.

2.3 Likelihood with Missing Data

2.3.1 Ignorable Likelihood

In some real practice such as speech recognition, it is possible that the observation chain have one or more missing data in the middle. It is possible that the microphone may fail during the recording as some tense noise is ignored by MFCC preprocessing, or in the term of finance there are missing history stock price simply due to technical errors, and thus some "empty data" are left in the observation.

For the HMM with parameter (A, B, λ) and observation X^{-t^*} with missing data at time t^*, the likelihood can be developed by simply skip time t:

$$
L_T^{-t^*}(X) = \lambda AB(x_1)...B(x_{t^*-1})A^2B(x_{t^*+1})...
$$

and the corresponding scaled likelihood is:

$$
\begin{aligned}
L_T^{-t^*}(X) &= \prod_{t=1}^{t^*-1} \left[\frac{w_t}{w_{t-1}} \right] \cdot \prod_{t=t^*+2}^{T} \left[\frac{w_t}{w_{t-1}} \right] \\
&= \prod_{t=1, t\neq t^*, t-1\neq t^*}^{T} \phi_{t-1} AB(x_t) \mathbf{1}^\top,
\end{aligned}
\tag{10}
$$

If instead we have missing values during a time interval $D = [t_1, t_2]$, it is straight forward to modify from $L_T^{-t^*}(X)$ to $L_T^{-D}(X)$ The likelihood $L_T^{-t^*}(X)$ is named as "ignorable likelihood" if we assume that the missing data at time t has little negative influence and can be ignored (Little et al. 2008 [12]). This likelihood is frequently used in many practical cases where missing values in the observation are scattered in the chain, especially when the distribution is truly discrete (such as protein sequence, Wu et al. [19]).

2.3.2 With Chain of Missing Observations

The aforementioned likelihood works efficiently with scattered missing values. However, similarly to the problem of optimization on a single time t for every c_t, when there is a chain D of missing values on the interval $[t_1, t_2]$ of length K, it is possible that the igorable likelihood would also ignore the influence of the missingness: since every hidden states are predicted individually in equation 7, it is sometimes possible that $\{c_{t_1-1}, c_{t_2+1}\}$ predicted from $L_T^{-D}(X)$ have their transition probability $P(c_{t_2+1}|c_{t_1-1}) = 0$. One possible modification is to reconstruct the original two chains $(X = \{x_1, x_2, ...\}, C = \{c_1.c_2, ...\})$ into $(X^{'} = \{(\ x_1, x_2), (x_3, x_4), ...\}, C' = \{(c_1, c_2), (c_3, c_4), ...\})$ and therefore with modified parameters $(A^{'}, B^{'}, \lambda^{'})$ and $\alpha^{'}$, by sacrificing the time complexity: the modification is able to decrease the length of missing chain by half and thus increase the accuracy, but by the definition of forward probability $\boldsymbol{\alpha_t}$ and recursive likelihood $L_T(X)$, the original time complexity of full or singly-missing observation chain for the HMM with N possible hidden states is $O(TN^2)$, while a pair-reconstructed HMM would increase the time to $O(TN^4)$.

Another possible solution is to partially apply forecasting on the chain of missing values through forward or backward probabilities before the likelihood. In fact, this approach is the same for both $\boldsymbol{\alpha_t}$ and $\boldsymbol{\beta_t}$, and we here only put our discussion in the case of $\boldsymbol{\alpha_t}$.

Suppose the time interval of missing values of a HMM is $[t_1, t_2]$, then it is possible to start with X_{t_1-1}, which is the last known observation before the missing chain D. Recall the definition of forward probability in Equation 1, we can derive $P(X_1 = x_1, ..., X_{t_1} = x_{t_1})$ from $\alpha_{t_1}(i) = P(X_1 = x_1, ..., X_{t_1} = x_{t_1}, c_{t_1} = i)$ by simply ignoring the hidden states at time t_1 through the summation, and thus:

$$\begin{aligned} x_{t_1}^{'} &= \arg\max_{x_{t_1}} \sum_{i=1}^{N} \alpha_{t_1}(i) \\ &= \arg\max_{x_{t_1}} \sum_{i=1}^{N} \left[b_i(x_{t_1}) \sum_{j=1}^{N} \alpha_{t_1-1}(j) a_{ji} \right] \end{aligned} \tag{11}$$

It takes $O(MN^2)$ to find the most likely observation at time t_1, and for the derived observation chain $X^{'} = \{x_t, ..., x_{t_1-1}, x_{t_1}^{'}, ..., x_{t_2}^{'}, x_{t_2+1}, ..., x_T\}$, the pseudo-likelihood is correspondingly:

$$L_T^{'}(X) = \lambda \prod_{t=1}^{t_1-1} AB(x_t) \cdot \prod_{t=t_1}^{t_2} AB(x_t^{'}) \cdot \prod_{t=t_2+1}^{T} AB(x_t) \mathbf{1}^{\top}, \tag{12}$$

and the scaled likelihood should still have its original formulation, but with respect to derived observation $X^{'}$.

Comparing with the ignorable likelihood, in the pseudo-likelihood we first derive $x_t^{'}$ through the forward probability, which takes all possible transition probabilities and filters transitions with probability $a_{ij} = 0$ by finding the maximizer. Therefore, it takes all observations into account, including missingness, and we can directly apply the pseudo-likelihood alternatively to predict the hidden chain, avoiding any zero probabilities in the missing chain. The idea of finding the hidden chain through maximizing the pseudo-likelihood is to confine the ignorable likelihood through the imputation on the missingness and solve the conflict between local optimization (on known observations) and the optimization with respect to the entire chain.

2.4 Prediction by Viterbi Algorithm

In order to generally avoid the problem of local optimization, Viterbi algorithm is introduced to take the full observation chain as a whole. Rather than directly analyzing all possible hidden chains, based on dynamic programming, it simplified the question through dividing the entire hidden sequence into multiple sub-sequences. Usually each subsequence marks one transition $c_t \to c_{t+1}$, and thus current subsequences are only related to future parts. Starting with the first state, the optimization is confined to current states, and the outcome would restrict the possibility of the future states in their maximization. Therefore, this iterative method would effectively reduce the complexity of the entire calculation.

In the context of Hidden Markov Models, Viterbi algorithm constructs two components of subsequences: $(\delta_t(i), \Psi_t(i))$, where $\delta_t(i)$ is the chain of the most possible previous transition chain from c_1 to c_{t-1} with $c_t = i$:

$$\delta_t(i) = \max_{(c_1,\dots,c_{t-1})} P(c_t = i, c_1, ..., c_{t-1},\ x_1, ..., x_t) \tag{13}$$

with its iterative formulation:

$$\begin{aligned} \delta_1(i) &= \lambda_i b_i(x_1) \\ \delta_{t+1}(i) &= \max_{(c_1,\dots,c_t)} P(c_{t+1} = i, c_1, ..., c_t,\ x_1, ..., x_{t+1}) \\ &= \max_{1\geq j\geq N} \left[\delta_t(j) a_{ji}\right] b_i(x_{t+1}), \end{aligned} \tag{14}$$

and $\Psi_t(i)$ is the state c_{t-1} in the most possible transition chain $c_{t-1} \to c_t$ with its initialization and iterative form as:

$$\begin{aligned} \Psi_1(i) &= 0 \\ \Psi_t(i) &= \arg\max_{1\geq j\geq N} \left[\delta_{t-1}(j) a_{ji}\right] \end{aligned} \tag{15}$$

From $t = 0$, the algorithm first keeps down $\delta_t(i)$ and $\Psi_t(i)$ for each hidden states. At $t = T$, it starts from c_T^* which has the greatest $\delta_T(i)$ and recursively check $\Psi_t(i)$ of each c_t^*.

Continued with the example in Section 1.4, if now the observation chain is given as $X =$ (Rise, Rise, Fall), to find the most possible hidden chain, Viterbi algorithm first calculates the sub-sequence $(\delta_t(i), \Psi_t(i))$ at time $t = 1$ for all possible hidden states i:

$$\begin{aligned} \delta_1(1) &= \lambda_1 b_1(x_1) = 0.175 \\ \delta_1(2) &= \lambda_2 b_2(x_1) = 0.025 \\ \delta_1(3) &= \lambda_3 b_3(x_1) = 0.44 \end{aligned} \tag{16}$$

$$\Psi_1(1) = \Psi_1(2) = \Psi_1(3) = 0 \tag{17}$$

Now we start the iteration of the subsequence on $t = 2$ with $x_2 =$ Rise:

$$\begin{aligned} \delta_2(1) &= \max_{1\le j\le 3} [\delta_1(j)a_{j1}]\, b_1(x_2) = 0.02625 \\ \Psi_2(1) &= \arg\max_{1\le j\le 3} [\delta_1(j)a_{j1}]\, b_1(x_2) = 1 \\ \delta_2(2) &= \max_{1\le j\le 3} [\delta_1(j)a_{j2}]\, b_2(x_2) = 0.022 \\ \Psi_2(2) &= \arg\max_{1\le j\le 3} [\delta_1(j)a_{j2}]\, b_2(x_2) = 3 \\ \delta_2(3) &= \max_{1\le j\le 3} [\delta_1(j)a_{j3}]\, b_3(x_2) = 0.2464 \\ \Psi_2(3) &= \arg\max_{1\le j\le 3} [\delta_1(j)a_{j3}]\, b_3(x_2) = 3 \end{aligned} \tag{18}$$

Finally, when $t = 3$ and $x_3 =$Fall:

$$\begin{aligned} \delta_3(1) &= \max_{1\le j\le 3} [\delta_2(j)a_{j1}]\, b_1(x_3) = 0.01232 \\ \Psi_3(1) &= \arg\max_{1\le j\le 3} [\delta_2(j)a_{j1}]\, b_1(x_3) = 3 \\ \delta_3(2) &= \max_{1\le j\le 3} [\delta_2(j)a_{j2}]\, b_2(x_3) = 0.03696 \\ \Psi_3(2) &= \arg\max_{1\le j\le 3} [\delta_2(j)a_{j2}]\, b_2(x_3) = 3 \\ \delta_3(3) &= \max_{1\le j\le 3} [\delta_2(j)a_{j3}]\, b_3(x_3) = 0.034496 \\ \Psi_3(3) &= \arg\max_{1\le j\le 3} [\delta_2(j)a_{j3}]\, b_3(x_3) = 3 \end{aligned} \tag{19}$$

Since $\arg\max_i \delta_3(i) = 2$ is the most possible hidden state at time $t = 3$, and according to $\Psi_3(2) = c_2 = 3$ and $\Psi_2(3) = c_1 = 3$, the hidden chain is thus constructed as $\{3, 3, 2\}$, which is {U, U, B}.

3 Parameter Estimation: EM-algorithm

EM algorithm provides an iterative approach when the true parameter Θ and hidden states are unknown, and only observable variables are given. It is widely used in the parameter approximation of probability models with latent variables (unobserved data such as hidden states) and Posterior probability (Dempster et al. 1977 [4]). The main idea is to find the expectation of the "complete data" constructed by known observations and latent variables, and approximate parameters through maximizing the complete expectation is supposed to be easier than directly maximizing the likelihood of the observation. This algorithm is composed by the E-step (expectation) and M-step (maximization) in each iteration:

1. **E-step**: In the previous section of direct maximization on likelihood of the observation chain, we are looking for the probability of having $X = \{X_1 = x_1,\ X_2 = x_2,\ ...,\ X_T = x_T\}$. Similarly, in E-step we are calculating the probability of the unknown variables $P(C|X)$ given the condition of observations and current parameters $\Theta^{(0)}$, which are initialized at the begining of the algorithm or given by the output from the last iteration, and then we can find the expression of the expectation of complete data with new parameters $\Theta^{(1)}$, but conditioned on the current data $\Theta^{(0)}$: $E\left[\log P(X, C|\Theta)|\Theta^{(0)}\right]$. Note that new parameters are unknown, so this expectation is actually a function on $\Theta^{(1)}$.

2. **M-step**: It is necessary and straightforward to find a new set of parameters $\Theta^{(1)}$ that maximize the expectation. Usually, partial derivative on (A, B, λ) and Lagrange multiplier is used for optimization.

We should first initialize the starting parameters $\Theta^{(0)}$, usually by some simple calculations, and in each iteration after the M-step, we should plug the new generation of approximated parameters into E-step, until they converge (usually determined by a given threshold supervising the degree of change from $\Theta^{(0)}$ to $\Theta^{(1)}$) or reach the limit of iterations. Although due to the choice of starting parameters, EM-algorithm sometimes falls into local maximum, some refined approaches such as Monte-Carlo EM-algorithm (Wei, et al. 1990 [17]) would offset this negative influence, or multiple trials with different starting parameters may also provide with comprehensive inspections.

3.1 Basic EM-algorithm for HMM with Full Observations

The EM-algorithm in the context of HMMs, also called Bawm-Welch algorithm (Baum et al.1970 [2]), is used when the hidden chain(latent variable) and set of true parameters $\Theta = (A, B, \lambda)$ are both unknown. The likelihood of complete data

is:

$$\begin{aligned} P(X, C) &= \lambda_{c_1} b_{c_1}(x_1) a_{c_1,c_2} b_{c_2}(x_2) ... \\ &= \lambda_{c_1} \prod_{i=1}^{T} b_{c_i}(x_i) \prod_{i=1}^{T-1} a_{c_i,c_{i+1}}. \end{aligned} \tag{20}$$

with the likelihood of hidden states given by:

$$P(C|X) = \frac{P(X, C)}{L_T(X)} \tag{21}$$

Since the likelihood is composed by parts of λ, A, and B and conditioned by the observation chain, it is necessary to find $\sigma_t(i) = P(c_t = i|X)$ and $\epsilon_t(i, j) = P(c_t = i, c_{t+1} = j|X)$ as references of A and B.

Note that in EM-algorithm we cannot find replacement of λ directly, since the initial distribution of the Markov Model is only related to the first observation x_1 and the first hidden state c_1, making this implausible to approximate it from only one observation. However, we can use the idea in Section2.2, because approximation on λ is the local optimization on a single time point $t = 1$. Therefore λ_i is calculated from the approximated $P(c_1 = i|X) = \sigma_1(i)$, which is derived after we find approximations on(A, B).

Before we start EM-algorithm, formulations of $\sigma_t(i)$ and $\epsilon_t(i, j)$ are given by:

$$\begin{aligned} \sigma_t(i) &= P(c_t = i|X) \\ &= \frac{\alpha_t(i)\beta_t(i)}{L_T(X)} \end{aligned} \tag{22}$$

$$\begin{aligned} \epsilon_t(i, j) &= P(c_t = i, c_{t+1} = j|X) \\ &= \frac{P(c_t = i, c_{t+1} = j,\ X)}{L_T(X)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(x_{t+1}) \beta_{t+1}(j)}{L_T(X)} \end{aligned} \tag{23}$$

The above calculations are based on the original form of likelihood, but one can also directly use the log-likelihood $\log P(X, C)$ or scaled likelihood.

1. HMM E-step: If the current parameters are $\Theta^{(0)} = (A^{(0)}, B^{(0)}, \lambda^{(0)})$, then the expectation of the complete data with the next generation of parameters $\Theta^{(1)}$ based on the probability of current parameters of $P(C|X, \Theta^{(0)})$, is constructed as:

$$E\left[\log P(X, C|\Theta)|\Theta^{(0)}\right] = \sum_{C} P(C|X, \Theta^{(0)}) \log P(C, X|\Theta) \tag{24}$$

To maximize the expectation, we should find $\Theta^{(1)} = (A^{(1)}, B^{(1)}, \lambda^{(1)})$ such that:

$$\begin{aligned}\Theta^{(1)} &= \arg\max_{\Theta} \sum_{C} P(C|X, \Theta^{(0)}) \log P(X, C|\Theta) \\ &= \arg\max_{\Theta} \sum_{C} P(C|X, \Theta^{(0)}) \Big(\log \lambda_{c_1} + \sum_{i=1}^{T} \log b_{c_i}(x_i) + \sum_{i=1}^{T-1} \log a_{c_i, c_{i+1}} \Big)\end{aligned} \tag{25}$$

and because $P(C|X, \Theta^{(0)}) = \frac{P(C, X|\Theta^{(0)})}{P(X|\Theta^{(0)})}$, and $P(X|\Theta^{(0)})$ is a constant likelihood $L_T(X)$ calculated using $\Theta^{(0)}$, the formula is then:

$$\Theta^{(1)} = \arg\max_{\Theta} \sum_{C} P(C, X|\Theta^{(0)}) \Big(\log \lambda_{c_1} + \sum_{i=1}^{T} \log b_{c_i}(x_i) + \sum_{i=1}^{T-1} \log a_{c_i, c_{i+1}} \Big) \tag{26}$$

2. HMM M-step: Equation 25 splits the expectation of the complete data into three partial maximization on $(A^{(1)}, B^{(1)}, \lambda^{(1)})$ separately. By taking derivatives and applying Lagrange multipliers (Liu et al. [13]), we can have iterative formulas as:

$$\begin{aligned}\lambda_i &= \sigma_1(i) \\ a_{ij} &= \frac{\sum_{t=1}^{T-1} P(X, c_t = i, c_{t+1} = j)}{\sum_{t=1}^{T-1} P(X, c_t = i)} = \frac{\sum_{t=1}^{T-1} \epsilon_t(i, j)}{\sum_{t=1}^{T} \sigma_t(i)} \\ b_j(k) &= \frac{\sum_{t=1}^{T} P(O, c_t = j, x_t = k)}{P(O, c_t = j)} = \frac{\sum_{t=1, x_t = k}^{T} \sigma_t(i)}{\sum_{t=1}^{T} \sum_{t=1}^{T} \sigma_t(i)}\end{aligned}$$

where all $\sigma_t(i)$ and $\epsilon_t(i, j)$ are calculated using current parameters $\Theta^{(0)}$. This new set of parameters $(A^{(1)}, B^{(1)}, \lambda^{(1)})$ will be applied into E-step, and iterations will carry on until all parameters fall into convergence.

3.2 EM-Viterbi algorithm for Missingness

The original EM-algorithm will still work on missing observations in M-step: for example, in Equation 25, the term of transition matrix $A^{(1)}$ is constructed as:

$$\sum_{C} \sum_{t-1}^{T-1} P(C, X|\Theta^{(0)}) (\log a_{c_i, c_{i+1}}) = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{t=1}^{T-1} P(X, c_t = i, c_{t+1} = j, |\Theta^{(0)}) \log a_{ij} \tag{27}$$

we can apply Lagrange multiplier because $\sum_{j=1}^{N} a_{ij} = 1$. However, with a single missing observation at time t^*, Equation 27 will be modified as:

$$\begin{aligned}
&\sum_{C}\sum_{t-1}^{T-1} P(C, X|\Theta^{(0)})(\log a_{c_i,c_{i+1}}) \\
=&\sum_{i=1}^{N}\sum_{j=1}^{M}\left[\sum_{t=1,t\neq t^*}^{T-1} P(X, c_t = i, c_{t+1} = j, |\Theta^{(0)}) \log a_{ij}\right. \\
&\left.+\sum_{k=1}^{N} P(X, c_{t^*-1} = i, c_{t^*} = k, c_{t^*+1} = j|\Theta^{(0)}) \log a_{ik}a_{kj}\right]
\end{aligned} \tag{28}$$

and thus the derivative of new Lagrange multiplier equation of any transition probability a_{uv} is with some extra terms related to missing time:

$$\begin{aligned}
0 = &\frac{P(X, c_t = u, c_{t+1} = v, |\Theta^{(0)})}{a_{uv}} + \gamma a_{uv} \\
&+ \frac{\sum_{j=1}^{N} P(X, c_{t^*-1} = u, c_{t^*} = v, c_{t^*+1} = j|\Theta^{(0)})}{a_{uv}} \\
&+ \frac{\sum_{i=1}^{N} P(X, c_{t^*-1} = i, c_{t^*} = u, c_{t^*+1} = v|\Theta^{(0)})}{a_{uv}}
\end{aligned} \tag{29}$$

in the equation above there are only two extra terms because we assume a single missing time t^*, in the case of consecutive missing values, there should be more terms. Therefore, the only problem is from the calculation on a longer missing chain: if we have a missing chain on the interval $D = [t_1, t_2]$ with length L, then the second term of Equation 28 will be an L-nested summation, with increased complexity to $O(N^L)$. Also, the probability in the second term can be regarded as the cumulative product with length L on a_{ij} entries of A, but for large L, the product will be close to λ_j regardless of the starting state i due to stationary property. Instead, we may first apply Viterbi algorithm to find an approximation of hidden states. After each iteration of EM-algorithm, Viterbi with current fitted parameters $(A^{(0)}, B^{(0)}, \lambda^{(0)})$ is able to provide with the most likely $c_1, ..., c_{t_1-1}$, and starting with c_{t_1-1} we can calculate the most possible $\{x_{t_1}..., x_{t_2}\}$ iteratively as a compensate of missing values.

3.2.1 EM-Viterbi Algorithm

For any observation chain with missing value starting at x_{t_1}, a nested EM-Viterbi algorithm is constructed. In the first iteration, we only use the first consecutive section of the observation chain $\{x_1, ..., x_{t_1-1}\}$ to EM-algorithm; because of the assumption on stationary HMMs, it is reasonable to start at the middle of the sequence.

After the iteration, Viterbi algorithm would give the most likely hidden chain according to current fitted parameters, which enable a derivation method, similar to forward/backward probabilities, of missing observations:

$$x^*_{t_1} = \arg\max_{x_{t_1}} \sum_{i=1}^{N} \left[\delta_t(j) a^{(0)}_{ji}\right] b_i(x^{(0)}_{t_1}) \tag{30}$$

This forecasted observation would be a replacement of the missing value. The new observation chain with one more forecasted value is then plugged in the next EM iteration, and the new set of parameters will give a different hidden chain for further calculation, providing prediction at x_{t_1+1}... If the current missing section up to x_{t_2} is filled, we can plug all known observations after t_2 until we have another section of missing values. Because EM-algorithm is dependent on all given observations, including predicted ones, the approximated parameters may change vastly when there are newly predicted observations, so we should not focus on the convergence before all missing values are replaced.

Since in the setting above we only predict one missing observation every time, and this modified algorithm performs two subparts (EM and Viterbi) in each iteration, approximation may take longer time to converge, which makes it impractical for longer chains. One can simplify the time complexity by predicting more than one observations in each iteration. However, such forecasting of longer time in one iteration would produce more errors because of memoryless property of Markov chain. It is recommended to apply such alternative prediction when there are more short missing chains. In missing chains long enough, even the prediction by only one more observation per iteration would still fail. It is illustrated in Section 5.

Algorithm 1 EM-Viterbi Algorithm

Parameters: $(A^{(0)}, B^{(0)}, \lambda^{(0)})$, X: observation chain, tol: threshold of convergence, n_{iter}: max iteration of the algorithm after the observation is fully imputed, n_{inside}: max iteration before the observation is fully imputed (usually, n_{inside} is around $\frac{1}{10}$ of n_{iter}), D: set of indices of missing values, d: number of imputed observations in Viterbi section of each iteration
initialize a HMM with given $(A^{(0)}, B^{(0)}, \lambda^{(0)})$
find the first missingness set D in the chain at time $t_{init}+1$ and the corresponding first full observation up to $X_{t_{init}}$
while length of $X_{t_{init}}$ is smaller than X **do**
    Apply EM algorithm on $X_{t_{init}}$ with n_{inside}
    compute the most likely hidden chain through Viterbi algorithm
    if $t_{init}+1 \in D$ **then**
        find the most likely observation at $t_{init}+1$, append it to $X_{t_{init}}$
        remove $t_{init}+1$ from D
    else
        find the next index t_{next} in D, append all observation up to $t_{next}-1$ to $X_{t_{init}}$
    end if
end while
Apply EM algorithm on the imputed observation chain with given n_{iter} and tol. The algorithm ends when the change of parameters in each iteration is below tol or when it reaches the limit of iteration n_{iter}
return approximated parameters $\hat{\Theta} = (\hat{A}, \hat{B}, \hat{\lambda})$

4 Model Selection: Assessing Accuracy

Like many other algorithms for approximation, EM algorithm starts with the assumption of unknown true parameters, and due to the complex structure of multi-stochastic model such as HMM, we cannot directly analyze the effectiveness through working on testing sets. Therefore, it is necessary to develop an alternative approach of model selection on HMM. Moreover, the unknown transition $N \times N$ matrix A also contains information on possible number of hidden states, while in EM algorithm we start with initial $A^{(0)}$ of given size. Therefore, a set of competitive models in approximating HMM is mainly derived from different N, and one kernel idea of model selection on HMM is to find the most likely number of states. In our thesis, we discuss two approaches of model selection: AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion).

4.1 AIC & BIC

Selections of AIC and BIC are built in terms of the Kullback-Leibler information with respect of the "difference" between a fitted model with approximated parameter $\hat{\Theta}$ and the true model (McLachlan et al. [14]):

$$I\{f(X), f(X, \hat{\Theta})\} = \int f(X) \log f(X) dX - \int f(X) \log f(X, \hat{\Theta}) dX \tag{31}$$

where $I\{f(X), f(X, \hat{\Theta})\}$ is the information and $f(X)$ is the true density. In the context of multi-stochastic process such as HMMs, $f(X)$ marks the probability for the observation given by all possible hidden states.

Almost all information criterion derived from Kullback-Leibler have similar forms in the minimization of:

$$-\log L(\hat{\Theta}) + b(F) \tag{32}$$

where $L(\hat{\Theta})$ is the likelihood from the fitted parameters, and $b(F)$ is the bias from the true distribution, which is more like a "penalty term" to correct the likelihood from being over-fitting.

With the general structure, AIC & BIC are formulated as:

$$\begin{cases} AIC(\hat{\Theta}) = -2 \log L(\hat{\Theta}) + 2N \\ BIC(\hat{\Theta}) = -2 \log L(\hat{\Theta}) + 2N \log T \end{cases} \tag{33}$$

N is the number of parameters (hidden states in the context of HMMs) in the distribution, and T is the number of observations. The essential idea is still to find parameters that maximize the posterior likelihood of the parameters (represented by $\log L(\hat{\Theta})$) but balanced by the complexity of the fitted model (represented by $b(F)$), which is, in most cases, mainly determined by the number of parameters. Apparently, choosing the penalty term $b(F)$ is the most significant but difficult part in the determination of information criteria. Many researchers (such as Kuha [9] and Aho [1]) have stated that with fixed penalties (when $b(F)$ is fixed with respect to the structure of the data and does not fluctuate much with different testing sample), AIC may have over-fitted outcomes of the distribution in cases of large set of observations, when the penalty $2d$ may not efficiently restrict $-2 \log L(\hat{\Theta})$. On the contrast, the situation of BIC suffer less from over-fitting, and is more generally applied by other authors. We will mainly discuss on BIC in following sections.

Recall that in Equation 6, in the context of HMMs, the likelihood is formulated from the sum of multiplication of forward/backward probabilities. But such approach may be time-consuming when facing large number of parameters, and most useless calculation is from the marginalization of hidden states c_t. Dridi and Hadzagic (Dridi et al. [6]) provide with an alternative formulation based on Bayes rule that for models M_i:

$$
\begin{aligned}
L(\hat{\Theta}) &= Q(\hat{\Theta}, \hat{\Theta}) - \log P(C|X, \hat{\Theta}) \\
M_{BIC} &= \arg\min_{M_i} \Big[-2 \log L(\hat{\Theta}) + 2N \log T \Big]
\end{aligned}
\tag{34}
$$

where:

$$
\begin{aligned}
Q(\hat{\Theta}, \hat{\Theta}) &= \log P(c_1) + \sum_{t=1}^{T} \log P(c_t|c_{t-1}) \\
&\quad + \sum_{t=0}^{T} \log P(x_t|C) \\
\log P(C|X, \hat{\Theta}) &= \log P(c_1|X, \hat{\Theta}) + \sum_{t=1}^{T} \log P(c_t|c_{t-1}, X, \hat{\Theta})
\end{aligned}
\tag{35}
$$

Therefore the only part having iterative calculation on forward/backward probabilities is the second term of $\log P(C|X, \hat{\Theta})$.

The only problem of BIC is that, some researchers (Pohle et al. [15]) had observed that BIC is driven by the potential Bayesian essence that specify the most likely model. Therefore, when running BIC we always assume that at least one of those tested models performs accurate approximation of the actual distribution. Unfortunately, in practical cases the true distribution, especially the emission process of observations, can be drastically complicated. In multi-stochastic process like HMMs it is unreasonable to assume that any single model would effectively fit the true model even though vectors of multi-variables are in a lower dimension, and thus if there is no "good approximations," BIC has the tendency on slight under-fitting by choosing in favor of more ordinary models indicated with less number of states, which have less number of parameters and capture some general structures of the distribution.

4.2 BIC with Missing Observations

For a partially missing observation chain X^{-D} with the set of time of missing values D, we can directly apply BIC with imputed observations in a situation of scattered missingness as scattered missing observations have little local influence on the entire chain, and therefore the BIC would perform with a similar accuracy. However, since consecutive missingness would vastly drive the model into the direction of imputed values (as it is illustrated in experiments of Section 5), BIC would have modifications similar to the "ignorable likelihood" of HMMs.

If there is a missing chain on the interval $D = [t_1, t_2]$, we can also apply the ignorable likelihood by simply take the form:

$$
\begin{aligned}
Q^{-D}(\hat{\Theta},\hat{\Theta}) = \log P(c_1) + \sum_{t=1}^{t_1-1} \log P(c_t|c_{t-1}) + \log P(c_{t_2+1}|c_{t_1-1}) \\
+ \sum_{t=t_2+2}^{T} \log P(c_t|c_{t-1}) + \sum_{t=0,t\notin D}^{T} \log P(x_t|C)
\end{aligned}
\tag{36}
$$

$$
\begin{aligned}
\log P^{-D}(C|X,\hat{\Theta}) = \log P(c_1|X,\hat{\Theta}) + \sum_{t=1}^{t_1-1} \log P(c_t|c_{t-1},X,\hat{\Theta}) \\
+ \log P(c_{t_2+1}|c_{t_1-1},X,\hat{\Theta}) + \sum_{t=t_2+2}^{T} \log P(c_t|c_{t-1},X,\hat{\Theta})
\end{aligned}
\tag{37}
$$

Therefore, the pseudo-likelihood of the model in this modified BIC would be:

$$
\begin{aligned}
L^{-D}(\hat{\Theta}) &= Q^{-D}(\hat{\Theta},\hat{\Theta}) - \log P^{-D}(C|X,\hat{\Theta}) \\
M_{BIC}^{-D} &= \arg\min_{M_i} \left[-2\log L^{-D}(\hat{\Theta}) + 2N\log\left(T - \|D\|\right) \right]
\end{aligned}
\tag{38}
$$

Generally, the algorithm of BIC for HMM is composed as:

Algorithm 2 Calculation of BIC-HMM

1: Parameters: (A, B, λ) of the model, (C, X) is the given chain of length T, where X should be known by given or by decoded through Viterbi algorithm, interval of missing value $D = [t_1, t_2]$ (if nothing is missing, D is set to be empty)
2: Initialize $Q^{-D}(\hat{\Theta},\ \hat{\Theta}) = \log \lambda_{c_1}$
3: Initialize $\log P^{-D}(C|X,\hat{\Theta}) = \log(\beta_1(c_1)\lambda_{c_1} b_{c_1}(x_1))$
4: **for** all t such that $t \notin D$ **do**
5: $\quad \log P(c_t|c_{t-1}) = \log a_{c_{t-1},c_t}$
6: $\quad \log P(x_t|C) = \log b_{c_t}(x_t)$
7: $\quad \log P(c_t|c_{t-1},X,\hat{\Theta}) = \log\left[\frac{\alpha_t(c_{t-1}) a_{c_{t-1},c_t} b_{c_t}(x_t)\beta_t(c_t)}{\alpha_t(c_{t-1})\beta_t(c_{t-1})}\right]$
8: $\quad$ add $(\log P(c_t|c_{t-1}) + \log P(x_t|C))$ to $Q^{-D}(\hat{\Theta},\ \hat{\Theta})$
9: $\quad$ add $\log P(c_t|c_{t-1},X,\hat{\Theta}$ to $\log P^{-D}(C|X,\hat{\Theta})$
10: **end for**
11: $\log P(c_{t_2+1}|c_{t_1-1}) = \log\left[\prod_{t=t_1}^{t_2-1} a_{c_t,c_{t+1}}\right]$
12: $\log P(c_{t_2+1}|c_{t_1-1},X,\hat{\Theta}) = \log\left[\frac{\alpha_t(c_{t_1-1})\prod_{t=t_1}^{t_2-1} a_{c_t,c_{t+1}}\beta_t(c_{t_2+1})}{\alpha_t(c_{t_1-1})\beta_t(c_{t_1-1})}\right]$
13: add the result in Step 11 and Step 12 to $Q^{-D}(\hat{\Theta},\ \hat{\Theta})$ and $\log P^{-D}(C|X,\hat{\Theta})$ separately
14: calculate number of hidden states N, which is the number of rows of A
15: $BIC = -2(Q^{-D}(\hat{\Theta},\hat{\Theta}) - \log P^{-D}(C|X,\hat{\Theta})) + 2N\log(T - \|D\|)$

5 Simulation and Results

5.1 Settings

In our experiments, we aim to compare the effectiveness between the original EM-algorithm and modified EM-Viterbi algorithm with missing values. In our thesis we apply a dataset of weekly corn price [18] originally downloaded from Quantopian corn futures price.

Generally, the experiment includes the comparison between original EM-algorithm and EM-Viterbi algorithm. In order to control the variable, in each pair of experiment, both algorithms are applied on the same observation chain, but EM-Viterbi algorithm only starts with partially full data, and missing parts are randomly selected. Also, unless specifically stated, some parameters in EM and EM-Viterbi algorithms are taken by default as stated below: the threshold of convergence is set to be 0.01 and max number of iterations is 100, and we will stop either if the difference of parameters in each iteration (measured by Frobenius norm) is below the threshold or if we finish 100 times of iterations. Step size d in Viterbi section of EM-Viterbi algorithm is set to be $d = 1$, and number of states is set to be $N = 3$.

To fully understand the performance of the two algorithms, we will analyze them in two directions: either by directly compare the output of EM-Viterbi and original EM algorithm through their difference, or check the difference in BIC of the two algorithms under different situations:

Output-1. We will compare the difference of approximated parameters in the modified algorithms based on their distance (Frobenius norm between matrices), and we should separately test conditions of scattered missing values or missing chains. The goal is to check any tendency related to the total number of missingness and the max length of missing chains.

Output-2. In order to have a more comprehensive understanding, the accuracy of prediction on missing observations is also evaluated. We will simulate multiple approximation on the same set of observations with a single missing chain of fixed length. The idea is to analyze the change of accuracy of imputation from the head to tail of the missing chain, with respect to different step size of imputed observations d in each iteration.

BIC-1. Since the number of hidden states (which is also manually set like initial parameters) in the approximation plays essential role in fitting, we compare the difference of BIC between two algorithms when fitting the same set of observations.

BIC-2. Also, since BIC indicates the approximation on number of states, we will investigate on the optimal number of states approximated from BIC under conditions of different lengths of the missing chain.

5.2 Results

Output-1. In this section, the observation is a series about corn price from 2013 to 2017 of length $T = 248$, and we first apply the EM-Viterbi algorithm to the observation with $m = \{5, 10, 15, 25, 35, 50\}$ missing values randomly scattered in the data with the consideration of avoiding consecutive missing values. Based on 50 random selection of missing observations with each m, bar-plots of distance between true transition matrix and approximated transition matrix are shown below, measured by Frobenius norm of matrices:

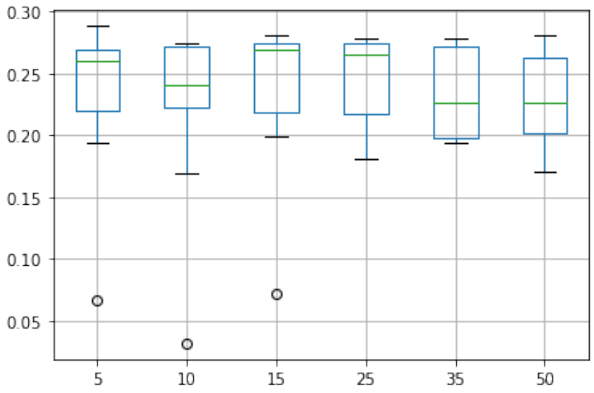


Figure 4: Difference in transition matrix norm for different m

From the outputs, low errors shows that all of approximations are comparatively close to true parameters, and there is no apparent tendency of change related to number of missing values. In fact, the main source of false approximation hinges to the length of each missing chain: in the context of consecutively missing values, modified algorithm still fails when the missing chain is long enough. After we tested on observations with number of consecutively missing values $m = \{2, 3, 5, 10, 20, 25\}$, the plot shows an apparent pattern of increasing errors:

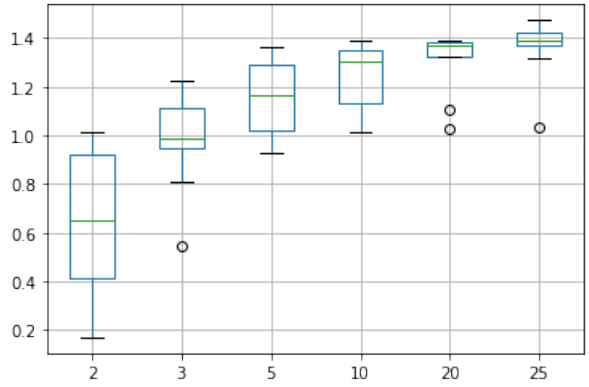


Figure 5: Errors with number of consecutively missing values

More interestingly, it shows that the increment of length is driving the model to a more "extreme form": some specific entries a_{ij} in transition matrix A will be close to 0, while some other entries will be close to 1. In the most extreme situation of $m = 25$, the initial vector λ even have $\lambda_3 = 1$, while all other initial probabilities are 0. It is possibly because those imputed observations are pulling the model in some direction, and thus longer missing chains lead to larger bias.

Output-2. With the output from last part, it is more important to investigate on the divergence of prediction on missing chains. In the experiment, the same dataset is applied but with fixed length of missing chain $m = 10$. Instead of predicting one more missing observation in each iteration, we now forecast different number of observations $d = 1, 2, 3$ every time. For each of d, we will apply 50 trials of approximation with EM-Viterbi algorithm, and the proportion of correct prediction at each missing time (starting at one, up to 10) is in the plot.

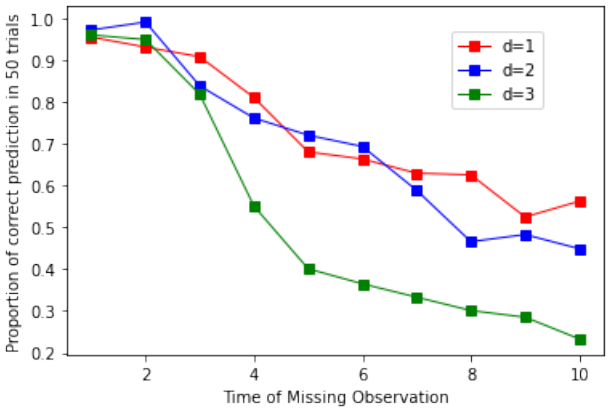


Figure 6: Accuracy of prediction on missing observations from $t^* = 1$ to $t^* = 10$. d is the number of imputed missing observations in Viterbi section of each iteration. Recall that $d = 1$ is what we usually assumed in the algorithm

According to the plot, there is a dominating trend of decreasing accuracy with increasing length of missing values in all 3 settings, and for larger steps it seems to have worse trend with faster decreasing speed. The failure of larger steps apparently derives from the memoryless property of Markov Model: it is relatively hard to apply a far-sighted prediction if only based on few previous states; while the reason of decreasing trend when $d = 1$ still remains to be discussed. It may be from the "raw approximation" in the first few iterations, where the approximation of parameters is still comparatively far from the convergence on true values, so some of the false prediction based on the current parameters may drive the approximation away in the next generation.

BIC-1. In this experiment, we apply the observation of time length $T = 248$, with 15 missing values scattered in the chain also with consideration of avoiding missing

chains. The goal is to find the approximated number of states favored by BICs, and the plot of two BICs is

Figure 7: BIC of different number of states (scattered missingness)

It is predicted and concluded that BIC of EM-Viterbi algorithm is higher than the original EM algorithm in most time and represents a moderately worse approximation preferring higher model complexity by choosing slightly more number of states for better approximation, and it may potentially lead to overfitting. Also, the trend of their difference is noteworthy: the changing trend of BIC of EM-Viterbi is close to the BIC of original EM algorithm, but with far greater degree of change: the two BICs are close to each other when scores are low (representing a favored number of states), while BIC of EM-Viterbi algorithm is vastly higher when the number is not preferred. Though the reason remains uncertain, it seems that the EM-Viterbi algorithm construct a far more sensitive model on detecting plausible number of states.

However, in the situation of all 15 consecutively missing observations, it is expected and concluded that the BIC of EM-Viterbi algorithm indicates an inaccurate model:

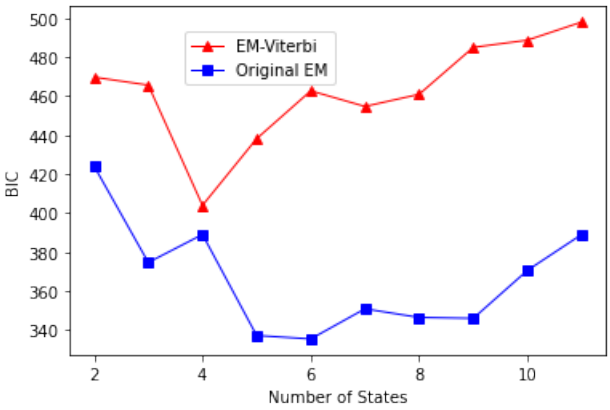
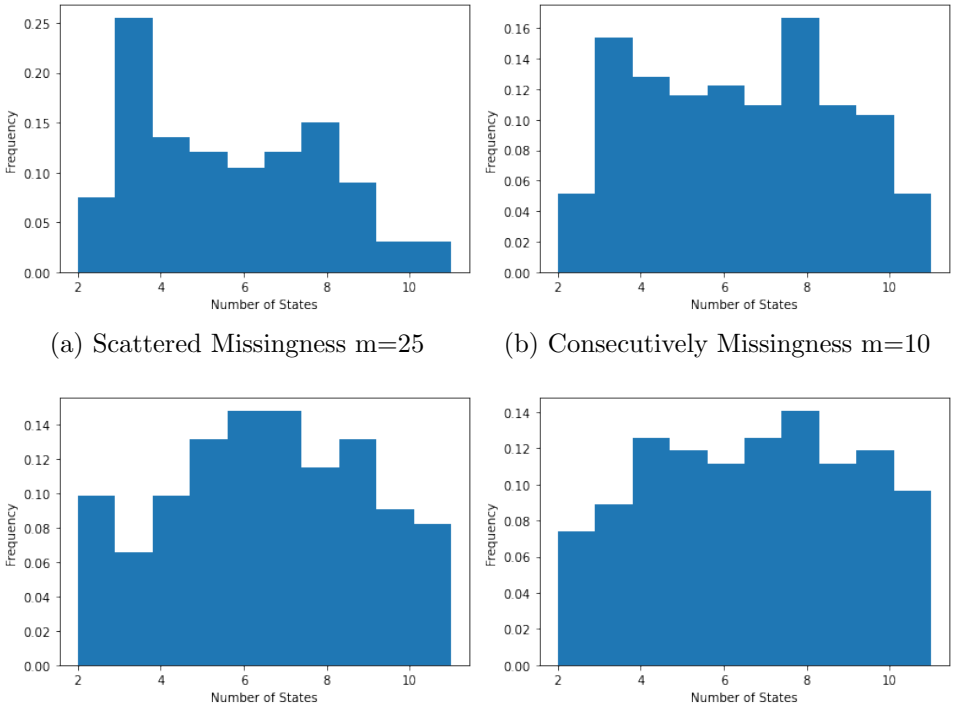


Figure 8: BIC of different number of states (consecutive missingness)

the lowest point of BIC curve is not corresponding to the curve of original EM, and the slightly more flat curve also indicates the model is not comparatively robust in approximating the number of states. This conclusion is discussed with more details in the next part:

BIC-2. On the other hand, we tested on the influence of consecutively missing values on performance of BIC. Similarly in **Output-1**, set of tested missing numbers are $m = \{2, 3, 5, 10, 20, 25\}$, but we will also have range of possible number of states $N = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$; 100 trials are simulated on each m with the same observation, and proportions of optimal number of states favored by BIC in each simulation are shown in histograms. Plots of $m = \{10, 20, 25\}$ are shown below. Histogram of simulation on scattered missing values when $m = 25$ are also displayed as a comparison:



(a) Scattered Missingness m=25

(b) Consecutively Missingness m=10

(c) Consecutively Missingness m=20

(d) Consecutively Missingness m=25

Figure 9: Histograms of Optimal Number of States

From the figures above, in situation of scattered missingness the optimal number of states produced by EM-Viterbi will be close to the number preferred by original EM(mostly at $n = 3$ with the second highest bar at $n = 8$), while with the increasing length of missing chain, it seems to be "uncertain" on the number, represented by similar frequency in multiple groups, producing a flat histogram in their corresponding intervals.

References

[1] Ken Aho, DeWayne Derryberry, and Teri Peterson. Model selection for ecologists: the worldviews of aic and bic. *Ecology*, 95(3):631–636, 2014.

[2] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

[3] Namrata Dave. Feature extraction methods lpc, plp and mfcc in speech recognition. *International journal for advance research in engineering and technology*, 1(6):1–4, 2013.

[4] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[5] José G. Dias, Jeroen K. Vermunt, and Sofia Ramos. Clustering financial time series: New insights from an extended hidden markov model. *European Journal of Operational Research*, 243(3):852–864, 2015.

[6] Noura Dridi and Melita Hadzagic. Akaike and bayesian information criteria for hidden markov models. *IEEE Signal Processing Letters*, 26(2):302–306, 2018.

[7] S R Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–763, 10 1998.

[8] Xiaotian Guo. Financial application of HMM. `https://www.zhihu.com/question/34868706/answer/106024559`.

[9] Jouni Kuha. Aic and bic: Comparisons of assumptions and performance. *Sociological methods & research*, 33(2):188–229, 2004.

[10] Brian G Leroux and Martin L Puterman. Maximum-penalized-likelihood estimation for independent and markov-dependent mixture models. *Biometrics*, pages 545–558, 1992.

[11] Li Li. Speech recognition based on hmm. `http://fancyerii.github.io/books/asr-hmm3/`.

[12] Roderick JA Little. Selection and pattern-mixture models. *Longitudinal data analysis*, pages 409–431, 2008.

[13] Pinard Liu. Hmm parameters from baum-welch. `https://www.cnblogs.com/pinard/p/6972299.html`.

[14] Geoffrey J McLachlan and Suren Rathnayake. On the number of components in a gaussian mixture model. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(5):341–355, 2014.

[15] Jennifer Pohle, Roland Langrock, Floris M. Beest, and Niels Martin Schmidt. Selecting the Number of States in Hidden Markov Models: Pragmatic Solutions Illustrated Using Animal Movement. *Journal of Agricultural, Biological and Environmental Statistics*, 22(3):270–293, September 2017.

[16] Douglas A Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, 741:659–663, 2009.

[17] Greg CG Wei and Martin A Tanner. A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704, 1990.

[18] Nick Wong. Weekly corn price. `https://www.kaggle.com/nickwong64/corn2015-2017`.

[19] Xiaoming Wu, Changxin Song, Bo Wang, and Jingzhi Cheng. Hidden markov model used in protein sequence analysis. *Journal of Biomedical Engineering*, 19(3):455–458, 2002.

[20] Walter Zucchini, Iain L MacDonald, and Roland Langrock. *Hidden Markov models for time series: an introduction using R*. CRC press, 2017.