3-1-2018

Estimation of Hidden Markov Model

Xuecheng Yin
Lehigh University, 124831816@qq.com

Estimation of Hidden Markov Model

by

Xuecheng Yin

A Thesis
Presented to the Graduate and Research Committee
of Lehigh University
in Candidacy for the Degree of
Master of Science
in
Industrial and Systems Engineering

Lehigh University
May, 2018

Approved and recommended for acceptance as a dissertation in partial fulfillment of the requirements for the degree of Master of Science.

Date

Prof Boris Defourny, Thesis Advisor

Prof Ted Ralphs, Interim Chairperson of Department

Acknowledgements

I want to thank my parents for the supporting throughout my life and study. Also, I'd like to thank Professor Boris Defourny for his patience, encouragement and guidance in the master's thesis process.

Content

Acknowledgements iii

List of Tables vi

List of Figures vii

Abstract viii

1. Markov Model Theory **1**

1.1 Markov Process 1

1.2 Hidden Markov Model Theory 1

1.3 Hidden Markov Model Structure 4

1.3.1 Hidden Markov Model Hypothesis 4

1.3.2 The Fundamental Form of Hidden Markov Model 4

1.3.3 The difference between MM and HMM 5

1.3.4 Continuous-time Hidden Markov Model 5

1.4 Summary 9

2. Three classical Problems of HMM **11**

2.1 Evaluation 11

2.2 Decoding 11

2.3 Learning 12

2.4 Algorithms of HMM 12

2.4.1 Forward-backward Algorithm 12

2.4.2 Viterbi Algorithm 14

2.4.3 Baum Welch algorithm 15

3. Poisson Hidden Markov Model **16**

3.1 Poisson Hidden Markov Model Hypotheses 16

3.2 Parameter Estimation of Poisson Hidden Markov Model 17

4. Estimating the Economic Performance of Dongfeng Motor Corporation **20**

5. Future Work......24

6. References......27

7. Appendix 29

8. Biography......35

List of Tables

1.1 The Observation State and Hidden State in Stock Market.....................................**2**

1.2 Comparison between the Markov and Hidden Markov Models **5**

4.1 2016-2017 Dongfeng Motor Corporation Sales...**20**

5.1 2005-2016 Dongfeng Market Share...**24**

List of Figures

1.1 The Traditional Markov Model 2

1.2 Hidden Markov Model 3

1.3 Transition Rate Matrix of Continuous-time Markov Model 6

1.4 Sample Path with value 100 7

1.5 Transition Probability Matrix 7

1.6 Sample Path with value 1000 8

1.7 Sample Path with value 10000 8

1.8 Continuous-time HMM with value 1000 9

1.9 Continuous-time HMM with value 10000 9

4.1 2016-2017 Sales volume of Dongfeng Motor Corporation 20

4.2 Sample Path of Dongfeng with significant change of G 21

4.3 Sample Path of Dongfeng with subtle change of G 22

4.4 Sample Path of Dongfeng with value 10000 23

5.1 2005-2006 Dongfeng Market share 25

Abstract

With the development of economy, estimation has gradually received attention. Economic performance is essential to a company, that's why data analyst is very popular. Since Dongfeng Motor Corporation is one of the magnate company in Chinese vehicle market, estimation the data of Dongfeng could be very meaningful. There are many methods used to estimate economic performance, in this thesis we mainly focus on Hidden Markov Model (HMM).

First of all, the thesis introduces the basic concept of Markov Process and Hidden Markov model, including three classes of problems, evaluation, decoding, learning problems. Also, the thesis introduces the corresponding solution algorithms, which are Forward-Backward algorithm, Viterbi algorithm, Baum-Welch algorithm.

Secondly, the thesis introduces a special case of HMM, named Poisson Hidden Markov Model (PHMM), including a very clear explanation of PHMM and parameter estimation.

Thirdly, the thesis gives an example of economic performance estimation of Dongfeng Motor Corporation. Several data sets can be used to do the estimation and different models should be used with the different kinds of data. The example uses sales volume data to make estimation with continuous-time hidden Markov model.

Finally, the thesis gives future work directions. The estimation of different data would be introduced in the last part. Potential applications of the Poisson Hidden Markov Models to estimate economic performance are proposed.

Keywords: Hidden Markov Model, Poisson Hidden Markov Model, Estimation

Estimation of Hidden Markov Model

Xuecheng Yin

May 4, 2018

1 Markov Model Theory

1.1 Markov Process

Firstly we set stochastic process {X(t),t$\in T\}.If$ X$_t$ means the state at time t and the process is Markov process, the specific performance we could find is that $X_t + 1$ has no concern with X_{t-1}, X_{t-2}, X_0. It only has relationship with X_t. Then we have $X_{t+1} = f(X_t)$.

That is, the future state has no relationship with the past state. Markov chain belongs to Markov process, the time and the state of Markov chain are discrete, it can be expressed as :

$$P(X_{t+1} = q_{t+1}|X_t = q_t, X_{t-1} = q_{t-1}, \cdots, X_1 = q_1) = P(X_{t=1} = q_{t+1})$$

Among which, $q_1, q_2, \cdots, q_m \in \{\theta_1, \theta_2, \cdots, \theta_n\}$ are the value of states.

Usually, initial probability vector Π and state transition matrix A are used to describe Markov chain. The initial probability vector $\Pi = \{\pi_1, \pi_2, \cdots, \pi_n\}$,$\pi_i = P(q_1 = \theta_i), 1 < i, j < N$. If state i at time t has k times transitions to state j, the probability is

$$P_{ij}(t, t + k) = P(X_{t+k} = \theta_j|X_t = \theta_i), 1 < i, j < N, k \geq 1$$

If {X(t),t$\in T$} is homogeneous Markov chain, the transition probability is not depend on t, and it only has relationship with i, j and k. So, $P_{i,j}(t, t + k) = P_{ij}(k) = P(X_{t+k} = \theta_j|X_t = \theta_i)$ is the transition probability of k times.

Set $a_{ij} = P_{ij}(t, t + 1)$, we have one-step transition matrix:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{pmatrix}$$

$P = \sum_{j=1}^{N} a_{ij} = 1$, k times transition matrix can be acquired by A^k.

1.2 Hidden Markov Model Theory

By using the example of stock market, we could understand the principle of Hidden Markov model and then introduce the theory of discrete-time Hidden Markov model. The stock market has three states, respectively are bull market, steady market and bear market. According to the variation of stock's price, there will be three observed results, which are rise, invariant and depreciate as shown in the table.

The observation state and hidden state of stock market			
	Performance 1	Performance 2	Performance 3
Observation state	Rise	Invariant	Depreciate
Hidden state	Bull market	Steady market	Bear market

Table 1.1

In Markov model, the states of Markov chain are one to one correspondence with the observation data, which means, if the given observation data is rise-invariant-rise, the corresponding observation state is bull market-steady market-bull market.

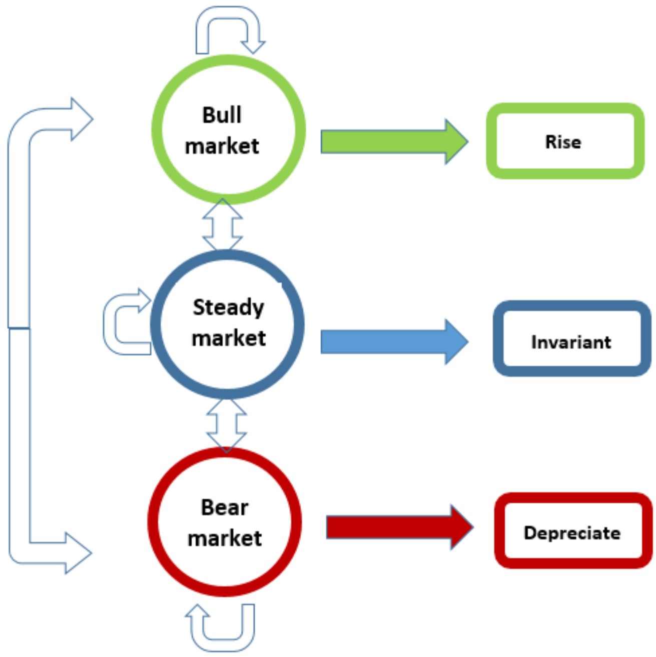

Figure 1.1

Hidden Markov(HMM) model gives a better description of the changing stock market. At a certain state, the observation of stock price change alternately, if the market is bull market, it has more rise states than the depreciate states. So, the Hidden Markov model doesn't exclude situation of invariant and depreciate, which is a big different with Markov model.

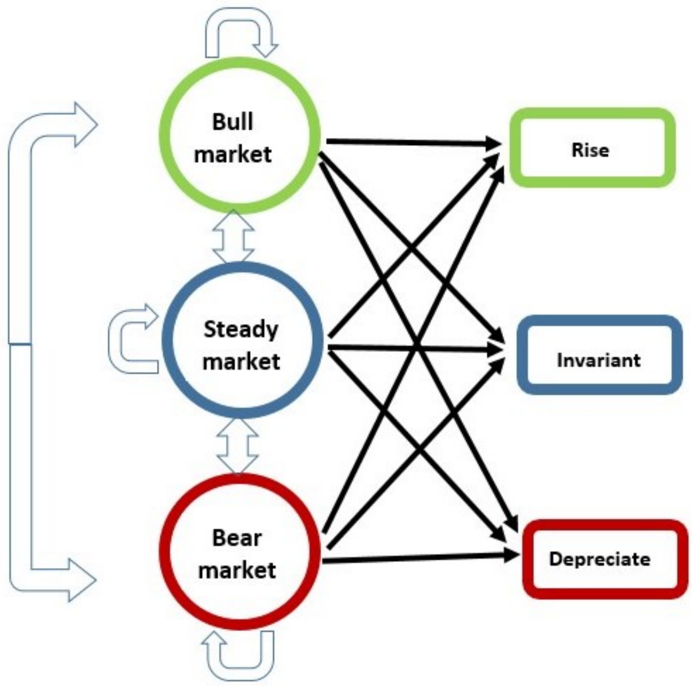

Figure 1.2

In HMM, each hidden state could have three observation states. For example, the stock price of bull market could have probability a_1 with rise, probability a_2 with invariant and probability a_3 with depreciate, in which $a_1 + a_2 + a_3 = 1$. For the same reason, the depreciate state could also have three observation possibility.

The observation state and hidden state have probabilistic relationship. Establish the HMM for this process, the model not only have Markov process hidden in the base course and changing with time, but also have a observation set, which is related with hidden state and observable.

So, if we have given state(bull market, steady market, bear market), we could formulate observation probability distribution matrix:

$$\begin{array}{c} \\ bullmarket \\ steadymarket \\ bearmarket \end{array} \begin{array}{c} \begin{array}{ccc} rise & invariant & depreciate \end{array} \\ \begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix} \end{array}$$

The essential difference between the two models is when the observation state is rise-invariant-depreciate, we still don't know the corresponding status switch, during this time, the status switch is called hidden state. However, we could elicit the most likely hidden state that generate the observation state from the given probability distribution.

1.3 Hidden Markov Model Structure

HMM models is described as follows: Firstly, the state of HMM are hidden states and it is not observable; Also, the observation state and hidden state are not one-to-one correspondence, the hidden state could only obtained by the probability distribution matrix; Last but not least, the HMM is a double-stochastic process, which are the Markov process and dominance random function set. The Markov process describes the transfer process between states, the dominance random function set is a output probability function of observation value, which describe a relationship between hidden state and observation state.

Traditional HMM divided in to discrete-time Markov model and continuous-time Markov model. When the observation distribution is discrete, HMM belongs to discrete-time Markov model, when the observation distribution is continuous, HMM belongs to continuous-time hidden Markov model.

1.3.1 Hidden Markov Model Hypotheses

HMM must satisfy three hypotheses:

1. The first order Markov hypothesis. The future state X_{t+1} only has relationship with current state X_t, it has no relationship with past state X_{t-1}, X_{t-2}, $\cdots$, X_0.

2. Immobility hypothesis. Transition status are not related with time, which means $P(X_{i+1}|X_i) = P(X_{j+1}|x_j)$.

3. Independent observation value hypothesis. The output of the observed value only related with current state.

1.3.2 The fundamental form of Hidden Markov Model

HMM is a double stochastic process, which means it is a combination of Markov process and dominant function set. For discrete-time Hidden Markov Model, let Q be the implicit state process, namely the unobservable discrete-time Markov chain, it is finite state, single step and homogeneous, so $Q = (q_t), (t \in N)$, q_t is the hidden state at time t, O is a observable stochastic process, so $O = (o_t), (t \in N)$. Similarly, we could let $Q^t = (q_1, q_2, \cdots, q_t)$ becomes sequence of hidden states and $O^t = (o_1, o_2, \cdots, o_t)$ becomes sequence of observation states.

HMM can be described by quintet $\mu = (S, V, A, B, \Pi)$, or $\mu = (A, B, \Pi)$.

S is a set of n states, $S = (s_1, s_2, \cdots, s_n)$.

V is a set that contains m observation states, $V = (v_1, v_2, \cdots, v_m)$.

A is a transition probability matrix, $A = \{a_{ij}\}, a_{ij} = P(q_t = j | q_{t-1} = i), 1 \leq i, j \leq N$. a_{ij} is transition probability from state i to state j, satisfied: $a_{ij} \geq 0; \sum_{j=1}^{N} a_{ij} = 1, \forall i, j$.

B is observation probability matrix, $B = \{b_i(k)\}; \sum_{k=1}^{M} b_i(k) = 1, \forall i = 1, 2, \cdots, N$.

Π is probability distribution of original state, $\Pi = \{\pi_1, \pi_2, \cdots, \pi_n\}, \pi_i = P(q_1 = s_i)$. It is the probability of choosing a state at the beginning.

1.3.3 The difference between Markov model and HMM

The difference of parameter setting of Markov model and HMM is shown in the table as follows.

Comparison between the Markov and Hidden Markov Models					
parameter / model	S	V	A	B	Π
Markov Model	√	√	√	X	√
Hidden Markov Model	√	√	√	√	√
√ means the parameter exists in the model. X means the parameter does not exist in the model					

Table 1.2

1.3.4 Continuous-time hidden Markov model

For the continuous-time hidden Markov model, the biggest difference with discrete-time hidden Markov model is that the parameter B is not the same. The parameter B of discrete-time hidden Markov model is a transition probability matrix but in the continuous-time hidden Markov model, B is a observation probability density function.

We could simulate the sample path of continuous-time Markov model first to understand the different between these models. Let P_t become the transition probability, we assume $P_0 = I$, so P_t is the standard transition matrix. Let G becomes the generator of bivariate Markov chain, thus the generator G has non-positive main diagonal elements, non-negative off-diagonal elements, and each of its rows sums to zero. Under this situation, we could get:

$$P_t = e^{Gt}$$

Each jump of the process corresponds to a state transition, and the process remains in each state for a random duration of time. When the chain enters a state z, and $\Delta\tau$ denotes the sojourn time of the chain in that state, then $\Delta\tau$ is exponentially distributed

$$P(\Delta\tau > t|Z(0) = z) = e^{-\lambda(z)t}$$

First step is to create 4 states in this task to simulate. The first step is to create $4 * 4$ matrix I.

Then we could generate G with a reasonable rate. The diagonal of G is non-positive and each line of G sums to zero.

	1	2	3	4
1	-0.8147	0.6638	0.1230	0.0280
2	0.8205	-0.9058	0.0773	0.0080
3	0.0161	0.0141	-0.1270	0.0968
4	0.8343	0.0723	0.0069	-0.9134

Figure 1.3

Set up an axis in Matlab, horizontal axis(x) is "Time(t)" and vertical axis(y) is "State". Firstly, set the maximum value of x as 100. Then we set up the beginning of y at state 1 and simulate the sample path. Here is the plot 1.

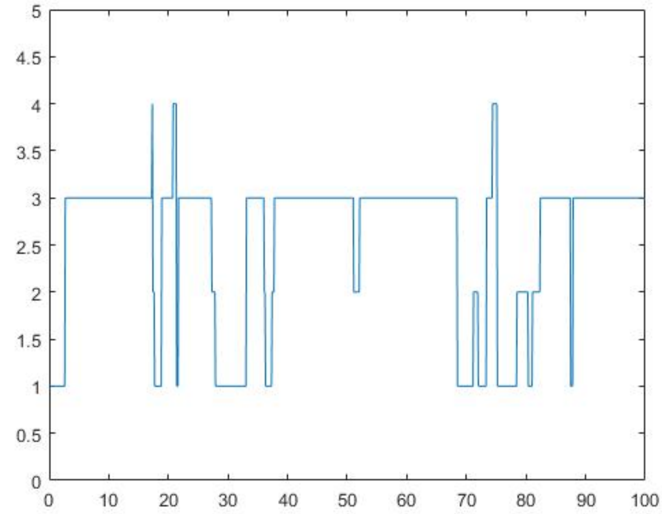

Figure 1.4

Also, we could see the final P_t after many times iteration.

	1	2	3	4
1	0.2933	0.2260	0.4247	0.0560
2	0.2933	0.2260	0.4247	0.0560
3	0.2933	0.2260	0.4247	0.0560
4	0.2933	0.2260	0.4247	0.0560

Figure 1.5

From the table we know that the empirical estimates converge to the true probabilities. For the more obvious graph, we need to add more value to x in order to get better outcome. Changing the maximum value of x to 1000

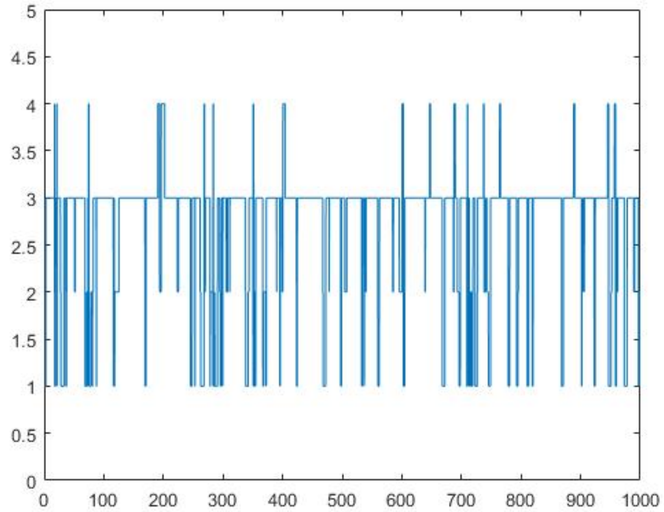

Figure 1.6

From the outcome we know that the state jumps most frequently and mainly remains in state 3, some times it may jumps to state 1 and 2 but shortly jump back to state 3. Also, the state barely jumps to state 4, which corresponds to the probability distribution matrix P_t. Then we could expand the maximum value of x to 10000 and see it more clearly.

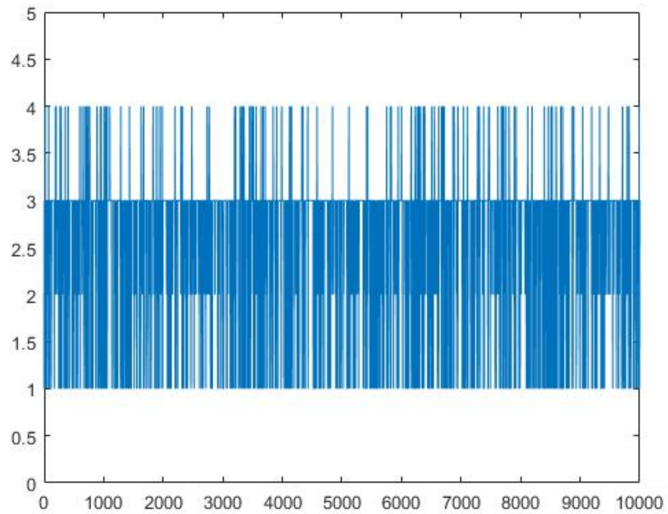

Figure 1.7

So, if we add the transition probability between observation states and hidden states, we could generate the sample path of HMM. Also, the arrival of observations could not be the same as the number of hidden states due to different situations. Sometimes fewer hidden states could generate more observation state and sometimes more hidden state could only lead to few observation states.

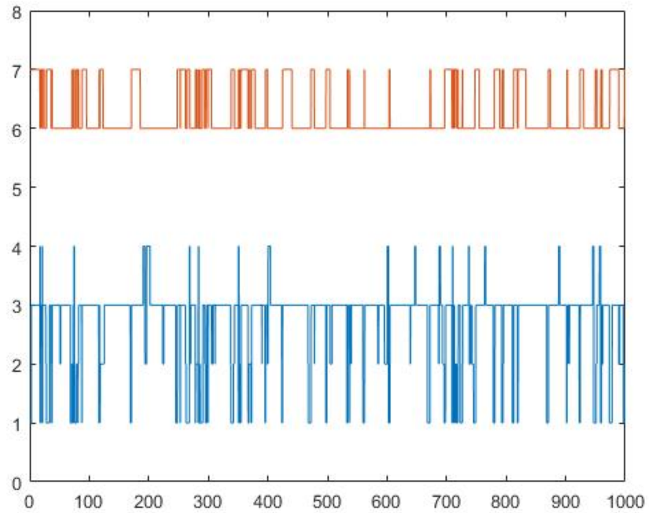

Figure 1.8

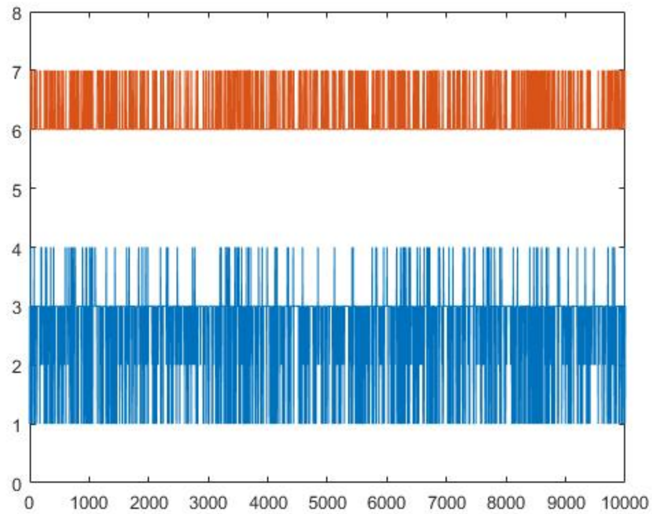

Figure 1.9

1.4 Summary

This chapter introduces the basic theory of hidden Markov model with some examples. It is a basic theory of hidden Markov model, it is also a basic theory of next chapter, which introduces the algorithms of hidden Markov model.

2 Three classical problems of HMM

2.1 Evaluation

Given the observation sequence $O = (o_t)$ and HMM model parameter $\mu = (A, B, \Pi)$, if we know the parameter of HMM, we could get the probability $P(O|\mu)$ of sequence $O = (o_t)$. Evaluation could verify the matching degree of observation sequence and HMM model and it can obtain the best matched model from models that have been trained.

Given the parameter $\mu = (A, B, \Pi)$, for the hidden state $Q^T = (q_1, q_2, \cdots, q_T)$, the probability of generated observation state $O^T = (o_1, o_2, \cdots, o_T)$ is:

$$P(O|Q, \mu) = \prod_{t=1}^{T} P(o_t|q_t, q_{t+1}, \mu) = b_{q_1}(o_1) b_{q_1 q_2}(o_2) \cdots b_{q_{T-1} q_T}(o_T)$$

The Transition probability of hidden states is $P(Q|\mu) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}$.

Observation states O and hidden states Q satisfy $P(O, S|\mu) = P(O|Q, \mu) \cdot P(Q|\mu)$.

Therefore:

$$P(O|\mu) = \sum_{q} P(O|Q, \mu) \cdot P(Q|\mu) = \sum_{q_1 \cdots q_{T+1}} \pi_{q_1} \prod_{t=1}^{T} a_{q_t q_{t+1}} b_{q_t q_{t+1}}(o_t).$$

From this equation we could get all the possible observation states of the given hidden states. Although this equation could get the probability of observation states, the computational efficiency is very general when the length of observation time window increases. Usually, we use Forward-Backward algorithm to solve this problem.

2.2 Decoding

Given the observation states $O^T = (o_1, o_2, \cdots, o_T)$ and parameter $\mu = (A, B, \Pi)$, decoding try to estimate hidden relationship between $\mu = (A, B, \Pi)$ and observation states $Q^T = (q_1, q_2, \cdots, q_T)$. So given $\mu = (A, B, \Pi)$ and $Q^T = (q_1, q_2, \cdots, q_T)$, the probability of hidden state i at time t is:

$$\gamma_i(t) = P(q_t = i|O, \mu) = \frac{P(q_t = i, O|\mu)}{P(O|\mu)}.$$

We use forward probability $\alpha_i(i)$ and backward probability $\beta_i(i)$ to express this probability:

$$\gamma_i(t) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{N} \alpha_i(t)\beta_i(t)}.$$

Therefore, the best states sequence Q' is:

$$Q' = \arg\max_{1 \leq i \leq N} \gamma_i(t), 1 \leq t \leq T+1, 1 \leq i \leq N.$$

Usually we use Viterbi algorithm to solve coding problems.

2.3 Learning

Learning is one of the main problems for Hidden Markov models, the mainly function of learning problem is adjust the parameter of HMM given the observation states $Q^t = (o_1, o_2, \cdots, o_T)$ and initial model $\mu = (A, B, \Pi)$ and make the generated model "learn" the observation states, and then generate the maximum observation probability $P(O|\mu)$. The optimization of related parameters gives the best explanation of the generation of given observation states.

Learning problem could expressed as: $\arg\max_(O|\mu)$. Usually we use Baum-Welch algorithm to solve learning problems as described in the subsequent section.

2.4 Algorithms of HMM

2.4.1 Forward-backward algorithm

Given the observation states $O^T = (o_1, o_2, \cdots, o_T)$ and parameter $\mu = (A, B, \Pi)$, the Forward-backward algorithm is used to calculate the output probability $P(O|\mu)$ of HMM.

Forward-backward algorithm defines the forward variable and backward variable:

Given the HMM, the forward variable $\alpha_t(i)$ is the probability of observation states sequence and hidden state s_i at time t:

$$\alpha_t(i) = P(o_1, o_2, \cdots, o_t, q_t = s_i|\mu); i = 1, 2, \cdots, N; t = 1, 2, \cdots, T.$$

Calculate $P(O^T|\mu)$ by forward recursion algorithm:

(1) Initialization:

$$\alpha_1(i) = P(o_1, q_1 = s_i|\mu) = \pi_i b_i(o_1).$$

(2) Recursion formula:

$$\alpha_{t+1}(j) = \{\sum_{i=1}^{N}[\alpha_t(i)a_{ij}]\}b_j(o_t + 1); j = 1, 2, \cdots, N; t = 1, 2, \cdots, (T - 1).$$

(3) End:

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

Analogously, given parameter $\mu = (A, B, \Pi)$ and hidden state s_i, backward variable β_i is the contingent probability of having time $T + 1$ to final time T if observation is $O^T = (o_{t+1}, 0_{t+2}, \cdots, o_T)$, that is:

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \cdots, o_T)|q_t = s_i, \mu).$$

Calculate $P(O^T|\mu)$ by backward recursion algorithm:

(1) Initialization:

$$\beta_T(i) = 1, i = 1, 2, \cdots, N.$$

(2) Recursion formula:

$$\beta_t(i) = \sum_{j=1}^{N}[a_{ij}b_j(0_{t+1})]\beta_{t+1}(j); t = (T - 1), (T - 2), \cdots, 1; i = 1, 2, \cdots, N.$$

(3) End:

$$P(O^T|\mu) = \sum_{j=1}^{N} \beta_1(i) \cdot \pi_i \cdot b_i(o_1).$$

2.4.2 Viterbi algorithm

Given the observation sequence O^T and parameter $\mu = (A, B, \Pi)$, based on the dynamic programming algorithm, Viterbi algorithm is used to solve the problem of state sequence Q^T. Viterbi algorithm could get state transition path and outcome probability of the path. It is the best algorithm for solving decoding problems and the complexity of Viterbi algorithm is $O(M^2T)$, M is the number of states, T is the length of observation sequence.

Given the parameter $\mu = (A, B, \Pi)$, all the hidden Markov state sequence with state s_i at time t and the highest possibility of $(o_{t+1}, o_{t+2}, \cdots, o_T)$. Viterbi algorithm could find best hidden states. Denote variable $\delta_t(i)$:

$$\delta_t(i) = \max_{\{q_1, q_2, \cdots, q_{t-1}\}} P(q_1, q_2, \cdots, q_{t-1}, q_t = s_i, o_1, o_2, \cdots, O_t|\mu);$$

$$i = 1, 2, \cdots, N; t = 1, 2, \cdots, T.$$

The calculation is as follows:

(1) Initialization:

$$\delta_1(i) = \pi_i b_i(o_1); \psi_1(i) = 0; i = 1, 2, \cdots, N.$$

(2) Recursion formula:

$$\delta_t(j) = [\max_{\{1 \leq i \leq N\}} (\delta_{t-1}(i) a_{ij})] b_i(o_t); j = 1, 2, \cdots, N; t = 1, 2, \cdots, T.$$

$$\psi_t(j) = \arg\max_{\{1 \leq i \leq N\}} [\delta_{t-1}(i) a_{ij}]; j = 1, 2, \cdots, N; t = 1, 2, \cdots, T.$$

(3) End:

$$P^* = \max_{\{1 \leq i \leq N\}} [\delta)_T(i)]; q_T^* = \arg\max_{\{1 \leq i \leq N\}} [\delta_T(i)].$$

(4) Deduce the best states sequence:

$$q_t^* = \psi_{t+1}(q_{t+1}^*); t = (T-1), (T-2), \cdots, 1.$$

2.4.3 Baum-Welch algorithm

Learning problem also called parameter evaluation problem, the calculation process is more complex than the first two problems. Baum-Welch algorithm is widely used in learning problem. The theory of this algorithm is that given the observation states $O^T = (o_1, o_2, \cdots, o_T)$, calculate the corresponding given parameter $\mu = (A, B, \Pi)$ that makes the max value of $P(O|\mu)$. Baum-Welch algorithm is an iteration based on the EM algorithm, at the beginning the values are the empirical estimates of parameters, the algorithm iterate parameter and finally obtain the best parameter of the models. The Baum-Welch algorithm can be described as follows:

(1) Initialization:
$\pi_i = \gamma_1(i)$ means the expected value of s_i when $t = 1$, $\mu = (A_0, B_0, \Pi)$;

(2) Recursion formula:
Given the parameter and observation states, the conditional probability that HMM is s_i at time t and then become s_j at time $t+1$ is $\varsigma_t(i, j)$:

$$\varsigma_t(i,j) = \frac{P(q_t = s_i, q_{t+1} = s_j, O|\mu}{P(O|\mu)} = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}.$$

The probability of being s_i at time t is: $\gamma_t(i) = \sum_{j=1}^{N} \varsigma_t(i,j)$.

The expected number of transitions from s_i: $\sum_{t=1}^{T-1} \gamma_t(i)$.

The expected number of transitions from s_i to s_j: $\sum_{t=1}^{T-1} \varsigma_t(i,j)$.

The formulation can be described as follows:

$$\tilde{a}_{ij} = \frac{\sum_{t-1}^{T-1} \varsigma_t(i,j)}{\sum_{t-1}^{T-1} \gamma_t(i)}, \tilde{b}_j = \frac{\sum_{t=1,o_t=v}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}.$$

Terminal condition:

$$|\log P(O|\mu) - \log P(O|\mu_0)| < \varepsilon.$$

Where ε is a pre-set tolerance threshold.

3 Poisson Hidden Markov Model

3.1 Poisson Hidden Markov Model Hypotheses

Poisson hidden Markov models(PHMM) are special hidden Markov models. It is discrete-time stochastic process $\{(X_t; Y_t)\}$. They also have unobservable finite state Markov chain $\{X_t\}$ and observation sequence $\{Y_t\}$, which are random variables and depending on $\{X_t\}_{t\in N}$. The PHMM have all the character that HMM have. Furthermore, $\{Y_t\}$ is different from HMM. It is a sequence of conditionally independent random variables. So, we could assume that,Y_t given a state of X_t is a Poisson random variable for every t. That's why we call them Poisson hidden Markov models, in this situation, X_t determines the Poisson parameter that used to generate Y_t.

For the Poisson hidden Markov models, we could have following assumptions. First of all, the hidden states X_t is a discrete Markov chain, which is homogeneous, irreducible, periodic and it has a finite state space. For $S_X = \{1, 2, \cdots, m\}$, we could have the transition probability from state i at time $t-1$ to state j at time t as:

$$\gamma_{ij} = P(X_t = j|X_{t-1} = i) = P(X_2 = j|X_1 = i)$$

.

Also, we could get transition probability matrix $\Gamma = [\gamma_{ij}]$. For any $i \in S_X$, the matrix is $(m \times m)$ and $\sum\limits_{i\in S_X} \delta_i = 1$. The most important parameter of Poisson hidden Markov model is the observation sequence $\{Y_t\}$. In Poisson hidden Markov models, any observed variable Y_t conditioned on X_t is Poisson process for any t, which means, when X_t is in state i, the conditional distribution of Y_t is a Poisson random variable and the parameter of PHMM can be denoted as λ_i. The biggest difference between Poisson Markov models and classic one is the emission probability matrix between hidden states and observation states. That is, given any $y \in N$, the state-dependent probabilities are given by:

$$\pi_{y,i} = P(Y_t = y|X_t = i) = e^{-\lambda_i}\frac{\lambda_i^y}{y!}.$$

Since $\sum_{y\in N} \pi_{yi} = 1$ for every $i \in S_X$ is true, hidden states X_t and observation states Y_t are strongly stationary, for every t, Y_t has the same marginal distribution:

$$P(Y_t = y) = \sum_{i\in S_X} P(Y_t = y, X_t = i) = \sum_{i\in S_X} P(Y_t = y|X_y = i)P(X_t = i) = \sum_{i\in S_X} \delta_i \pi_{yi}.$$

This is a finite mixture of Poisson distributions, and we could easily get the expected value of Y_t for every t, that is:

$$E(Y_t) = \sum_{i \in S_X} \delta_i \lambda_i.$$

Usually we do not know the underlying chain and the Poisson rate λ, we need to estimate the related parameters to build the models. That's why usually people need to build their own model when they try to estimate data with Poisson hidden Markov models.

3.2 Parameter Estimation of Poisson Hidden Markov Model

In order to estimate the parameters of Poisson Markov model, we first need to create initial distribution, denote $\delta = (\delta_1, \delta_2, \cdots, \delta_m)$, also we need transition probabilities γ_{ij} and state dependent probabilities π_{yi}.

Also, we need to find the estimators of these parameters. In particular, we need maximum likelihood estimators of the $m^2 - m$ transition probabilities γ_{ij} with $i \neq j$. Because the row of Γ sum to one, the diagonal elements could obtained by $\gamma_{ii} = 1 - \sum_{j \in S_X, j \neq i} \gamma_{ij}$. So we could estimate off-diagonal elements and maximum likelihood estimator of the m Poisson parameters λ_i entering the state-dependent probabilities π_{yi}. The estimated matrix Γ could help us get the estimator of the initial distribution δ by equality $\delta' = \delta' \Gamma$.

Denote ϕ as the unknown parameters to be estimated by the maximum likelihood method, we have:

$$\phi = (\gamma_{12}, \gamma_{13}, \cdots, \gamma_{mm-1}, \lambda_1, \lambda_2, \cdots, \lambda_m).$$

Let Φ be the parameter space.

Since we don't know the unobservable states $X_t = (x_1, x_2, \cdots, x_T)$, the vector y of the observation states $Y = (y_1, y_2, \cdots, y_t)$ is incomplete. So under this situation, if i denotes the times, the vector of complete data should be $(i_1, y_1, i_2, y_2, \cdots, i_T, y_T)$.

Denote the likelihood function of complete data of joint probability of T unobservable states and observable states as $L_T^c(\phi)$. Because of the Markov characteristic, we could get:

$$L_T^c = P(Y_1 = y_1, \cdots, Y_t = y_t, X_1 = i_1, \cdots, X_t = i_t) = \delta_{i1} \pi_{y_1 i_1} \prod_{t=2}^{T} \gamma_{i_{t-1} i_t} \pi_{y_t i_t}.$$

If we sum over $i_1, i_2, \cdots, i_t$ both sides, we could obtain the likelihood function of incomplete data:

$$L_T(\phi) = P(Y_1 = y_1, Y_2 = y_2, \cdots, Y_T = y_T) = \sum_{i_1 \in S_X} \sum_{i_2 \in S_X} \cdots \sum_{i_T \in S_X} \delta_{i_1} \pi_{y_1 i_1} \prod_{t=2}^{T} \gamma_{i_{t-1} i_t} \pi_{y_t i_t}.$$

Where :

$$\pi_{y_t i_t} = e^{-\lambda_{i_t}} \frac{\lambda_{i_t}^{y_t}}{y_t!}.$$

Since the function is very complex, it is hard to estimate ϕ. However, we have EM algorithm for the Poisson Hidden Markov Model. It is based on the iteration of two steps: the first step, E step, which means the Expectation, and the second step, M step, means the Maximization.

Denote:

$$Q(\phi, \phi') = E_{\phi'}(\ln L_T^c(\phi)|y).$$

Suppose we have k^{th} iteration:

$$\phi^{(K)} = (\gamma_{12}^{(k)}, \gamma_{13}^{(k)}, \cdots, \gamma_{mm-1}^{(k)}, \lambda_1^{(k)}, \cdots, \lambda_m^{(k)}).$$

At the $(k+1)^{th}$ iteration:
E step, compute:

$$Q(\phi; \phi^{(k)}) = E_{\phi^{(k)}}(\ln L_T^c(\phi)|y).$$

M step, find $\phi^{(k+1)}$ that maximize $Q(\phi; \phi^{(k)})$:

$$Q(\phi^{(k+1)}; \phi^{(k)}) \geq Q(\phi; \phi^{(k)}).$$

We should repeat EM steps until the sequence of log-likelihood values $\{\ln L_T(\phi^{(k)})\}$ converges, which means $\ln L_T(\phi^{(k+1)}) - \ln L_T(\phi^{(k)})$ is less than or equal to a very small arbitrary value.

The estimation is very complex, in order to implement the algorithm, the EM algorithm could be simplified by using forward and backward algorithm, so the forward probability could be shown as:

$$\alpha_t(i) = P(Y_1 = y_1, \cdots, Y_t = y_t, X_t = i).$$

Usually we could estimate the parameters by:

$$\alpha_1(i) = \delta_i \pi_{y_1 i}, i = 1, 2, \cdots, m.$$

$$\alpha_t(j) = (\sum_{i \in S_X} \alpha_{t-1}(i)\gamma_{ij})\pi_{y_t j}, t = 2, \cdots, T, j = 1, 2, \cdots, m.$$

Also, we could have backward probability:

$$\beta_t(i) = P(Y_{t+1} = y_{t+1}, \cdots, Y_T = y_T | X_t = i).$$

The probabilities of $\beta_t(i)$ could be obtained recursively as follows:

$$\beta_T(i) = 1, i = 1, 2, \cdots, m.$$

$$\beta_t(i) = \sum_{j \in S_X} \pi_{y_{t+1} j} \beta_{t+1}(j)\gamma_{ij}, t = T-1, \cdots, 1, i = 1, 2, \cdots, m.$$

The maximum likelihood estimator of $\gamma_i j$ at $(k+1)^{th}$ iteration of EM algorithm can be shown as:

$$\gamma_{ij}^{(k+1)} = \frac{\sum_{t=1}^{T-1} \alpha_t^{(k)}(i)\gamma_{ij}^{(k)}\pi_{y_{t+1}}^{(k)}\beta_{t+1}^{(k)}(j)}{\sum_{t=1}^{T-1} \alpha_t^{(k)}(i)\beta_t^{(k)}(i)}.$$

Also, we could get the estimator of λ_i for any different states i and j:

$$\lambda_i^{(k+1)} = \frac{\sum_{t=1}^{T} \alpha_t^{(k)}(i)\beta_t^{(k)}(i)y_t}{\sum_{t=1}^{T} \alpha_t^{(k)}\beta_t^{(k)}(i)}.$$

The EM algorithm considers m number of states as known and fixed. However, in the application, we usually don't know the exact value of m, the estimation of m is difficult. Fortunately, we could use log-likelihood to estimate m. Two algorithm are used to measure m, the Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC). We use following versions of AIC and BIC as model selection criteria for Poisson hidden Markov model, that is, $AIC(m) = -2l(m) + 2m^2$ and $BIC(m) = -2l(m) + m^2 \log T$, $l(m)$ denotes the maximized log-likelihood for a model with m components; T was defined above. In the algorithms, we select number of components to be the number that minimizes $AIC(m)$ and $BIC(m)$.

4 Estimating the Economic Performance of Dongfeng Motor Corporation

Here we have the Data from Dongfeng Motor Corporation, according to their sales from 2009-2017, we found a very interesting fact:

2016-2017 Dongfeng Motor Corporation Sales			Unit: 10 thousands
Month	China automobile sales	Dongfeng sales	Market share
9	17.27	4.59	0.27
10	15.48	4.35	0.28
11	17.26	5.54	0.32
12	23.11	6.94	0.3
1	24.68	6.43	0.26
2	26.65	7.22	0.27
3	18.44	5.76	0.31
4	17.54	5.86	0.33
5	17.78	5.42	0.3
6	12.55	3.25	0.26
7	11.57	2.98	0.26
8	12.86	2.79	0.22

Table 4.1

We cut out the data from 2016 September to 2017 August, because the data is very representative. We hypothesis that the sales volume exhibits seasonality:

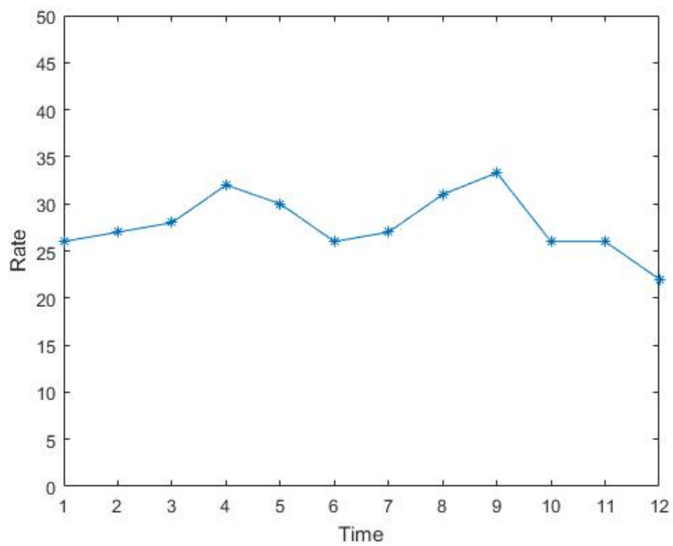

Figure 4.1

We found that, the sales of every 3 months is relatively stable, but is different with others. In the previous introduction of continuous-time hidden Markov model, we assume that the transition rate matrix G is not influenced by other factors. However, in the real situation, some factors may influence the changing rate of G. We made a market study and found that in the second half of the year the cars manufacture introduces new models, which would make the sales in a very considerable state. Also, at the end of the year, there are many sales promotion and makes the sales reaching the summit, then the infrastructure boom slows down and goes into underestimate. However, this explanation is only hypothetical.

According to the table, we made a assumption that the sales of Dongfeng Motor Corporation has some rules to follow, it is medium-good-medium-bad in each fiscal year. The sales of each season may affect the transition rate G. In this situation, we change the model in to a regulating process, if the sales is good, we will increase the value of G, on the contrary, if the sales is bad, we will decrease the value of G.

Suppose the sales volume situation is medium-good-medium-bad, respectively. We made a simulation of the states in the continuous-time Markov model. Firstly we made G increasing and decreasing prominently. If sales volume is in good regime, G would expand 1.8 times, if sales volume is in bad regime, G would reduce to 0.2 times. The graph is shown below:

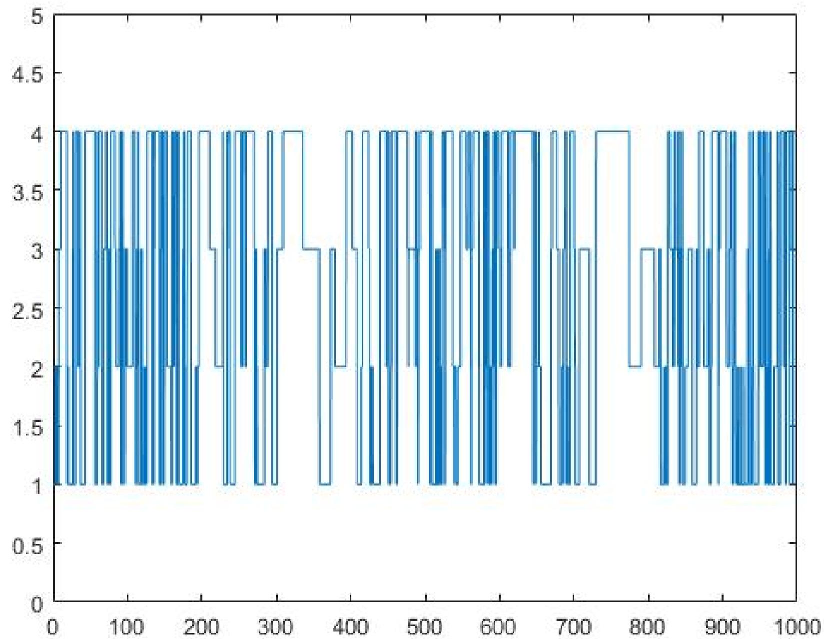

Figure 4.2

In the graph the $0-100$ means medium, $100-200$ means good, $200-300$ means medium and $300-400$ means bad, respectively. The graph shows that the state is very unstable in good stage and its too stable in bad stage. We found that it is not very conform to the real model. Then we change the changing rate of G. If the sales is in good stage, G would expand 1.05 times, if the sales is in bad stage, G would reduce to 0.95 times.

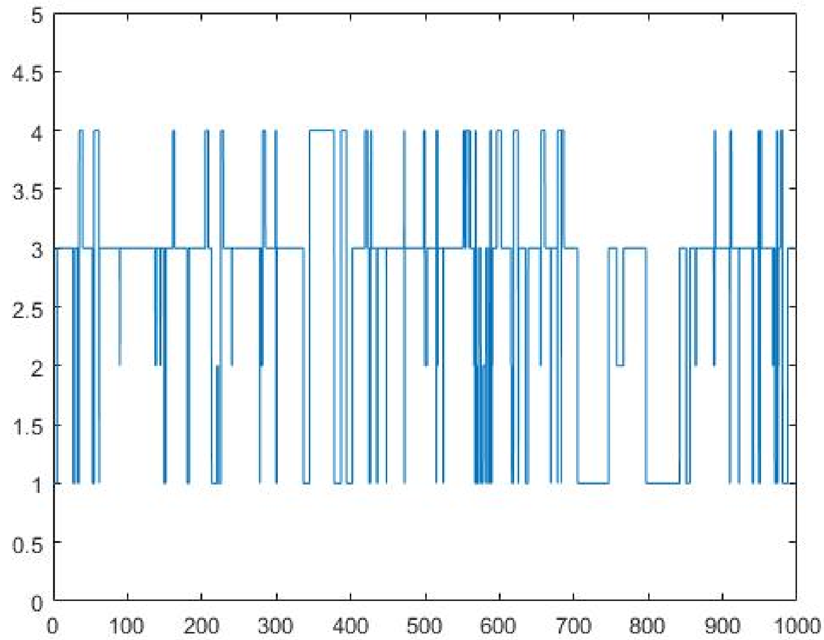

Figure 4.3

From the graph we can see that the difference between the good stage and bad stage is not very prominent, but we still can found the difference between them. Since the observation sequence has 4 states, which are high, little high, little low, low, we basically divided hidden states into two parts, good and bad. Using the new adjusted continuous-time hidden Markov model, we made a simulation of the economic performance of Dongfeng Motor Corporation on the monthly sales part:

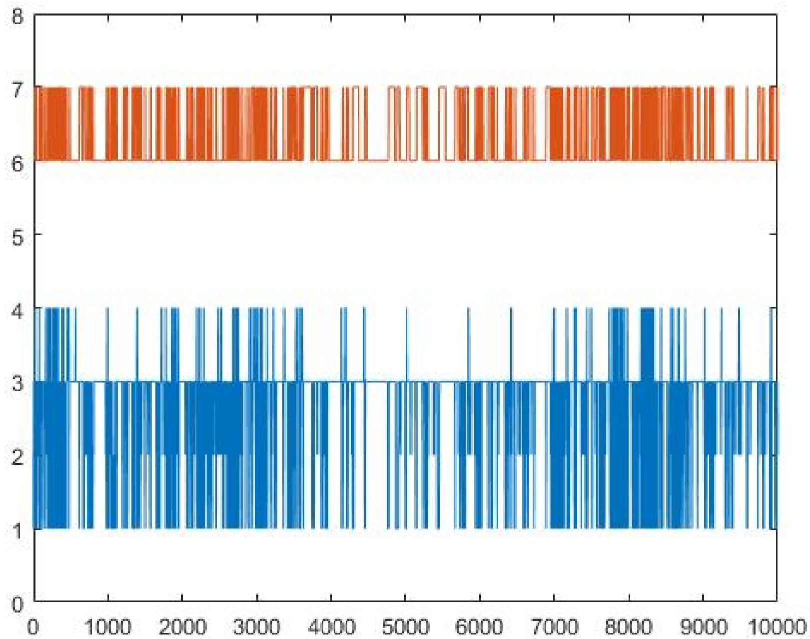

Figure 4.4

The model and parameter are very basic, we could have more detailed estimation by using some complex factors and considering other situations.

5 Future Work

Here is the brief introduction of our future work on the estimation. Since we got many kinds of data, different models can be used in the estimation.

Here is the data of market share of Dongfeng Motor Corporation.

2005-2016 Dongfeng Market Share			Unit: 10 thousands
Year	China automobile sales	Dongfeng sales	Market share
2005	72.66	19.07	26.25
2006	98.35	24.91	25.33
2007	124.53	40.06	32.17
2008	130.82	43.51	33.26
2009	211.73	62.55	29.54
2010	293.33	123.77	31.47
2011	294.64	82.38	27.96
2012	304.96	62.61	20.53
2013	330.61	65.03	19.67
2014	277.49	81.42	29.34
2015	243.03	53.1	21.85
2016	234	69.33	29.63

Table 5.1

From the data we know that both Donfeng and Chinese market grow every year. But the market share of Dongfeng Motor Corporation fluctuates. Under this situation, we could use Hidden Markov model to estimate the market share of Dongfeng.

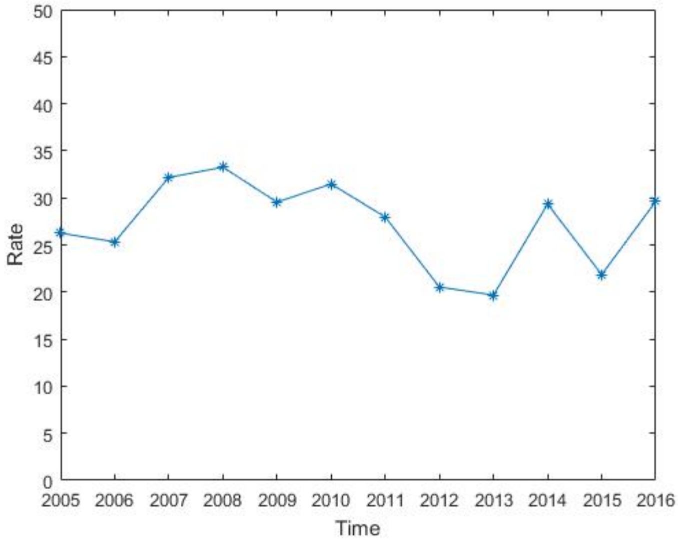

Figure 5.1

Although the variance is not large, for the economic performance, the gap is significant. Also, we should set a very sensitive boundary between each hidden stage to make estimation.

What is more, we could use the stock market of Dongfeng Motor Corporation to estimate its economic performance.

1.Sample selection:
Choosing the observation sequence of the model, including the opening price, closing price, top price and minimum price, showing as:
observation sequence = [openning price, closing price, top price, minimum price]

2.Confirming the number of hidden states:
Choosing 2, 3, 4, 5 as the number of hidden states and using Odd-Even-Half-Sampling method to do the simulation test, finding the exact number of hidden states. Divide the observation in to two part, one for odd numbers and another for even numbers. For the odd numbers, randomly generate 15 initial distribution, transition probability matrix and mix normal distribution, implementing EM algorithm for training parameters estimation for 100 times iteration.

3.Parameters estimation: Implementing hmmtrain model in Matlab to estimate the initial probability distribution, transition probability matrix and approximate probability density function for mixed normal distribution. After iterations the training parameters could be obtained.

4.Forecasting: After all the preparation work finished, the logarithmic likelihood value of historical data is obtained by implementing the function $mhmm_logprob$ in Matlab to find the closest likelihood value to the same day, and forecast the stock price by single day and multi day weighted forecasting, respectively.

References

[1] Yariv Ephraim and Brian L. Mark , Bivariate Markov Processes and Their Estimation, Vol. 6, No. 1 (2012) 195

[2] L. E. Baum and T Petrie, Statistical inference for probabilities functions of finite state Markov chains, Ann. Math. Stat, 1966, 37: 1554-1563

[3] F. Jelinek. A fast sequential decoding algorithm using a stack, IBM J, Res. Develop, 1969, 13: 675-685

[4] Baum, L.E. (1972). "An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of a Markov Process". Inequalities. 3: page $1-8$.

[5] M. Y. Boudaren, E. Monfrini, and W. Pieczynski, Unsupervised segmentation of random discrete data hidden with switching noise distributions, IEEE Signal Processing Letters, Vol. 19, No. 10, pp. 619-622, October 2012

[6] Chandima Piyadharshani Karunanayake, multivariate Poisson hidden Markov models for analysis OF spatial counts, Copyright Chandima Piyadharshani Karunanayake, June 2007

[7] Madalina Olteanu and James Ridgway, Hidden Markov models for time series of counts with excess zeros, Bruges (Belgium), 25-27 April 2012

[8] Fatemeh Sarvi, Azam Nadali, Mahmoud Khodadost, Melika Kharghani Moghaddam, and Majid Sadeghifar, Application of Poisson Hidden Markov Model to Predict Number of PM2.5 Exceedance Days in Tehran During 2016-2017, Published online 2017 June 30

[9] K. Orfanogiannaki, D. Karlis, G. A. Papadopoulos, Identifying Seismicity Levels via Poisson Hidden Markov Models, Pure Appl. Geophys. 167 (2010), 919931

[10] Majid Sadeghifar, Maryam Seyed-Tabib, Saiedeh Haji-Maghsoudi, Kourosh Noemani, Fariba Aalipur-Byrgany, The application of Poisson hidden Markov model to forecasting new cases of congenital hypothyroidism in Khuzestan province, Journal of Biostatistics and Epidemiology 2016. 2(1):14-19

[11] Junko Murakami, Bayesian Posterior Mean Estimates for Poisson Hidden Markov Models, Computational Statistics Data Analysis, vol. 53, no. 4, pp. 941-955, Feb. 2009

[12] Roberta Paroli, Giovanna Redaelli, Luigi Spezia, Poisson hidden Markov models for time series of overdispersed insurance counts

[13] Albert, P. S. (1991). A Two-State Markov Mixture Model for a Time Series of Epileptic Seizure Counts. Biometrics, 47, 1371-1381

[14] Dempster,er A. P., Laird N. M., Rubin D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with Disciission). Journal of the Royal Statistical Society, Series B, 39, 1-38.

[15] Giudici P., Ryden T., Vandekerkhove P. (1998). Likelihood ratio tests for hidden Markov models. Technical report, 1998: 19, Lund University, Sweden

[16] Leroux B. G. and Puterman M. L. (1992). Maximum-Penalized-Likelihood Estimation for Independent, and Markov-Dependent, Mixture Models. Biometrics, 48, 545-558

Appendix

```
% The first code is used to generate the most probable path between
% states given the observation sequences, transition probability between
% states and observations and transition probability between states.



%Firstly, we set up the transition probabilities between states and
% emission probabilities between observations and states. Here we have
% 2 states and the initial probability of them are both 0.5.



trans = [0.5,0.5;
          0.4,0.6];
emis = [1/5 1/5 1/5 1/5 1/5;
        1/8 1/8 1/8 1/8 1/2];

start= [0.5, 0.5];

[seq,states] = hmmgenerate(10,trans,emis);

% Now we have obervation sequences, transition probability between states
and
% observations, transition probablity between states.The task is using
Verterbi
% algorithm to compute probable path between states.
%  1. The probability of most probable path ending in state X with
observation
%  Y is: P_l(i,x)=p_l(i)*max_k(p_k(j,x-1)*p_kl)
%  Where:
%      p_l(i) = probablity to observe element i in state l.
%      p_k(j,x-1) = probability of the most probable path ending at position x-
                  1 in state k with element j.
%      p_kl = probability of the transition from stake l to state k.
%  So we can compute recursively the probability of the most probable path
%  (from the first to the last element of our sequence).



% step 1 do log 2 to trans and emission matrix trans
= log2(trans);
```

```
emis = log2(emis);


P51 = -1 + emis(1,5);
P52 = -1 + emis(2,5);

path = zeros(2,10); esstate
= zeros(10,1); path(1,1) =
P51; path(2,1) = P52;
if path(1,1)>path(2,1)
        esstate(1) = 1;
else
        esstate(1) = 2;
end

for i = 2:10
    path(1,i)        =        path(esstate(i-1),i-1)+trans(esstate(i-
1),1)+emis(1,seq(i));
    path(2,i)        =        path(esstate(i-1),i-1)+trans(esstate(i-
1),2)+emis(2,seq(i));
    if path(1,i)>path(2,i)
            esstate(i) = 1;
    else
            esstate(i) = 2;
    end
end
```

% The second code is used to simulate the continuous-time Hidden Markov Model with the change of transition rate matrix G.

% Firstly we generate the transition probability matrix, transition rate matrix.

```
rng(0); Z
= 4;
X = 4;
Pt = zeros(X,Z);
P_0 = eye(X,Z);
G = zeros(X,Z);
lambda = ones(1,4);
lambda = lambda/5;
detel_tao = 0;
for i = 1:Z
        sm = -1*rand();
        temp = -sm;
        for j = 1:Z
              if j == i
                    G(i,j) = sm;
               else
                    if i == Z && j == Z-1
                          continue;
                    end
                    if j == Z
                          G(i,j) = temp;
                    else
                         G(i,j) = temp*(-sm);
                         temp = temp - G(i,j);
                    end
              end
        end

        if i == Z
              G(i,Z-1) = temp;
        end
  end

tmax = 10000;
dt = 0.1;
x= dt:dt:tmax;
m = 1/dt;
```

```
y = ones(1,tmax*m);
i = 1;
flag = 0;
P_a = zeros(Z,Z);

% Initial the variable to compute the hidden states emis =

[1/4 1/4 1/4 1/4 ;
        1/3 1/8 7/24 1/4];

trans1 = log2(Pt);
emis = log2(emis);

P1 = -1 + emis(1,1);
P2 = -1 + emis(2,1);

path = zeros(2,tmax*m);
esstate = zeros(tmax*m,1);

path(1,1) = P1;
path(2,1) = P2;
if path(1,1)>path(2,1)
      esstate(1) = 1;
else
      esstate(1) = 2;
end


% According to the different situation, we change the transition rate matrix G.

while i <= tmax*m
    Pt = expm(G*i*dt);
    Pt
    period111 = floor(mod(i,4000)/1000);
    switch period111
        case 0
            G = G;
        case 1
            G = G * 1.05;
        case 2
            G = G;
        case 3
            G = G * 0.95;
```

```
    end
    period111

  % Generate delta_tao.

 delta_tao = round(m*exprnd(1/-G(y(i),y(i))));



   trans1 = log2(Pt);
    for j = 1:Z
        sm = 0;
        for k = 1:Z
            sm = sm + Pt(j,k);
            P_a(j,k) = sm;
        end
    end

    trans = P_a;
    for j = i:i + delta_tao y(j)
        = y(i);
         esstate(j) = esstate(i);
        if j>=tmax*m
            flag = 1;
            break;
        end
    end

    if flag == 1
        break;
    end
    i = i + delta_tao;
    check = rand();
    for j = 1:Z
        if check <= trans(y(i),j)
            y(i+1) = j;
            break;
        end
    end


    %compute the hidden states by trans and emission matrix

        if i>=2
        path(1,i)         =          path(esstate(i-delta_tao),i-
delta_tao)+trans1(esstate(i-delta_tao),1)+emis(1,y(i));
        path(2,i)         =          path(esstate(i-delta_tao),i-
```

```
delta_tao)+trans1(esstate(i-delta_tao),2)+emis(2,y(i));
                    if  path(1,i)>path(2,i)
                         esstate(i)  =  1;
               else
                         esstate(i)  =  2;
               end
        end


        i=i+1;
        path(1,i)  =  path(1,i-1); path(2,i)
        =  path(2,i-1); esstate(i)  =
        esstate(i-1);
end



plot(x,y); hold  on;
plot(x,esstate+5);
ylim([0  8]);
```

Biography

Xuecheng Yin received the B.Eng. degree in mechanical engineering from Huaiin Institute of Technology(HIT), Huaian, China, in 2015. He would finish the M.S degree within the Department of Industrial Systems Engineering at Lehigh University at May 2018 and will going to New Jersey Institute of Technology studies for his doctor degree.