VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE
Fakulta financí a účetnictví
katedra bankovnictví a pojišťovnictví

DIPLOMOVÁ PRÁCE

2023 Bc. Adam Pešek

VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE
Fakulta financí a účetnictví
katedra bankovnictví a pojišťovnictví
studijní program: Finanční inženýrství

Hidden Markov Models for Cryptocurrency Trading

Autor práce: Bc. Adam Pešek

Vedoucí práce: Ing. Milan Fičura PhD.

Rok obhajoby: 2023

Čestné prohlášení

Prohlašuji, že jsem diplomovou práci „Uveďte název práce“ vypracoval samostatně a veškerou použitou literaturu a další prameny jsem řádně označil a uvedl v přiloženém seznamu.

V Praze dne 5. května 2023

Acknowledgements

And I would like to acknowledge ...

Abstract

Hidden Markov model (HMM) is a statistical signal prediction model, which has been widely used to predict economic regimes and stock prices. In this paper, we introduce the application of HMM in trading stocks (with S&P 500 index being an example) based on the stock price predictions. The procedure starts by using four criteria, including the Akaike information, the Bayesian information, the Hannan Quinn information, and the Bozdogan Consistent Akaike Information, in order to determine an optimal number of states for the HMM. The selected four-state HMM is then used to predict monthly closing prices of the S&P 500 index. For this work, the out-of-sample R^2_{OS}, and some other error estimators are used to test the HMM predictions against the historical average model. Finally, both the HMM and the historical average model are used to trade the S&P 500. The obtained results clearly prove that the HMM outperforms this traditional method in predicting and trading stocks.

Table of contents

List of figures **xi**

List of tables **xiii**

1 Markov Processes **1**
1.1 Discrete-time Markov Chains 1
1.1.1 Classification of states 5
1.1.2 Stationary distribution 6
1.1.3 Cryptocurrency market movements I. 7
1.2 Continous-time Markov Chains 10
1.2.1 Cryptocurrency market movements II. 12
1.3 Hidden Markov Model 12
1.3.1 Forward and Backward algorithm 14
1.3.1.1 Forward algorithm 15
1.3.1.2 Backward algorithm 17
1.3.2 Viterbi algorithm 18
1.4 Kalman Filters 22

2 Parameter Estimation for Hidden Markov Models **23**
2.1 Expectation–Maximization algorithm 23
2.1.1 Baum-Welsch algorithm 30
2.2 Observable states 33
2.3 Moving Average Convergence Divergence 34
2.4 Stochastic Oscillator 35
2.5 Chaikin Oscillator 37
2.6 Relative Strength Index 39
2.7 Aroon Indicator 39

3 Mixture Models **43**
3.1 Mixture Distributions 43
3.2 Gaussian Mixture Models 44
3.2.1 Motivating EM algorithm 45
3.2.2 EM algorithm for GMM 46
3.2.3 Markov Chain Monte Carlo 47
3.2.3.1 Metropolis-Hastings Algorithm 47
3.2.3.2 Gibbs Sampling 49

4 Proposed versions of Hidden Markov Models **51**
4.1 Gaussian HMM 51
4.1.1 Poisson HMM 51
4.1.2 Context-sensitive HMM 51

5 Trading strategies based on Hidden Markov Models **53**
5.1 Market regimes and trading strategies 53
5.1.1 Market states 53
5.1.2 Trading strategies 53

Appendix A Maximum likelihood results **57**

Appendix B Python code **59**

List of figures

1.1 BTC-USD daily close prices from 23rd August 2020 to 15th May 2023 obtained from Binance. [1] 8
1.2 Distribution of the log returns with respect to predefined market states. [1] . 9
1.3 An HMM with 4 hidden states which emits 2 discrete observable states denoted by x_1 or x_2. a_{ij} is the probability to transition from state z_i to state z_j. $b_j(x_k)$ is the probability to emit symbol x_k in state z_j. 13
1.4 Trellis of the observation sequence x_1,...,x_5 for the above HMM. As an example, the transition between state z_1 at time t=2 and state z_4 at time t=3 has probability $\alpha_2(z_1)p_{14}\theta_{z_4}(x_3)$, where $\alpha_t(i)$ is the probability to be instate i at time t. 16
1.5 Trellis of the observation sequence x_1, ...,x_5 for the HMM. Bold lines illustrate the most likely sequence found by the Viterbi algorithm. 22

2.1 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with subplot containing two lines for MACD and Signal line and a bar chart as their difference* 25
2.2 *E-step as a problem of minimization of KL divergence represented as a gap between two likelihood functions* 29
2.3 Trellis interpreting the conditional probability of hidden state i at time t using Forward-Backward algorithm. 31
2.4 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with subplot containing two lines for MACD and Signal line and a bar chart as their difference* 36
2.5 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with subplot with line indicating Stochastic Oscillator* 37
2.6 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with subplot indicating Chaikin Oscillator* 38

2.7 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with a subplot containing RSI* 40
2.8 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with subplots containing two lines for Aroon Up/Down Indicator and Aroon Oscillator* 41

List of tables

Chapter 1

Markov Processes

Put here the overall definition for sigma algebra and notation and introduction plus motivation

1.1 Discrete-time Markov Chains

Let $\Omega \neq \emptyset$ and $A \subseteq 2^{\Omega}$ be a σ-algebra on Ω, and P a measure on A with $P(\Omega) = 1$, i.e. P is a *probability measure*. Then the triplet (Ω, A, P) is called a *probability space*. Where Ω denotes a sure event, and it holds that $\forall \omega \in \Omega$ is called an elementary event. Furthermore, $\forall A \in A$ is a random event so that $P(A)$ is a probability of such a random event.

Let I be a countable set and Θ a σ-algebra on I. Each $i \in I$ is called a *state* and (I, Θ) a *state-space*. We say that $\lambda = (\lambda_i : i \in I)$ is a measure on I if $0 \leq \lambda_i \leq \infty$. If in addition the *total mass* $\sum_{i \in I} \lambda_i = 1$ then λ is a *distribution* (or probability measure).

Suppose now that we have two measurable spaces (Ω, A) and (I, Θ) and a random variable $X : \Omega \to I$ assuming that X is measurable. Thus, we call (I, Θ) a state space and (Ω, A) an underlying space. Therefore, we may set:

$$\lambda_X(i) = P(X = i) = P(\{\omega : X(\omega) = i\}) \tag{1.1}$$

Since we are allowing only for the discrete realizations of the random variable X, given previous assumptions, $\lambda_X(i)$ is a *probability mass function*.

Then measure λ defines a distribution of X. Given such a setting random variable X is assumed to denote random state i with probability λ_i.

Let us now assume that we have a sequence of random variables $\{X_t : t \in T\}$ where T is a countable set of time steps. We say that $\{X_t : t \in T\}$ is a *stochastic process* and if it also holds that $T = \mathbb{N}_0$ then *discrete-time stochastic process*. In the context of Markov Chains we call measurable space I as a *state space* and X_t as a *state* at time t respectively. Given the

initial setup, we may define a *discrete-time Markov Chain* as a stochastic process $\{X_t : t \in T\}$ with a state space I and a distribution λ such that for all $t \in T$ and $i_0, i_1, \ldots, i_{t+1} \in I$ it holds that:

$$P(X_{t+1} = i_{t+1} | X_t = i_t, \ldots, X_0 = i_0) = P(X_{t+1} = i_{t+1} | X_t = i_t) \tag{1.2}$$

which holds for all $t \in T$ and $i_0, i_1, \ldots, i_{t+1} \in I$. In other words, the probability of observing a state i_{t+1} at time $t+1$ given the sequence of states $i_0, i_1, \ldots, i_{t+1}$ is equal to the probability of observing a state i_{t+1} at time $t+1$ given only the last observed state i_t. This fundamental relationship is called *Markov property*, and it is a consequence of the *memoryless property* of Markov Chains. Conditional property of Markov Chains may be equivalently expressed using current state as $i \in I$ and a previous state $j \in I$ as:

$$P(X_{t+1} = i | X_t = j) = p_{j,i}(t, t+1) \tag{1.3}$$

where $p_{j,i}(t, t+1)$ is a *transition probability* from state j to state i at time t and $t+1$ respectively. Sometimes we refer to these transitions as *one-step transitions* since they are only dependent on the previous state. As an extension of the Markov property we may also define a *k-step transition* as a probability of observing a state i at time $t+k$ given the state j at time t as:

$$P(X_{t+k} = i | X_t = j) = p_{j,i}(t, t+k) \tag{1.4}$$

One important distinction is that if the transition probabilities $p_{j,i}(t, t+k)$ do not depend on time t then the Markov Chain is called *homogeneous* otherwise these probabilities vary over time, therefore *heterogeneous* Markov Chain. Considering only first order homogeneous Markov Chain we may define a *transition matrix* $A = (p_{i,j} : i, j \in I)$ as a matrix of transition probabilities between each state $i, j \in I$ such that:

$$p_{j,i} \geq 0 \quad i, j \in I; \quad \sum_{j \in I} p_{i,j} = 1, \quad \forall i \in I \tag{1.5}$$

Rectangular matrix P that satisfies property given by Equation 1.5 is called *stochastic matrix*. Furthermore, we ought to define a probability distribution $\mathbf{p} = \{p_i, i \in I\}$ as a vector of probabilities of observing a each state at time $t = 0$ such that:

$$p_i = P(X_0 = i), \quad i \in I \tag{1.6}$$

and

$$p_i \geq 0 \quad i \in I; \quad \sum_{i \in I} p_i = 1 \tag{1.7}$$

which is also called *initial distribution* of Markov Chain.

Once we have transition matrix A and initial distribution $\mathbf{p}$ that satisfy constraints given by Equation (1.5) and (1.7) respectively, then $\{X_t, t \in \mathbb{N}_0\}$ is a discrete-time homogeneous Markov Chain with transition matrix A and initial distribution $\mathbf{p}$ if and only if all finite dimensional distributions of $\{X_t, t \in \mathbb{N}_0\}$ are consistent with the following equation:

$$P(X_0 = i_0, X_1 = i_1, \ldots, X_k = i_k) = p_{i_0} p_{i_0,i_1} \ldots p_{i_{k-1},i_k} \tag{1.8}$$

where $i_0, i_1, \ldots, i_k \in I$ and $k \in \mathbb{N}_0$. If we abstract from the initial distribution p, such equation is also called *Chapman-Kolmogorov equation*. Above stated equation also holds for non-homogeneous Markov Chains with the only difference that the transition probabilities $p_{i,j}$ are time dependent:

$$P(X_0 = i_0, X_1 = i_1, \ldots, X_k = i_k) = p_{i_0}(0) p_{i_0,i_1}(0,1) \ldots p_{i_{k-1},i_k}(k-1,k) \tag{1.9}$$

another substantial property of homogeneous Markov Chains is that their nth order transition probabilities can be expressed as a product of first order transition probabilities:

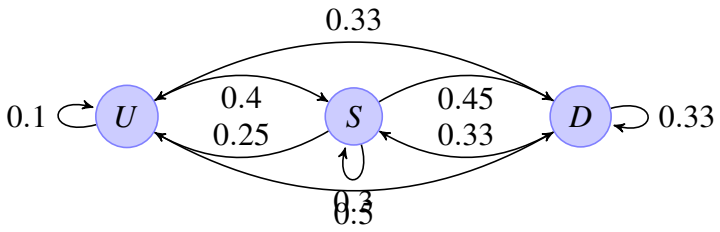
$$P(X_{m+n} = j | X_m = i) = p_{i,j}^{(n)}, \quad i, j \in I \tag{1.10}$$

where generally $p_{i,j}^{(m+n)} = \sum_{k \in I} p_{i,k}^{(m)} p_{k,j}^{(n)}$ is referred to as *Chapman-Kolmogorov equality* and holds for $m, n \in \mathbb{N}_0$ and $P(X_m = i) \geq 0$.

To simply illustrate the idea behind discrete-time homogeneous Markov Chains let us assume a situation where the future market movements transition between a countable number of states $I = \{\text{upward, side, downward}\}$ and there is a transition matrix A and initial distribution p:

$$A = \begin{pmatrix} 0.1 & 0.4 & 0.5 \\ 0.25 & 0.3 & 0.45 \\ 0.33 & 0.33 & 0.33 \end{pmatrix}, \quad p = \begin{pmatrix} 0.2 & 0.3 & 0.5 \end{pmatrix} \tag{1.11}$$

Each row represents full set of transition probabilities between states, also visible from $\sum_{j \in I} p_{i,j} = 1$, i.e. each row of matrix A represents a conditional probability distribution given $i \in I$. Such a relationship can be represented as a diagram indexing each state by U, S and D respectively as follows:



This may be easily interpreted for each given state. For example if we assume that the market moved upwards on the last trading day there is a 0.1 chance that the market will move in positive direction today, in other words the conditional probability of observing the state U today given the state U yesterday is 0.1. On the hand if we suppose that today the market actually transitioned to the state S with probability 0.4 there is now a probability of 0.45 to transition to state D since the future transition is only conditioned by its previous state.

Up until now we have assumed random variable X that modelled the probability of observing a state $i \in I$ given the initial distribution λ. We ought to consider thus we have to allow for a system of discrete random variables that are also identically distributed for each time step $\{X_t, t \in T\}$, for $\{t_0, t_1, \ldots, t_n\} \subset T$ where $n \in \mathbb{N}_0$.

Suppose now that we have observed a given sequence of states for the last week as $\{U, S, D, D, U\}$, and we would like to know the sequence joint probability given the transition matrix A and initial distribution p:

$$\begin{aligned} P(X_{t_0} = x_0, \ldots, X_{t_n} = x_n | A, p) &= P(X_0 = x_0) \prod_{k=1}^{4} P(X_k = x_k | X_{k-1} = x_{k-1}) \\ &= p_{x_0} p_{1,2}\ p_{2,3}\ p_{3,3} p_{3,1} \\ &= 0.2 * 0.4 * 0.45 *\ 0.33 * 0.33 \\ &= 0.002376 \end{aligned}$$

Where the probability of observing state A is determined by our distribution λ evaluated at respective state A since we have no prior knowledge about what exactly happened before t_0.

On the other hand, we might consider a situation in which we have observed such a sequence of events, and we need to determine the next state given the sequence. As in the last example, we have our transition matrix P, distribution λ and a sequence of events observed until now $\{A, B, C, C, A\}$ for $t_{k-4}, \ldots, t_k$. Let us also assume that t_{k+1} is a time of next event for which we are trying to determine its probability.

$$P(X_{t_{k+1}}|X_{t_k},\ldots,X_{t_{k-4}}) = P(X_{t_{k+1}}|X_{t_k}) \tag{1.12}$$

We know that last observed state was U which directs us straight to the first row of our transition matrix A since from the properties of Markov Processes we know that the next state will depend solely on the present state, so we can abstract from the given sequence of past states and focus only on X_{t_k}. Finally, we may conclude that the most likely future state at time t_{k+1} is D with the probability of 0.5. Formally we may write:

$$i_{k+1} = \arg\max_{i\in I} P(X_{t_{k+1}} = i_{k+1}|X_{t_k} = i_k) \tag{1.13}$$

1.1.1 Classification of states

Markov Chains may be classified into several categories based on their properties. Firstly, we may distinguish between *transient* and *recurrent* states and as a convenient notation we will introduce so called *return time* τ_j as a random variable that denotes the time of kth return to state $j \in I$:

$$\tau_j(k+1) = \inf\{n \geq \tau_j(k) : X_n = j\}, \quad k \in \mathbb{N}_0 \tag{1.14}$$

if $\tau_j(k) \leq \infty$ and we assume that $\inf\{\emptyset\} = \infty$ and $\tau_j(0) = 0$.

This random variable also satisfies properties of *recurrence time*. Any random variable $\tau : \Omega \to \mathbb{N}_0 \cup \{\infty\}$ for which outcomes $[\tau = n]$ belong to σ-algebra $\mathscr{F}_n = \sigma(X_0,\ldots,X_n)$ generated by random variables $X_0, X_1, \ldots, X_n$ is called a *recurrence time*.

A state $j \in I$ is called *transient* if there is a non-zero probability that the process will never return to state j once it has left it, i.e.:

$$P(\tau_j(1) = \infty|X_0 = j) > 0, \quad \sum_{n=0}^{\infty} p_{j,j}^{(n)} \leq \infty \tag{1.15}$$

for some $k \in \mathbb{N}_0$. On the other hand, a state $j \in I$ is called *recurrent* if it is not transient, i.e.:

$$P(\tau_j(1) \leq \infty|X_0 = j) = 1, \quad \sum_{n=0}^{\infty} p_{j,j}^{(n)} = \infty \tag{1.16}$$

We can further distinguish between *positive recurrent* if the expected return time is finite $E[\tau_j(1)|X_0 = j] < \infty$ and *null recurrent* if the expected return time is infinite $E[\tau_j(1)|X_0 = j] = \infty$.

Let us make one more distinction regarding a Markov chain states. State is said to be periodic if it can only be revisited (i.e., return to the same state) in multiples of some integer larger than 1. The greatest common divisor of these numbers is the period. If the period is larger than 1, the state is called periodic and on the contrary, if there is no such integer and the state can be revisited at any time, then the state is called aperiodic. If all states in a Markov chain are aperiodic, the Markov chain itself is said to be aperiodic as well. Aperiodicity is a desirable property for a Markov chain because it ensures that the chain does not get trapped in oscillating sequences of states. To clarify, a periodicity does not mean that each state must be reachable from every other state in just one step. It's a more subtle concept, meaning that the greatest common divisor of the lengths of all cycles in the chain must be 1, so that any state can be reached from any other state in a variable number of steps. Previous properties of states lead us to define ergodic state for which it holds that it needs to be positive recurrent and aperiodic. Such trait is desirable since it implies that the state will be visited infinitely often and the expected time between visits is finite. That also implies that if all states of a Markov chain are ergodic, then the chain itself is ergodic and irreducible, i.e. all states communicate with each other.

1.1.2 Stationary distribution

A pivotal concept linked to Markov Chains is that of the stationary distribution, a distinct probability distribution that remains invariant under the transition dynamics of the chain. If we denote $\pi = \{\pi_j, j \in I\}$ as a probability distribution, and it satisfies following equality:

$$\pi_j = \sum_{i \in I} \pi_i p_{i,j}, \quad j \in I \tag{1.17}$$

then π is called a stationary distribution of Markov Chain. That also implies that if the initial distribution of homogeneous Markov Chain is stationary in the sense of Eq. (1.14) then Markov Chain is called strictly stationary stochastic process since the joint distribution of any finite number of random variables is invariant under the transition dynamics of the chain. More specifically, for any $n, k \in \mathbb{N}_0$ and $i_0, i_1, \ldots, i_n \in I$ it holds that:

$$P(X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n) = P(X_k = i_0, X_{k+1} = i_1, \ldots, X_{k+n} = i_n) \tag{1.18}$$

and also for π_j called initial stationary probabilities:

$$p_j(n) = P(X_n = j) = \pi_j, \quad j \in I \tag{1.19}$$

It's important to stress that the existence and uniqueness of a stationary distribution is not guaranteed for all Markov Chains, but under specific conditions a unique stationary distribution does exist and any initial distribution converges to this stationary distribution as time progresses. If all states of the chain are transient or null recurrent, then no stationary distribution exists. On the other hand if the chain is positive recurrent, then a stationary distribution exists and is unique.

The fundamental significance of the stationary distribution arises from its ability to dictate the long-term, steady-state behavior of the chain. Furthermore, the stationary distribution plays an essential role in the calculation of expected return times, the analysis of limiting probabilities, and forms the backbone of algorithms such as the Metropolis-Hastings algorithm widely used in Monte Carlo simulations.

1.1.3 Cryptocurrency market movements I.

The probabilities in previously introduced transition matrix A were imaginary and served only as a mere example of the main properties of homogeneous discrete-time Markov Chains. For now consider a dataset of BTC-USDT daily close prices from public cryptocurrency exchange Binance website from 23rd August 2020 to 15th May 2023. Firstly, we ought to make several assumptions about the data in order to satisfy properties of Markov Chains, namely define finite state space, transition period and memoryless process.

1) **Transition period**: We will define a transition period as a day, i.e. the transition probabilities will be calculated for each day.

2) **State space**: We will define a state space as a set of 3 states {upward, side, downward} which will be determined by the percentage change of the close price of the current day with respect to the previous day as follows:

    a) **Upward**: If the percentage change of the close price of the current day with respect to the previous day is greater than 0.5%.

    b) **Side**: If the percentage change of the close price of the current day with respect to the previous day is between -0.5% and 0.5%.

    c) **Downward**: If the percentage change of the close price of the current day with respect to the previous day is less than -0.5%.

3) **Memoryless process**: We will assume that the future state of the market only depends on the current state, i.e. the transition probabilities are independent of time.

Given these assumptions we first examine BTC-USDT close price time series data in 1.1 and the distribution of the percentage change of the close price of the current day with respect to the previous day as shown in Figure 1.2.

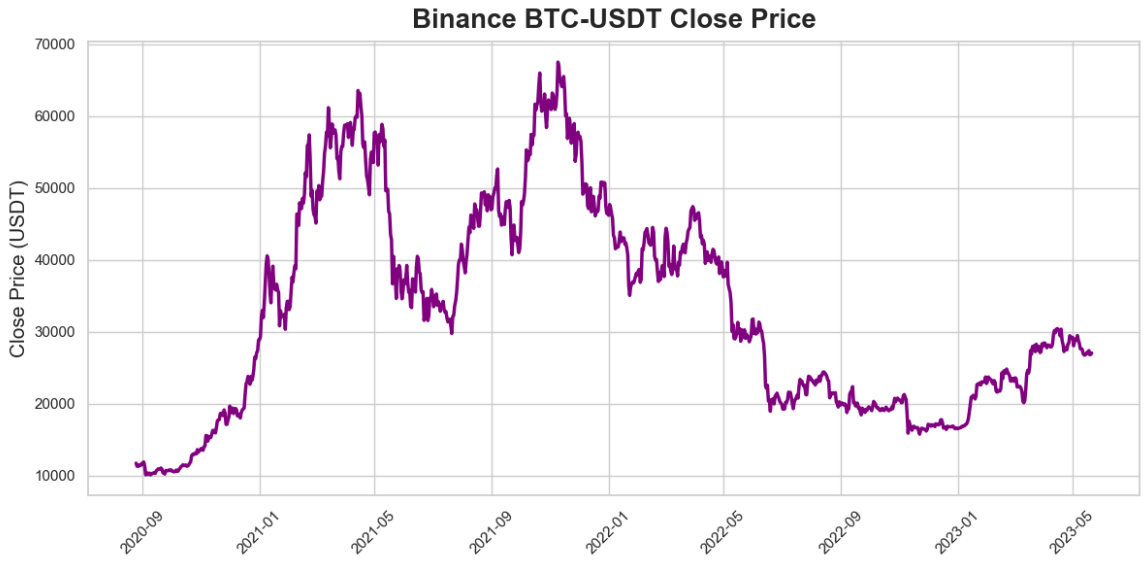


Fig. 1.1 BTC-USD daily close prices from 23rd August 2020 to 15th May 2023 obtained from Binance. [1]

Observing the distribution of the log-returns of the daily BTC-USDT close price in 1.2 we may conclude that such a random variable is normally distributed, given the symmetric property of the distribution with respect to the mean value. Furthermore, it is visible that the kurtosis might be greater than 3, which implies that the distribution has heavier tails than the normal distribution, i.e. the extreme events are more likely to occur than in the normal distribution. Such a property is also called *leptokurtic* distribution which is a result of the high volatility of the cryptocurrency market.(source?)

Let us now take the ordered sample of states from the BTC-USDT close price time series data and calculate the transition and initial probabilities for each state as follows:

$$A = \begin{pmatrix} 0.22 & 0.16 & 0.62 \\ 0.32 & 0.27 & 0.41 \\ 0.6 & 0.19 & 0.21 \end{pmatrix}, \quad p = \begin{pmatrix} 0.4 & 0.19 & 0.41 \end{pmatrix} \tag{1.20}$$

where we note that the frequency of observing each state is proportional to the initial distribution p, and we assume that state space is $I = \{U, S, D\}$ By the definition of the transition matrix A, such matrix is also a stochastic matrix since it satisfies the properties

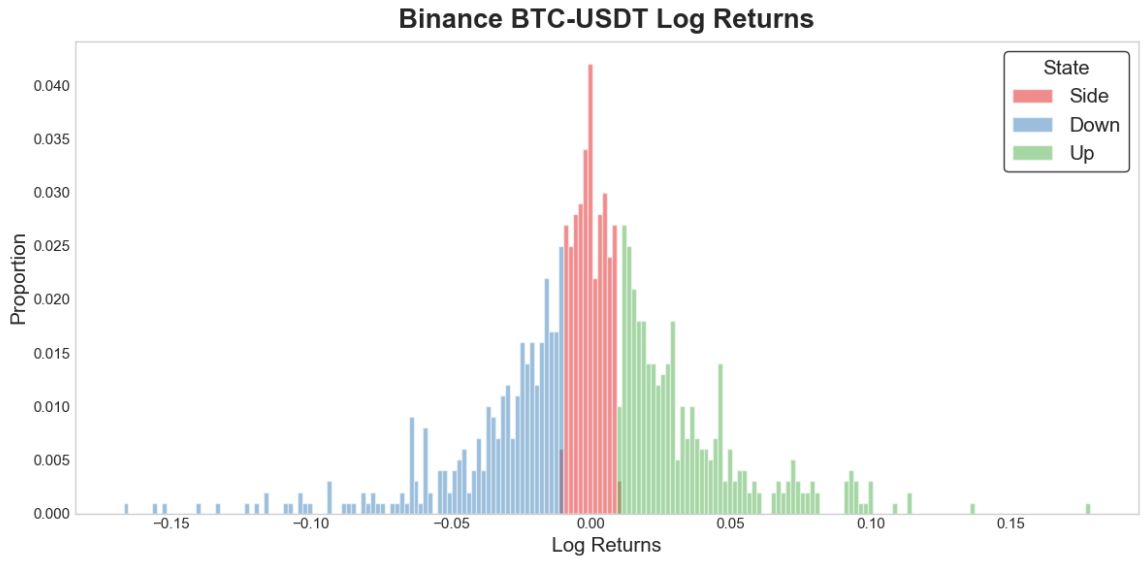


Fig. 1.2 Distribution of the log returns with respect to predefined market states. [1]

given by Equation (1.5). Each state of the transition matrix A is also non-absorbing since the probability of observing a state i at time $t+1$ given the state j at time t is greater than 0 and aperiodic since the greatest common divisor of the lengths of all cycles in the chain is 1. Positive recurrence of each state is satisfied as well. Furthermore, we may also conclude that the state space is irreducible since all states communicate with each other, i.e. there is a non-zero probability of transitioning from any state to any other state. Therefore, we may conclude that the Markov Chain is ergodic, and the stationary distribution exists, is unique and is approximated by the initial distribution p using Equation 1.19. Finally, we may also calculate the expected return time for each state as follows:

$$E[\tau_U(1)|X_0 = U] = \frac{1}{\pi_U} = \frac{1}{0.4} = 2.5 \tag{1.21}$$

$$E[\tau_S(1)|X_0 = S] = \frac{1}{\pi_S} = \frac{1}{0.19} = 5.26 \tag{1.22}$$

$$E[\tau_D(1)|X_0 = D] = \frac{1}{\pi_D} = \frac{1}{0.41} = 2.5 \tag{1.23}$$

where π is the stationary distribution of the Markov Chain. In other words, the expected time of returning to state U and D is 2.5 days, and 5.26 days for state S. Although, the expected return times provide interesting behavioral insights, they are simplified by the Markov property of memoryless process, stock and cryptocurrency markets do have certain

memory and path-dependence properties as well as they are effected by external factors such as news, social media, etc. Therefore, the expected return times are only approximations of the real expected return times.

1.2 Continous-time Markov Chains

In the previous section we have considered a discrete-time Markov Chains, i.e. the state space and transition period was discrete. Such period means that the chain can stay in a state for integer number of time steps before transitioning to another state. For continous-time Markov Chains we will assume that the transition period is continuous, more specifically, the period is exponentially distributed with parameter λ.

Let us consider a stochastic process $\{X_t, t \geq 0\}$ with a state space I and a distribution λ such that for all $t \in \mathbb{R}_0^+$ and $i_0, i_1, \ldots, i_{t+1} \in I$ it holds that:

$$P(X_t = j | X_s = i, X_{t_n} = i_n, \ldots, X_{t_1} = i_1) = P(X_t = j | X_s = i) \tag{1.24}$$

where $0 \leq t_1 < \ldots < t_n < s < t$ and so it is trivially seen that such expression is equivalent to the discrete-time Markov Chain property given by 1.2 with the only difference of continuous transition period.

Since the state space remains the same as in discrete-time Markov Chains, we refer to the same transition matrix A and initial distribution $\mathbf{p}$. In upcoming subsection, continuous-time Markov Chain is assumed to be homogeneous, i.e. the transition probabilities are independent of time:

$$p_{i,j}(s, s+t) = p_{i,j}(t), \quad i, j \in I \tag{1.25}$$

which also implies that the transition probability is determined only by the length of the transition period t. Chapmam Kolmogorov equality for $s, t \geq 0$ also holds for continous-time Markov Chains:

$$p_{i,j}(s+t) = \sum_{k \in I} p_{i,k}(s) p_{k,j}(t), \quad i, j \in I \tag{1.26}$$

Here we also assume that $\lim_{t \to 0_+} p_{i,j}(t) = \delta_{i,j}$ where $\delta_{i,j}$ is a Kronecker delta function, i.e. the transition probabilities $p_{ij}(t)$ are right continuous at $t = 0$. If such condition is satisfied for any homogeneous continous-time Markov Chain, then the underlying stochastic process is said to be continuous and there exists its version that is separable, measurable

and its trajectories are step functions almost surely. Such version allows us to infer certain properties of the stochastic process.

For example, important property for such Markov Chain given Doob convergence theorem for $s \geq 0$ and $h > 0$ is:

$$P(X_t = i, s \leq t \leq s+h | X_s = i) = \exp(-q_i h) \tag{1.27}$$

which means that the probability of staying in state i for time h is equal to $e^{-q_i h}$. Non-negative real number $q_{i,j}$ is called *transition rate* from state i to state j and absolute transition rate $q_i = \sum_{j \neq i} q_{i,j}$ respectively. Trivially, in cases where the transition rate $q_i = 0$, the $p_{i,i} = 1$, i.e. the state i is absorbing, once the chain enters such state it remains in such state for infinite amount of time. On the contrary, if $0 < q_i < \infty$ then the state i is non-absorbing and stable, therefore the chain will eventually leave such a state. For infinite transition rate $q_i = \infty$ the state i is called *unstable* where the time of staying in such state is almost surely zero.

If we consider a stable state i then its expected time of staying in such state is exponentially distributed with expected value of $1/q_i$. In other words, the expected time of staying in state i is equal to the inverse of the transition rate q_i.

Since, we have already defined that the process has exponentially distributed transition period, we may define a *holding time* T_i as a random variable that denotes the time of staying in state i:

$$T_i = \inf\{t \geq 0 : X_t \neq i | X_0 = i\} \tag{1.28}$$

from which it follows that $P(T_i > s) = P(X_t = i, 0 \leq t \leq s | X_0 = i) = e^{-q_i s}$ and its probability density functions is:

$$f(x) = \begin{cases} q_i e^{-q_i x}, & x \geq 0 \\ 0, & \text{elsewhere} \end{cases} \tag{1.29}$$

According to the properties of the transition rates, such time-homogeneous continuous Markov Chain should satisfy following equations:

$$\begin{aligned} P(X_{t+h} = i | X_t = i) &= 1 - q_i h + o(h) \\ P(X_{t+h} = j | X_t = i) &= q_{i,j} h + o(h), \quad i \neq j \end{aligned} \tag{1.30}$$

where $o(h)$ is a function of h such that $\lim_{h \to 0} \frac{o(h)}{h} = 0$. Let us denote $\mathbf{Q}$ as transition rates matrix with entries $\{q_{i,j} : i, j \in I\}$.

Non-negative real number $q_{i,j}$ is called *transition rate* from state i to state j and absolute transition

Such intensities resemble the probability functions of Poisson process, and indeed homogeneous continous-time Markov Chain is a special case of Poisson process with intensity $\lambda \geq 0$ if following conditions are satisfied:

1) Stochastic process is viewed as a jump process, i.e. current state i either stays in state i or jumps to another state $j = i+1$. Therefore, given an interval $[t, t+h]$ the probability of jumping to another state is $\lambda h + o(h)$ and the probability of staying in state i is $1 - \lambda h + o(h)$.

2) Intensity λ is constant, i.e. the probability of jumping to another state is independent of time, i.e. depends only on the length of the interval. We refer to such process as homogeneous Poisson process.

3) Number of jumps in disjoint intervals are independent, i.e. the probability of jumping to another state in disjoint intervals $[t_1, t_1+h]$ and $[t_2, t_2+h]$ is equal to the probability of jumping to another state in interval $[t_1, t_1+h] \cup [t_2, t_2+h]$.

4) The probability of more than one jump in a sufficiently small interval is negligible, i.e. the probability of jumping to another state in interval $[t, t+h]$ is $o(h)$.

5) Process starts in state $i = 0$ at time $t = 0$.

In a case of constant return rates, the matrix $\mathbf{Q}$ with entries $q_i = -\lambda$ and $q_{i,j} = \lambda$ as follows:

$$\mathbf{Q} = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \dots \\ 0 & -\lambda & \lambda & 0 & \dots \\ 0 & 0 & -\lambda & \lambda & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{1.31}$$

1.2.1 Cryptocurrency market movements II.

1.3 Hidden Markov Model

Up until now we have considered visible states in a sense that the sequence of states was known, we refer to these models as *visible Markov Models*. In this section we will consider a situation in which we do not observe the states directly but only as a guess given other

visible observations that are available to us. These "visible" observations are considered as an emissions from the hidden state sequence. Thus, the observations are assumed to be generated by the hidden states. We may interpret this relation by the following diagram where we have a hidden state sequence $Z = \{z_1, z_2, \ldots, z_T\}$:

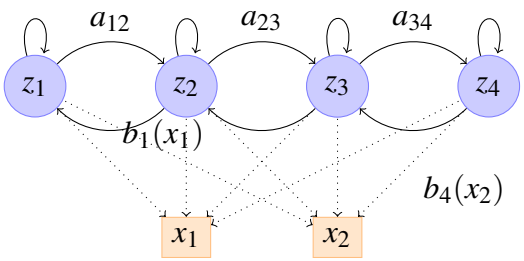


Fig. 1.3 An HMM with 4 hidden states which emits 2 discrete observable states denoted by x_1 or x_2. a_{ij} is the probability to transition from state z_i to state z_j. $b_j(x_k)$ is the probability to emit symbol x_k in state z_j.

There are 3 main assumptions of Hidden Markov Models as a consequence of the properties of Markov processes. For $Z = \{z_t\}_{t=0}^{T}$ being the hidden state sequence:

1) The Markov assumption - this assumption states that the next hidden state z_{t+1} depends only on the current state z_t, so that the transition probabilities are defined as:

$$P(z_{t+1} = j | z_t = i, z_{t-1} = l, \ldots, z_0 = n) = P(z_{t+1} = j | z_t = i) = p_{ij} \tag{1.32}$$

It it also possible to assume that the states in HMM are dependent beyond the current state therefore giving rise to k-order HMMs as opposed to classical first-order HMM, moreover, such variations are uneasy to analyse.

2) The stationary assumption - The transition matrix is invariant of the time, thus for any arbitrarily set time t_1 and t_2:

$$P(z_{t_1+1} = j | z_{t_1}) = P(z_{t_2+1} = j | z_{t_2} = i) = p_{ij} \tag{1.33}$$

3) The observation independence assumption - The current observation or output is statistically independent of the previous observations. So if we have a observation sequence $X = \{x_1, x_2, \ldots, x_T\}$ then:

$$P(X | z_1, z_2, \ldots, z_T, \lambda) = \prod_{t=1}^{T} P(x_t | z_t, \lambda) \tag{1.34}$$

Given these assumptions we define the joint probability of the hidden states and the observations $P(X,Z)$ as follows:

$$P(X,Z) = \prod_{t=1}^{T} P(z_t|z_{t-1})P(x_t|z_t) \tag{1.35}$$

There are mainly 3 fundamental problems in HMM that need to be resolved as in (Oliver C. Ibe):

1. Evaluation problem - Given a model denoted as $\lambda = (P,\theta,\pi)$ and an observation sequence $X = x_1,x_2,\ldots,x_T$, how to efficiently compute the probability that the model generated the observation sequence, in other words, what is $P(X|\lambda)$?

2. Decoding problem - Given a model $\lambda = (P,\theta,\pi)$, what is the most likely sequence of hidden states that could have generated a given observation sequence? Thus we would like to find $Z = \arg\max_{Z} P(Z,X|\lambda)$, where Z is the hidden state sequence.

3. The learning problem - Given a set of observation sequences find the HMM that best explains the observation sequence. Thus, find the values of λ that maximise $P(X|\lambda)$ or in order to estimate the most likely parameters of HMM for a given observation sequence.

The most traditional approaches in solving these 3 fundamental problems differ and one may not suffice in solving all three. The evaluation problem is usually by forward-backward algorithm, the decoding problem by well-known Viterbi algorithm and the last learning problem by Baum-Welsch algorithm which is a special case of Expectation-maximization (EM) algorithm.

1.3.1 Forward and Backward algorithm

While given a sequence of observations, in our case observable states, denoted by $X = x_1,x_2,\ldots,x_T$ and a model $\lambda = (P,\theta,\pi)$ we wold like to compute the conditional probability of observing the sequence X under the model constraints. Thus:

$$P(X|\lambda) = \sum_{Z} P(X|Z,\lambda) * P(Z|\lambda) \tag{1.36}$$

where $Z = z_1,z_2,\ldots,z_T$ is a fixed sequence of hidden states. Our goal may be divided into two parts while the first part $P(X|Z,\lambda)$ computes the conditional probability of the observation sequence X given the sequence Z and model λ and the second part $P(Z|\lambda)$ accounts only for the transition probabilities among the hidden states. To summarise:

$$P(X|Z,\lambda) = \prod_{t=1}^{T} P(x_t|z_t,\lambda) = \theta_{z_1}(x_1)\prod_{t=2}^{T}\theta_{z_t}(x_t) \tag{1.37}$$
$$P(Z|\lambda) = \pi_{z_1} * \prod_{t=2}^{T} p_{z_t z_{t+1}}$$

where $\theta_{z_t}(x_t)$ is the emission probability from observation x_t into z_t in our model and $p_{z_t z_{t+1}}$ are the transition probabilities for the given sequence Z. After substitution of terms from Eq. 1.8 into Eq 1.7 we obtain:

$$\begin{aligned} P(X|\lambda) &= \sum_{Z} P(X|Z,\lambda) * P(Z|\lambda) \\ &= \sum_{z_1,\ldots,z_T} \theta_{z_1}(x_1)\pi_{z_1}\prod_{t=2}^{T} p_{z_t z_{t+1}}\theta_{z_t}(x_t) \end{aligned} \tag{1.38}$$

The summation above refers to all possible permutations of the sequence of hidden states Z which implies that we would have N^T possible sequences if we assume that N indicates all possible hidden states at each step. Furthermore, in order to calculate $P(X|\lambda)$ we have $2TN^T$ calculations which is exponential in T and not feasible for real application. As opposed to brute-force procedure as described above we take use of a forward and backward algorithm.

1.3.1.1 Forward algorithm

Previous introduction required huge amounts of calculations that were essentially unnecessary, redundant. Each sequence can be decomposed into multiple subsequences which are shared among different sequences and do not need to be recomputed again. These subsequences can be represented by a trellis as shown in Fig 1.2. With a help of such diagram we may imagine recording the probability of each distinct subsequence at each time step. In other words, we wish to compute the joint probability $P(z_t, x_1, x_2, \ldots, x_t)$ while taking advantage of the conditional independence of x_t which only depends on z_t, moreover, due to Markov assumption, z_t depends only on z_{t-1}. Let us now define the forward probability variable $\alpha_t(i)$ as follows:

$$\alpha_t(i) = P(x_1, x_2, \ldots, x_{t-1}, x_t, z_t = i|\lambda) \tag{1.39}$$

Which we may also define as the probability of being in state i at time t after having observed the sequence $x_1, x_2, \ldots, x_t$. The calculation therefore results in summing the incoming arcs at trellis node which is derived from:

$$\begin{aligned}\alpha_t(i) &= P(x_1, x_2, \ldots, x_{t-1}, x_t, z_t = i|\lambda) \\ &= P(x_t|z_t = I, \lambda)P(x_1, x_2, \ldots, x_{t-1}, z_t = i|\lambda) \\ &= P(x_t|z_t = i, \lambda)\sum_{j\in I} P(z_t = I|z_{t-1} = j, \lambda)P(x_1, x_2, \ldots, x_{t-1}, z_{t-1} = j|\lambda) \\ &= \theta_i(x_t)\sum_{j=1}^{N} p_{ji}\alpha_{t-1}(j)\end{aligned} \tag{1.40}$$

Where I is a set of all possible hidden states, i, j denote elements of I and N the size of set $|I|$. We would repeat the iterative procedure until the termination at time T where $P(X|\lambda) = \sum_{i=1}^{N} \alpha_T(i) = \sum_{i=1}^{N} P(X, z_T = I|\lambda)$. Moreover, we initialise the trellis with $\alpha_1(i) = \pi_i\theta_i(x_1)$. Resulting algorithm requires time complexity of $O(TN^2)$ which is a significant improvement from the brute force approach with complexity of $O(2TN^T)$.

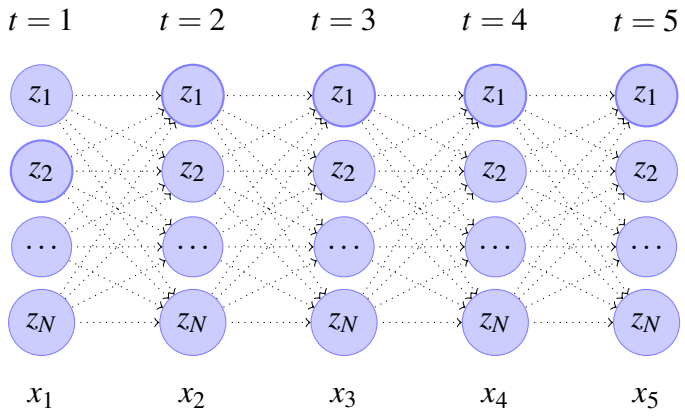


Fig. 1.4 Trellis of the observation sequence $x_1,\ldots,x_5$ for the above HMM. As an example, the transition between state z_1 at time t=2 and state z_4 at time t=3 has probability $\alpha_2(z_1)p_{14}\theta_{z_4}(x_3)$, where $\alpha_t(i)$ is the probability to be instate i at time t.

1.3.1.2 Backward algorithm

While computing the backward probability variable denoted as $\beta_t(i)$ we assume the reversed iterative procedure. Let us define the backward probability variable as:

$$\begin{aligned}\beta_t(i) &= P(x_{t+1}, x_{t+2}, \ldots, x_{t-1}, x_t | z_t = i, \lambda) \\ &= \sum_{i,j \in I} P(x_{t+1}, x_{t+2}, \ldots, x_{t-1}, x_t, z_{t+1} = j | z_t = i, \lambda) \\ &= \sum_{i,j \in I} P(x_{t+1} | z_{t+1} = j) P(x_{t+2}, \ldots, x_{t-1}, x_t | z_{t+1} = j) P(z_{t+1} = j | z_t = i, \lambda) \\ &= \sum_{j=1}^{N} \theta_j(x_{t+1}) p_{ij} \beta_{t+1}(j)\end{aligned} \tag{1.41}$$

As we could have in forward algorithm we define effectively 3 steps:

1. Initialization step: with default value of $\beta_T(i) = 1$ for $i \in 1, 2, \ldots, N$.

2. Induction step:

$$\beta_t(i) = \theta_j(x_{t+1}) p_{ij} \beta_{t+1}(j) \tag{1.42}$$

   Afterwards, we update the time t = t-1 and check whether expression above holds which is as long as $t > 0$.

3. Termination step: once the iterative procedure is exhausted, i.e. when t=0 we have the estimate of $P(X|\lambda)$ as:

$$P(X|\lambda) = \sum_{j=1}^{N} \theta_j(x_1) \pi_i \beta_1(i) = \sum_{j=1}^{N} \alpha_1(i) \beta_1(i) \tag{1.43}$$

Getting everything together we may prove that the forward and backward algorithms coincide in solving the posterior joint and marginal distributions of all hidden states. Consider the case of posterior joint distribution:

$$
\begin{aligned}
P(X|\lambda) &= \sum_{i=1}^{N} P(X, z_t = i|\lambda) \\
&= \sum_{i=1}^{N} P(x_1, x_2, \ldots, x_t, x_{t+1}, \ldots, x_T | z_t = i, \lambda) P(z_t = i|\lambda) \\
&= \sum_{i=1}^{N} P(x_1, x_2, \ldots, x_t | z_t = i, \lambda) P(x_{t+1}, \ldots, x_T | z_t = i, \lambda) P(z_t = i|\lambda) \\
&= \sum_{i=1}^{N} P(x_1, x_2, \ldots, x_t, z_t = i|\lambda) P(x_{t+1}, \ldots, x_T | z_t = i, \lambda) \\
&= \sum_{i=1}^{N} \alpha_t(i)\beta_t(i)
\end{aligned}
\tag{1.44}
$$

See that above we take a use of conditional independence of observations given the hidden state and therefore we may transform the problem using already known variables for forward and backward passes over the hidden states. Thus, the posterior marginal distribution is:

$$
P(X, z_t = i|\lambda) = \alpha_t(i)\beta_t(i) \tag{1.45}
$$

1.3.2 Viterbi algorithm

Once we solve the evaluation problem, thus finding the optimal procedure in computing the posterior marginal distributions of all hidden states and $P(X|\lambda)$, we may visualise the result in a trellis as shown in Figure 1.2. The second problem, decoding problem, aims to find the sequence of hidden states Z^* that most likely produced observation sequence X. Finding the most likely sequence Z^* maximazes $P(Z|X,\lambda)$. As in the encoding problem, the complexity of the optimisation problem explodes if we decide to compute all possible sequences of hidden states to N^T calculations. The solution as in the forward and backward algorithm simplifies when we calculate the hidden state with highest probability individually rather than entire sequence up to time t. The idea is that once we find the most likely hidden state given observation and the model at each time step we may discard the rest of the possible hidden states since they obviously could not have most likely produced the observation. The complexity of the optimal sequence of hidden states decreases significantly to NT thus transforming the exponential complexity into linear. As in forward and backward algorithm we define variable $\gamma_t(i)$ as:

$$\gamma_t(i) = P(z_t = i|X,\lambda) = \frac{P(X, z_t = i|\lambda)}{P(X|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)} \propto \alpha_t(i)\beta_t(i) \tag{1.46}$$

Where the expressions in the third equality are obtained as a solution to a before mentioned equations. Also, the $P(X|\lambda)$ serves only as a normalising constant and is irrelevant for the optimisation given the respective time step. Moreover, the conditional probability $\gamma_t(i)$ forms a conditional probability mass function since it is non-negative for all possible values of hidden states and observations and the sum over all possible hidden states at each time step is:

$$\sum_{i=1}^{N}\gamma_t(i) = 1 \tag{1.47}$$

This would be one of the possible ways to estimate the most likely hidden states individually and then combine them sequentially to obtain the whole sequence of hidden states as:

$$z_t^* = \arg\max_{i\in I}\{\gamma_t(i)\} \tag{1.48}$$

Method as described by the Eq 1.17 and 1.19 is also called maximum a posteriori probability estimate (hereinafter 'MAP estimate') which equals to the mode of the posterior distribution. If the posterior distribution is discrete, the mode is simply the value of a random variable with the highest probability. For continuous distributions the mode is a value of a random variable for which the likelihood/log-likelihood function attains its local maximum/maxima. Assuming now Z denotes the continuous random variable of hidden states and observation sequence X up to time t is present, the MAP estimate of the hidden state $\hat{Z}_{MAP}$ at a given point in time is:

$$\begin{aligned}\hat{z}_{t,MAP} &= \arg\max_{z_t} f(Z_t|X) \\ &= \arg\max_{z_t} \frac{f(X|Z_t)g(Z_t)}{\int_I f(X|Z_t)g(Z_t)\,dZ} \\ &= \arg\max_{z_t} f(X|Z_t)g(Z_t)\end{aligned} \tag{1.49}$$

Where g indicates probability density function of Z_t, so called prior distribution, and I is its domain. The second equality follows from the Bayes' Theorem. The result clearly holds for the discrete case:

$$\begin{aligned}\hat{z}_{t,MAP} &= \arg\max_{z_t} P(Z_t = i|X) \\ &= \arg\max_{z_t} \frac{P(X|Z_t = i)P(Z_t = i)}{\sum_{i=1}^{N} P(X|Z_t = i)P(Z_t = i)} \\ &= \arg\max_{z_t} P(X|Z_t = i)P(Z_t = i) \\ &= z_t^*\end{aligned} \tag{1.50}$$

However, the solution as such might not produce the most likely sequence of states given the sequence of observations. This is due to the fact that the individual estimates do not incorporate the transition probability between most likely states at time t-1 and t. Hence, it might be possible that some states are highly unlikely to transition into other that were evaluated as most likely. Fortunately, the mentioned shortcomings of such approach are easily solved by Viterbi algorithm.

In order to find the most likely sequence of hidden states $Z^* = z_1^*, z_2^*, \ldots, z_T^*$ given the observation sequence $X = x_1, x_2, \ldots, x_T$ it is necessary to maximise with respect to the whole sequence rather than individually to avoid less likely transitions as introduced. The algorithm therefore defines a variable as follows:

$$\delta_t(i) = \max_{Z_{t-1}} P(z_1, z_2, \ldots, z_t = i, x_1, x_2, \ldots, x_{t-1}, x_t|\lambda) \tag{1.51}$$

That is the most likely state sequence given the observation sequence up to time t. Moreover, for the purpose of the algorithm we also define variable $\psi_t(i)$ that stores the node of the incoming arc that leads to the most probable state path.

$$\psi_t(j) = \arg\max_{i \in I} \delta_{t-1}(i) p_{ij} \tag{1.52}$$

Then we can easily provide the steps to obtain the most likely state sequence:

1. Initialize the algorithm given the initial distribution of hidden states π:

$$\delta_1(i) = \pi_i b_i(o_1) \tag{1.53}$$
$$\psi_1(i) = 0 \tag{1.54}$$

where we compute the initial values of above variables for each hidden state $i \in I$.

2. Recursive step:

$$\delta_1(i) = \arg\max_{i \in I} \delta_{t-1}(i) p_{ij} b_j(o_t) \tag{1.55}$$
$$\psi_t(j) = \arg\max_{i \in I} \delta_{t-1}(i) p_{ij} \tag{1.56}$$

At each iterative step we update the time so that $t = t + 1$. The second step continues as long as the $t < T$ if that does not hold the third step terminates the algorithm and we backtrack the most likely state sequence. Although, recursive step is very similar to the induction step in the forward algorithm there exist a main difference between the two that lies in the fact that Viterbi algorithm uses maximization instead of summation over previous states.

3. Termination step for the last observation

$$P^* = \max_{i \in I} \delta_T(i) \tag{1.57}$$
$$q_T^* = \arg\max_{i \in I} \delta_T(i) \tag{1.58}$$

4. Backtrack the optimal state sequence:

$$z_t^* = \psi_{t+1}(z_{t+1}^*) \tag{1.59}$$

For $t = T-1, T-2, \ldots, 1$.

The resulting most likely state path can be visually viewed as in the form of a path within the trellis introduced in Fig. 1.2:

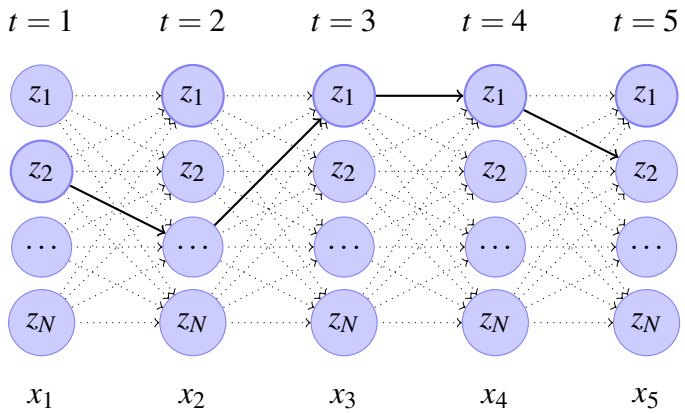


Fig. 1.5 Trellis of the observation sequence $x_1, \ldots, x_5$ for the HMM. Bold lines illustrate the most likely sequence found by the Viterbi algorithm.

1.4 Kalman Filters

Chapter 2

Parameter Estimation for Hidden Markov Models

2.1 Expectation–Maximization algorithm

Also abbreviated as EM algorithm is an iterative approach for computing the maximum likelihood estimates. It is used in situations where incomplete data are present therefore a part of a complete data set is hidden, and we may not be able to apply straightforward analytical procedures for computing maximum likelihood estimates as in a case of complete data.

In other words we want to find the best estimate of the parameters for which the observed sequence is the most likely. This is of great importance and efficiency for Hidden Markov models where we have a sample space of observed variables, let us denote it as X and a hidden discrete sample space Z. We assume that there is a mapping from X into Z, where $\forall x \in X$ is a realization from sample space X and the mapping from X into Z in many-one, as seen in Figure 1. We also assume that given our model the realizations $z \in Z$ are not observable directly but only as a projection from X into Z.

Let us also denote $\theta \in \Theta$ which is a vector of parameters of the given distribution belonging to sampling distribution Θ. The aim of EM algorithm is essentially to find the best estimate of θ that maximizes the likelihood function $L(\theta) = p(X|\theta)$,this is known as Maximum likelihood (ML) estimate of θ.

To visualize and apply the non-iterative approach, let us consider now the daily log returns of BTC/USDT trading pair in past 5 years. If we plot the histogram we may see that the log-returns may asymptotically follow normal distribution. Since the probability density function in such a case is unimodal and has only one global maximum, the logarithmic

transformation converts multiplicative structure into additive with the preservation of global maximum to be optimized by taking the partial derivative w.r.t. each parameter. First, we formulate the likelihood function of two-parametric normal distribution.

$$L(\mu,\sigma^2|x_1,\ldots,x_N) = P(\mu,\sigma^2|x_1,\ldots,x_N) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(x_i-\mu)^2}{\sigma^2}} \tag{2.1}$$

$$l(\mu,\sigma^2|x_1,\ldots,x_N) = \ln L(\mu,\sigma^2|x_1,\ldots,x_N) \tag{2.2}$$

where $x_1,\ldots,x_N$ is a vector of log-returns of length N and μ and σ^2 are parameters to be estimated.

$$l(\mu,\sigma^2|x_1,\ldots,x_N) = -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2 \tag{2.3}$$

If we now take the partial derivative w.r.t. the parameter μ and σ^2 and set it to zero we obtain the ML estimate of the parameter as follows:

$$\frac{\partial}{\partial\mu} l(\mu,\sigma^2|x_1,\ldots,x_N) = \frac{1}{\sigma^2}(\sum_{i=1}^{N} x_i - N\mu) \tag{2.4}$$

$$0 = \frac{1}{\sigma^2}(\sum_{i=1}^{N} x_i - N\mu) \tag{2.5}$$

$$\mu_{MLE} = \frac{\sum_{i=1}^{N} x_i}{N} \tag{2.6}$$

$$\frac{\partial}{\partial\sigma^2} l(\mu,\sigma^2|x_1,\ldots,x_N) = -\frac{N}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{N}(x_i-\mu)^2 \tag{2.7}$$

$$0 = -\frac{N}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{N}(x_i-\mu)^2 \tag{2.8}$$

$$\sigma^2 = \frac{\sum_{i=1}^{N}(x_i-\mu)^2}{N} \tag{2.9}$$

Given the daily returns of BTC/USDT since 2017, ML estimate of the mean denoted by μ is 0.003 and σ^2 is 0.042. Moreover we may compare the ML fit with non-parametric estimate obtained by kernel density estimate that uses a sum of gaussian kernel functions for each sampled data point to estimate the smooth density given only data and prior bandwidth parameter h set to 0.2.

Kernel density estimate is calculated as follows:

$$\rho_K(y) = \sum_{i=1}^{N} K(y - x_i; h) \tag{2.10}$$

$$K(x;h) \propto e^{\left(-\frac{x^2}{2h^2}\right)} \tag{2.11}$$

where $K(x;h)$ denotes the Gaussian kernel function with bandwidth parameter h and $\rho_K(y)$ is a kernel density estimate at point y.

Fig. 2.1 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with subplot containing two lines for MACD and Signal line and a bar chart as their difference*

Although the assumption of complete data simplifies the analytical procedure of calculating the ML estimate in closed form, it may only be used by taken in consideration observable data. The task of EM algorithm can be used to iteratively compute the most likely parameter given the data, i.e. the parameter that most likely produced such data, but since the closed form solution is available at this point, it would be redundant to use iterative procedure since both would provide the same result. Nevertheless, the iterative EM algorithm given the fully observed complete data behaves in following way:

In order to estimate the single or vector of parameters θ we use log-likelihood function:

$$l(\theta|X) = \ln p(X|\theta) \tag{2.12}$$

Since the natural logarithm is strictly monotonic increasing function the value of θ maximizes the log-likelihood as well as the likelihood function. Afterwards, in a simplistic sense, the EM algorithm iterates over possible values of θ to find the best estimate, i.e. until convergence criterion is satisfied:

$$|l(\theta^i|X) - l(\theta^{i-1}|X)| \leq \varepsilon \tag{2.13}$$

where the current estimate of θ in i^{th} iteration is denoted by θ^i and the convergence threshold ε.

However, certain part of the complete data is hidden in the case of HMM, we speak about observing the incomplete part. The likelihood of the complete data is therefore a joint probability of observed and hidden part of the data. Joint probability of the complete data may also be formulated by summing over all possible values of z.

$$l(\theta|X) = \ln p(X|\theta) = \ln \sum_z p(X, z|, \theta) \tag{2.14}$$

Alternatively, the goal of such an algorithm is to maximize the log likelihood of the complete data by estimating the optimal parameter or vector of parameters denoted by θ. Hence, the marginal probability of X given vector of hidden variables Z:

$$\ln p(X|\theta) = \ln \sum_{j=1}^{M} p(x, z = j|\theta) \tag{2.15}$$

Since traditional ML procedure aims to maximize the marginal log-likelihood of complete data, part of which we do not observe, it would be unnecessarily hard to maximise. It is however possible to introduce several assumptions that will eventually suffice in providing direct solution to the maximisation problem. Let us start by constructing the lower bound for the objective function as in Equation 2.15 that is easy to optimise with respect to parameters represented. In order to find the function q, we start by multiplying the marginal likelihood by $\frac{q(z)}{q(z)}$. Such expression will allow for a construction of artificial weights and with the use of Jensen's inequality:

$$\ln \sum_{j=1}^{M} q(z=j) \frac{p(x,z=j|\theta)}{q(z=j)} \geq \sum_{j=1}^{M} q(z=j) \ln \frac{p(x,z=j|\theta)}{q(z=j)} = L(\theta,q) \tag{2.16}$$

$$\ln p(X|\theta) \geq L(\theta,q), for \forall q \in Q \tag{2.17}$$

We have established the Jensen's inequality within the usage of concavity of logarithmic function. The constructed lower bound $l(\theta,q)$ is factorable into the expected value of the log likelihood of complete data and entropy of probability function $q(x)$. Moreover, the maximisation of such likelihood function is completely determined by the former term since the latter is independent of θ.

$$l(\theta,q) = \sum_{j=1}^{M} q(z=j) \ln p(x,z=j|\theta) + \sum_{j=1}^{M} q(z=j) \ln \frac{1}{q(z=j)} \tag{2.18}$$

The optimization problem transforms into finding the function q for a fixed θ^{k-1}, given the k-th iteration, that maximizes $L(\theta,q)$:

$$q^k = \arg\max_q l(\theta,q) \tag{2.19}$$

In other words, we want to minimize the gap between the complete data log-likelihood function $l(\theta|X)$ and incomplete data $l(\theta,q)$ with respect to function q. Let us now elaborate more on the gap. The final expression that we shall arrive at gives a simplifying interpretation known as Kullback-Leibler divergence (hereafter KL divergence) measures the dissimilarities between two distributions and the measure may be interpreted as a geometrical statistical distance, it also is asymmetric and non-negative. Such measure is commonly used in image or signal processing in calculation of the expected excess surprise from using q as a probability distribution for our model given that the true or actual probability distribution is p. Substantially the measure is a difference of cross-entropy denoted by $H(p,q)$ and entropy by $H(p)$ which is always non-negative as a result of Gibb's inequality and is zero if and only if the two probability measure p and q are equal.

$$D_{KL}(p||q) = \sum_X p(x) \ln \frac{p(x)}{q(x)} \tag{2.20}$$

$$= \sum_x p(x) \ln \frac{1}{q(x)} - \sum_x p(x) \ln \frac{1}{p(x)} \tag{2.21}$$

$$= H(p,q) - H(p) \tag{2.22}$$

The lower bound introduced in Equation 2.18 and the complete data log-likelihood function $l(\theta|X)$ are obviously not the same following even intuitively from the fact that we are observing only the visible part of the data, the hidden part is still unknown. More rigorously, we have only defined the lower bound for the complete data log-likelihood function using arbitrary probability function $q(z)$, but the deeper examination of the lower bound with the use of KL divergence yields direct choice of the probability function $q(x)$. Let us decompose the lower bound in terms of the original log-likelihood function of the complete data:

$$l(\theta,q) = \sum_{j=1}^{M} q(z=j) \ln \frac{p(x,z=j|\theta)}{q(z=j)} \tag{2.23}$$

$$= \sum_{j=1}^{M} q(z=j) \ln \frac{p(z=j|x,\theta)p(x|\theta)}{q(z=j)} \tag{2.24}$$

$$= \sum_{j=1}^{M} q(z=j) \ln \frac{p(z=j|x,\theta)}{q(z=j)} + \sum_{j=1}^{M} q(z=j) \ln p(x|\theta) \tag{2.25}$$

$$= -D_{KL}(q(z)||p(z|x,\theta)) + \ln p(x|\theta) \tag{2.26}$$

Now we see that the statistical distance between the two likelihood functions is determined solely by the KL divergence. In order to find the minimal distance, ideally distance of zero length, we simply set the function $q(x)$ equal to the $p(z|X,\theta)$. This might be the posterior distribution Adame???

$$\ln p(x|\theta) = l(\theta,q) + \sum_{j=1}^{M} D_{KL}(q(z=j)||p(z=j|x,\theta)) \tag{2.27}$$

The last expression is just sum of KL divergences of function $g(z_i = j)$ and the posterior distribution of the latent variable z given the data and current parameter θ.

Recalling Equation 2.18, the optimisation goal lies in finding the probability distribution q that maximises $L(\theta,q)$ given the current estimate of the parameter θ, traditionally called as $E-step$. To extend and present a solution in finding the function q, following interpretation within the knowledge of Equation 2.26 is emphasized. We see that the maximum of the lower bound, i.e. of the log likelihood function, with respect to function $q(z)$ is identical to minimising the KL divergence introduced above because the first term in objective function to minimise is independent of $q(z)$:

$$q^{k+1} = \arg\max_q L(\theta_k, q) = \arg\max_q (\ln p(X|\theta) - \sum_{i=1}^{N} D_{KL}(q(z_i)||p(z_i|x_i,\theta))) \tag{2.28}$$

To visualize E-step, let us consider arbitrary log-likelihood function of complete data $\ln p(X|\theta)$ and $l(\theta^k, q^{k+1})$ for a fixed parameter of θ^k. Notice the visual representation of resulting selection of q^{k+1} as a solution to E-step. Consider also the new parameter estimate of θ^{k+1} which is a part of the next step called Maximisation step, also abbreviated as M-step.

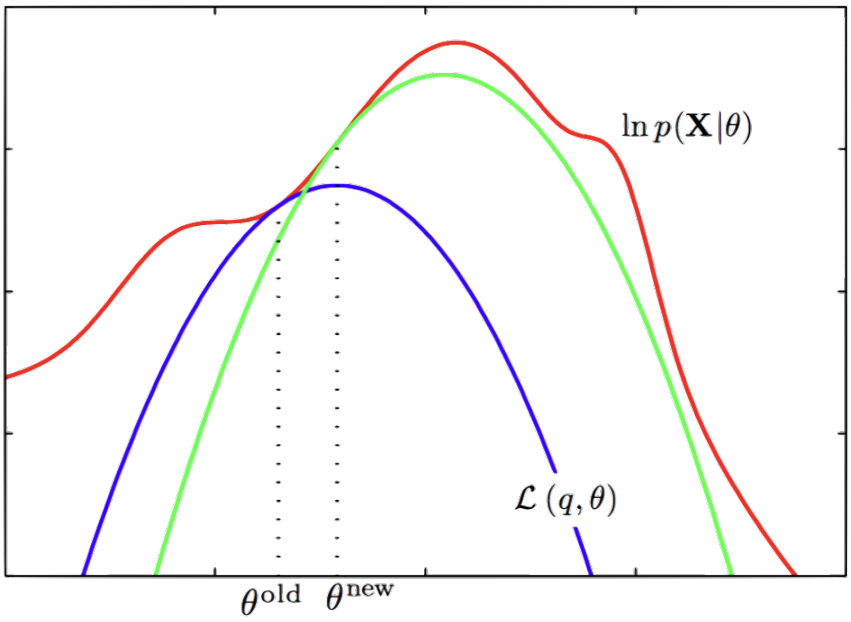


Fig. 2.2 *E-step as a problem of minimization of KL divergence represented as a gap between two likelihood functions*

Until now, we have considered the optimal manner in which we compute the conditional expectation $E_{Z|X,\theta_n}[\ln P(X,z|\theta)]$ by finding the appropriate function q. Next step is, as is typical for coordinate descent, selection of parameter θ_{k+1} by fixing the function q^{k+1} from E-step. The goal of M-step is to:

$$\theta^{k+1} = \arg\max_\theta L(\theta, q^{k+1}) \tag{2.29}$$

Thus, the EM algorithm involves two main steps:

1) Expectation step (E-step) - choose a function q, i.e. probability distribution, that maximizes $L(\theta, q)$, which may be also viewed as computing the conditional expectation $E_{Z|X,\theta_n}[\log p(X,z|\theta)]$ based on the current parameter of θ.

2) Maximization step (M-step) - estimating the parameter θ_{k+1} that maximizes the conditional expectation $E_{Z|X,\theta_n}[\log p(X,z|\theta)]$.

Both of these steps are repeated until convergence.

2.1.1 Baum-Welsch algorithm

In previous section we defined Expectation-Maximization algorithm used to compute Maximum Likelihood estimates given the incomplete data, i.e. supposing that part of the data is hidden. Two main steps are called E-step and M-step which were discussed generally, but we need to establish direct connection to the estimation of the Hidden Markov model parameters. We will show that former step is easily computed given variables α, β and γ established in last chapter, where we defined Viterbi and Forward-Backward algorithm. These computations result in estimating the conditional probability distribution of hidden states given our observations sequence and model parameters $\mathbb{P}(z_t = i|X, \theta^k)$. The E-step in EM algorithm is just a derivation of conditional probability of particular hidden state given the data and the model parameters. In order to compute such quantity we will slightly modify the γ definition to express it using Bayes Theorem for each observation at time t. Meaning that now we are:

$$\gamma_t(i) = \mathbb{P}(Z_t = i|X,\theta) = \frac{\mathbb{P}(X|Z_t = i,\theta)\mathbb{P}(Z_t = i|\theta)}{\sum_{j=1}^{k}\mathbb{P}(X|Z_t = j,\theta)\mathbb{P}(Z_t = j|\theta)} \tag{2.30}$$

where the numerator includes the conditional distribution of our data given particular hidden state i and the latter term, also known as prior distribution, is our estimate of the probability distribution of hidden states. In upcoming chapter about Gaussian Mixture Models we will show that this first term is a Gaussian distribution given hidden state i and the common choice for prior will be Multinomial distribution with k parameters where we will stress the importance of its conjugate prior the Dirichlet distribution. Although, in case our model assumptions we use already defined variables $\alpha_t(i)$ and $\beta_t(i)$. Visually, we can interpret $\gamma_t(i)$ using a similar form of a trellis as in section about Viterbi algorithm.

In Chapter 1 we denoted the transition matrix $N \times N$ as $A = \{a_{i,j}\} = \mathbb{P}(Z_t = j|Z_{t-1} = i)$, i.e. probability of transitioning to state j given that we were at state i at previous time step. The initial distribution of hidden states $\pi_i = \mathbb{P}(Z_1 = i)$, also called probability vector of length N, and emission matrix $N \times K$ defined as $B = b_i(x_t) = \mathbb{P}(X_t = x_t|Z_t = i)$, where K denotes

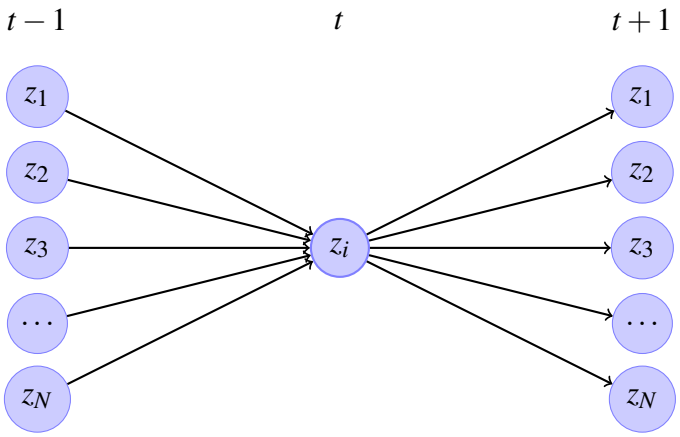


Fig. 2.3 Trellis interpreting the conditional probability of hidden state i at time t using Forward-Backward algorithm.

state space, i.e. all possible outcomes, of random variables $X_t, \forall t \in T$ (stochastic process???). First, we need to specify the equation for complete data log-likelihood function for the Hidden Markov Model, which we can intuitively represent as joint probability distribution of observed and hidden data given our model parameters, i.e. $\log p(X,Z|\theta)$. Also note that here θ is considered as a vector of parameters containing initial distribution of states, transition and emission matrices, $\theta = \{\pi, A, B\}$, and is time invariant therefore:

$$L(\theta|x_1,\ldots,x_T) = \pi A \prod_{t=1}^{T} b_{x_t} \tag{2.31}$$

Time only effects the likelihood function through the emission matrix where we have to condition by the realisation of the observed process $\{X\}_{t=1}^{T}$, therefore πA is a row vector and the product of column vectors. Unfortunately, the triplet of parameters does not form commuting matrices, so we represent them via the sums of its elements after the logarithmic transformation below.

$$l(\theta|x_1,\ldots,x_T) = \sum_{i=1}^{N} \log \pi_i + \sum_{i,j=1}^{N} \log a_{i,j} + \sum_{t=1}^{T}\sum_{i=1}^{N} \log b_{i,x_t} \tag{2.32}$$

Next step according to EM algorithm is to take the expectation of the log-likelihood function above with respect to $\mathbb{P}(Z_t = i|X,\theta)$. Afterwards we may use standard tool of partially differentiating such function with respect to model parameters.

$$\mathbb{E}_{Z|X,\theta}[l(\theta|x_1,\ldots,x_T)] = \sum_{i=1}^{N} \gamma_1(i) \log \pi_i + \sum_{t=2}^{T} \sum_{i,j=1}^{N} \gamma_t(i,j) \log a_{i,j} + \sum_{t=1}^{T} \sum_{i=1}^{N} \gamma_t(i) \log b_{i,x_t} \tag{2.33}$$

To summarize, the E-step of the EM algorithm in case of HMM lies in estimation of $\gamma_t(i)$ which is simply a probability of being at state i given our observation sequence and model parameters. Certainly, the iterative method of EM algorithm also needs an initial guess about the model parameters, meaning the initial distribution, transition and emission matrix. Since the hidden states are assumed discrete we may express transition matrix using a stochastic (Markov) matrix introduced in Chapter 1.

Note that the initial estimate of the parameters will never guarantee that a global maximum of the likelihood function is attained, therefore on of the optimal solutions to this problem is to initialize parameters multiple times and compare the maxima of the log likelihood function.

Once we have a formal estimate of the initial parameters, the EM algorithm performs E-step as defined above and subsequently M-step. Then maximization step is determined by the argument that maximizes the expectation of log likelihood function given by Equation 2.33 and also parameter constraints:

$$\sum_{i}^{N} \pi_i = 1 \tag{2.34}$$

$$\sum_{i}^{N} a_{i,j} = \mathbb{1} \quad \forall j \in I \tag{2.35}$$

$$\sum_{i}^{N} b_{i,x_t} = \mathbb{1} \quad \forall x_t \in \mathscr{X} \tag{2.36}$$

where $\mathbb{1}$ is a unit vector of length N. Given parameter constraints we maximize expected log likelihood function using Lagrange multipliers where the objective function is defined as:

$$\mathscr{L}(\theta, \lambda_k|x_1,\ldots,x_T) = E_{Z|X,\theta_n}[\log p(X,z|\theta)] - \lambda_1(\sum_{i}^{N} \pi_i - 1) \tag{2.37}$$

$$- \lambda_2(\sum_{i}^{N} a_{i,j} - \mathbb{1}) - \lambda_3(\sum_{i}^{N} b_{i,x_t} - \mathbb{1}) \tag{2.38}$$

where $k = \{1,2,3\}$. Taking partial derivatives with respect to each parameter on our model and setting that partial derivative equal to zero we arrive at parameter estimates which is also a final solution to our maximization step as follows:

$$\hat{\pi}_i = \gamma_1(i) \tag{2.39}$$

$$\hat{a}_{i,j} = \frac{\sum_{t=2}^{T} \gamma_t(i,j)}{\sum_{t=2}^{T} \gamma_t(i)} \tag{2.40}$$

$$\hat{b}_{i,x_t} = \frac{\sum_{t=1}^{T} \gamma_t(i) \mathbb{1}_{[X_t = x_t]}}{\sum_{t=1}^{T} \gamma_t(i)} \tag{2.41}$$

We repeat these steps until the desired convergence is achieved. To summarize the Expectation Maximization algorithm, the goal in such instance is to estimate model parameters by Equation 2.29 given a convergence criterion of $|l(\theta^k|X,q^k) - l(\theta^{k-1}|X,q^{k-1})| \leq \varepsilon$ after which we claim that the parameter estimates of k-th iteration are such that we have maximized the value of our log likelihood function given the observable data.

2.2 Observable states

There is a huge number of observable variables that one could abstract from cryptocurrency market. A possibility of discrete states within the given state space is plausible and feasible, it would, given our model constraints, provide poor inference since additional information would remain hidden. Imagine a situation where our observable states are defined as a relative change in price or a sudden drop/uprise in the traded volume on the exchange. In order to discreticize our states and construct transition and emission probabilities we are forced to construct intervals that would well represent the boundaries upon which the model defines structure and predictions.

Assuming that the price increase in the idea predefined Hidden Markov Model will assume

However it is much more efficient to assume continuity in our predefined observable states. It is nowadays empirically proved, as in (citace) that using technical indicators as predictors for the future spot price yields more accurate machine learning models. As will

be demonstrated each of the technical indicators can be classified into several families of indicators, such as momentum, volume, volatility and cycle indicators. For our purposes we will consider mainly momentum indicators that are calculated using Open, High, Low, Close prices (hereinafter "OHLC") and Volume indicators. There is a huge variety of technical indicators to choose from therefore the selection was made according the most used and well known indicators or their transformed versions.

In our case we will consider following observable states that will be defined and elaborated on in the upcoming sections:

1) Moving Average Convergence/Divergence (MACD)

2) Stochastic Oscillator

3) Chaikin Oscillator

4) Relative Strength Index (RSI)

5) Aroon Oscillator

2.3 Moving Average Convergence Divergence

Also knows as MACD is a trend-following momentum indicator that represents the differences between two exponential moving averages (hereinafter "EMA"). The most common and traditional moving averages are 26-period EMA and 12-period EMA.

The indicator is often used with so-called ""signal line" that is constructed as a 9-period EMA and is used as a trigger for a buy and sell signal. In practical application a trader decides to buy a stock if the signal line crosses MACD line from above and sell if it crosses from below, assuming simplistic trading strategy using only MACD. EMA also called exponentially weighted moving average is a type of moving average that differs from weighted moving average WMA by the distribution of weights to past observations. While WMA considers the linearly decreasing distribution of weights, the EMA assumes exponential decrease in weights. Furthermore it is necessary to elaborate over the values of weights because it might not always be unambiguous. WMA distributes weights chronologically and linearly, e.g. 10-period WMA gives weight 1 to the earliest observation and 10 to the most recent observation, the case within EMA is often not that simple. The weights given to each observation are computed as $(1-\lambda)^i$ where $i \in \mathbb{N}_0$ and is bounded from above by the assumed period of interest, e.g. 3, 10, 26-period denoted as T for the sake of . As i increases identically with the time lag the value of weights decreases. The important role that ought to be questioned

is the parameter λ that is defined as $\frac{k}{T+1}$ where k represents the so called "smoothing" parameter. Traders and analysts use value 2 for the smoothing parameter but the number may be defined on the interval $(0,T)$. Higher values of k mean bigger weights given to most recent observations.

The Figure 2.1 illustrates MACD line, signal line as well as "MACD histogram" which is displayed as a bar chart indicating the difference of the former ones. Traders use such a distance to identify whether the bullish or bearish momentum is high, i.e. bigger the distances of these two lines higher the price momentum.

MACD has its unfortunate limitations that mainly arise from the non-trending moments. When the price enters sideways movement the MACD histogram signals decreases distances between MACD and signal line, the trend reversal is possible but the price moves sideways which eventually results in false positive signal. Moreover when the price moves sideways for longer periods MACD may signal too many false trend reversals. The most common practice for traders is to combine MACD signals with other indicators such as Relative Strength index (RSI) that measures overbought or oversold market. The RSI uses average price gains and losses usually over 14 periods and yields values between 0 and 100, indicating overbought market for values 70 (80) to 100 and 30 (20) to 0 for oversold market. The idea is that when the distances between MACD line and signal line increase and RSI signals overbought market the trader might consider this as a strong trend reversal signal. The idea is that signals from MACD strategy often produce false signals when price suddenly moves sideways and RSI helps to indicate the false positive signal.

2.4 Stochastic Oscillator

A Stochastic Oscillator is a momentum indicator that compares the most recent closing price of a security with its predeceasing ones. Naturally, the range of preceding closing prices or the range of closing prices is 14 periods but it is regular that such an assumption is often edited to best fit the current needs of a trader. Also slight variation in taking the (weighted) moving average of the oscillator values is often introduced. The indicator is used to generate trading signals that refer to the current overbought state of the market, which means that the indictor values range from 0 to 100 where the values closer to the number 0 indicate oversold market and inversely values closer to 100 overbought market.

Stochastic Oscillator is computed as follows:

$$SO_t = \frac{C_{t-1} - L_{14}}{H_{14} - L_{14}} \tag{2.42}$$

Fig. 2.4 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with subplot containing two lines for MACD and Signal line and a bar chart as their difference*

where C_{t-1} denotes the most recent closing price of a security, H_{14} and L_{14} are the highest and lowest price traded during 14-period interval respectively. SO_t is sometimes referred to as a "fast" stochastic indicator. As said before this interval may be changed arbitrarily. Traders also developed so called "slow" Stochastic Oscillator which is defined as a 3-period moving average of SO_t. Thus when Stochastic Oscillator crosses the smooth "Slow" Stochastic Oscillator a trading signal is generated.

Considering values above 80, the indicator signals overbought market and oversold market when the value drops below 20. Although it remains to hold true that the indicator often produces false indications that may be caused by periods of time where the price remains overbought/oversold for some time and trading with respect to such oscillator may result in losses. It is rather recommended to observe the values of stochastic oscillator and use it for trend reversal indication.

Fig. 2.5 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with subplot with line indicating Stochastic Oscillator*

2.5 Chaikin Oscillator

Chaikin Oscillator is a momentum based indicator of the Accumulation/Distribution Line (hereinafter "A/D line"), which is a cumulative indicator that aims to identify potential divergences between stock price and trading volume. The oscillator is calculated as a difference between 3- day and 10-day Exponential Moving Average of A/D line.

The calculation of the Chaikin Oscillator may be broken down into several steps:

(i) First, we ought to calculate the Money Flow Multiplier for each time step denoted by "N".

$$N_t = \frac{(Close_t - Low_t) - (High_t - Close_t)}{High_t - Low_t} \tag{2.43}$$

(ii) Now we may multiply N_t by the trading volume in given period of time to get the Money Flow Volume denoted as M_t. With that we recursively construct the A/D line as:

$$ADL_t = M_{t-1} + M_t \tag{2.44}$$

(iii) Given the constructed A/D line, we compute the Chaikin Oscillator values as a difference of 3-day and 10-day exponential moving averages.

$$CO_t = \frac{\sum_{i=0}^{3}(1-\alpha)^i * Close_{t-i}}{\sum_{i=0}^{3}(1-\alpha)^i} - \frac{\sum_{i=0}^{10}(1-\beta)^i * Close_{t-i}}{\sum_{i=0}^{10}(1-\beta)^i} \tag{2.45}$$

where we assume that the weights denoted as α and β are computed as $2/(days+1)$. Numerator as a smoothing factor is often declared as 2. However, the indicator may be set to absolutely different number between 0 and 1 according to the needs and assumptions made by the trader/analyst, therefore setting the parameter close to 1 is putting more weight to the most recent price.

One way to interpret the indicator is to trade with respect to the time when the Chaikin Oscillator crosses zero from below and above which signals buy and sell signals respectively.

Fig. 2.6 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with subplot indicating Chaikin Oscillator*

2.6 Relative Strength Index

Given the recent price changes Relative Strength Index measures its magnitude in order to indicate the overbought or oversold market. In technical analysis such indicator is represented by an oscillator ranging from values 0 to 100. Empirically it was determined that values above 70 and below 30 signal overbought and oversold asset respectively. Therefore, we may also use RSI to produce buy and sell signals from former logic, which means that when the RSI crosses 30 from below, buy signal is generated as well as sell signal in case it crosses 70 from above. We also could measure the strength and continuation of the trend for cases in which the RSI crosses value of 50. Such interpretation results from the RSI formula where the value of 50 means that the average gain equals the average loss in the last period.

The formula below explains the procedure within which the values of RSI are calculated. It is obvious that the RSI rises as the number of positive closing prices increases, i.e. the relative change in prices is positive, and falls if otherwise. The standard time interval for the calculation is 14 preceding periods with respect to t, hereby denoted as T.

$$RSI_t = 100 - \frac{100}{1 + r_t} \tag{2.46}$$

where r is a ratio of average gains and losses as follows:

$$r_t = \left| \frac{\sum_{i=1}^{T} (\frac{P_{t-i+1}}{P_{t-i}} - 1) \mathbb{1}_{[P_{t-i+1} - P_{t-i} > 0]}}{\sum_{i=1}^{T} (\frac{P_{t-i+1}}{P_{t-i}} - 1) \mathbb{1}_{[P_{t-i+1} - P_{t-i} < 0]}} \right| \tag{2.47}$$

given that P_t denotes the value of an asset at time t.

As stated before there are several drawbacks of using the RSI as a trading indicators only by itself, it happens that the price usually rises and stays overbought for a substantial period of time in times of significant and strong bullish trend. RSI as an oscillator is used as an auxiliary trading tool below the price chart:

2.7 Aroon Indicator

Aroon indicator is used for trend reversal identification and a measure of its strength. Indicator is composed out of two lines aroon up and aroon down that measure the time between new highs or lows respectively. Alternatively, they measure the strength of a bullish or bearish trend. Obviously the main idea of the indicator is based upon the fact that bullish trends are naturelly formed by subsequently creating new highs while bearish trends form new lows. Aroon Up and Aroon Down are computed as follows:

Fig. 2.7 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with a subplot containing RSI*

$$AroonUp = \frac{25-h}{25} * 100 \tag{2.48}$$

$$AroonDown = \frac{25-l}{25} * 100 \tag{2.49}$$

Where h represents the number of periods from the last 25-period High and l the number of periods from last 25-period Low.

The interpretation of the indicator is very intuitive since the situation in which the Aroon Up line is above Aroon Down line signals bullish trend and when these two lines cross the signal of the trend reversal is generated. That also implies that for higher values of Aroon Up the bigger the strength and for lower values the uptrend is weaker and vice versa. In practice the crossover of these two lines is what generates the buy or sell signals, i.e. if Aroon Up crosses Aroon Down line from below a buy signal is generated and vice versa.

Although, Figure 2.4 graphically illustrates the Aroon Up and Down lines well, it is simpler to transform these two lines into one oscillator that would produce buy or sell signal in the case of zero crossover from above and from below. That is achieved by subtracting Aroon Up and Aroon Down line creating Aroon Oscillator.

Fig. 2.8 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with subplots containing two lines for Aroon Up/Down Indicator and Aroon Oscillator*

Chapter 3

Mixture Models

As shown in the last chapter the joint probability distribution of the complete data is not strictly concave thus we can not guarantee the convergence to the global optimum. Approach we might take is to randomly initialize the parameters multiple times to see if the maxima of the log likelihood function increases. Fig 2.2 appropriately displays possible shape of the joint probability distribution of the complete data. There are visible multiple local optima of the log-likelihood function caused by the hidden states. Such probability distributions are often called mixture distributions.

They were implicitly introduced in the last chapter while estimating the parameters of the Hidden Markov Model. Although, most of the time we consider unimodal distributions for our data since it may be justifiable empirically or considering the mixture distributions could unnecessarily complicate the computation for an indistinguishable improvement, the statistical inferences about the subpopulations within the population require such tools in cases where the subpopulations significantly differ. However, in these models we pose no requirement of the knowledge about the number of subpopulations within the population or their actual distribution. They are interpreted the same way as for the HMM as hidden variables.

3.1 Mixture Distributions

Let Y_j denote a p-dimensional random vector with probability density function $f(y_j)$ on $\mathbb{R}^p$ selected from random sample $Y_1, \ldots, Y_n$. A realization of such a random vector is denoted by y_j.

Mixture distribution or density is defined as a convex combination of g component distributions or densities respectively as follows:

$$f(y_j) = \sum_{i=1}^{g} \pi_i f_i(y_j) \tag{3.1}$$

where $f_i(y_j)$ are component densities of the mixture and π_i are mixing proportions or mixture weights with following properties:

$$0 \leq \pi_i \leq 1 \quad \forall i \in \{1, 2, \ldots, g\} \tag{3.2}$$

and

$$\sum_{i=1}^{g} \pi_i = 1 \tag{3.3}$$

Therefore, the density (3.1) of $f(y_j)$ is referred to as a g-component finite mixture density, conversely $F(y_j)$ as a g-component finite mixture distribution.

Formula (3.1) assumes that the number of components g is fixed. In many applications this is not the case, and we have to infer the number of components from the data. Furthermore, mixing proportions π_i are also unknown and have to be estimated along with the respective parameters of the component densities $f_i(y_j)$.

Analogously to previous definition we can define a finite mixture of random variables Y_j as follows:

$$f(y_j, z) = g(z) f(y_j|z) \tag{3.4}$$

where Z is defined as a random variable with multinomial distribution with a vector of parameters $\pi = \{\pi_1, \ldots, \pi_g\}$ and $g(z)$ as its probability density function. Summing over all possible values of Z we obtain the marginal density of random vector Y_j:

$$f(y_j) = \sum_{i=1}^{g} P(Z = i) f(y_j|Z = i) \tag{3.5}$$

where $P(Z = i) = \pi_i$ and $f(y_j|Z = i) = f_i(y_j)$ are the mixing proportions and component densities respectively Therefore seeing the equivalence of Equation (3.1) and (3.4).

3.2 Gaussian Mixture Models

The Gaussian mixture model (GMM) is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions. The component distributions are often chosen to be members of the same parametric family, such as Gaussian

distributions with respective mean and covariance parameter. Assuming that the number of components is K the probability density function of the Gaussian mixture model can be expressed as follows:

$$f(x|\mu,\Sigma) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k,\Sigma_k) \tag{3.6}$$

where μ_k and Σ_k are the mean and covariance matrix (symmetric and positive semi-definite) of the k-th Gaussian component.

Thus, the vector of parameters is $\theta = \{k \in \mathbb{N} : (\pi_k, \mu_k, \Sigma_k)\}$. Probability density function of each component of GMM depends on the dimensionality of the data. Let the dimensionality of the data be $d > 1$ then the random vector $Y = (y_1, \ldots, y_d)$ has the probability density function of the k-th component defined as following multivariate Gaussian distribution:

$$f_k(y) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(y_j - \mu_k)^T \Sigma_k^{-1}(y_j - \mu_k)\right) \tag{3.7}$$

where $|\Sigma_k|$ denotes the determinant of the covariance matrix Σ_k.

3.2.1 Motivating EM algorithm

Given the data $\mathbf{Y} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_N\}$, where each element is an independently and identically distributed p-dimensional random vector, s.t. $Y_n \in \mathbb{R}^p$ and:

$$\mathbf{Y}_n \sim \mathcal{N}_p(\mu,\Sigma) \quad \forall n \in \{1,\ldots,N\} \tag{3.8}$$

The goal is to estimate θ of the mixture model, i.e. for each mixture component k we need to estimate its mean μ_k, covariance matrix Σ_k and mixing proportion π_k.

Note that now we will assume that each element of random vector $\mathbf{Y}_n$ is generated by one of the mixture components, and we know which one. This is called the complete data since we have information about the hidden variable Z which indicates the component from which the data point was generated. Therefore, the complete data likelihood function is defined as follows:

$$L_c(\theta|\mathbf{Y},\mathbf{Z}) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{Y}_i|\mu_k,\Sigma_k) \tag{3.9}$$

and the log-likelihood function is:

$$\ell_c(\theta|\mathbf{Y},\mathbf{Z}) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{Y}_n|\mu_k,\Sigma_k) \tag{3.10}$$

Equation (3.10), shows that the log-likelihood function has a form of a sum of logarithms which results in no strict analytical solution for the maximum likelihood estimation of model parameters.

Given the assumption that we know the hidden variable $\mathbf{Z}$ directly from the data defined as a K-dimensional random vector $\mathbf{Z} = (z_1, \ldots, z_K)$ with multinomial distribution and parameters $\pi = \{\pi_1, \ldots, \pi_K\}$, the log-likelihood function can be rewritten as follows:

$$\ell_c(\theta|\mathbf{Y},\mathbf{Z}) = \sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik}\left(\log \pi_k + \log \mathcal{N}(\mathbf{Y}_n|\mu_k,\Sigma_k)\right) \tag{3.11}$$

with variable z_{ik} defined as follows:

$$z_{ik} = \begin{cases} 1 & \text{if } \mathbf{Y}_i \text{ was generated by the } k\text{-th component} \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

also notice that for each sample, random vector z_i is distributed identically and independently according to the multinomial distribution with vector of parameters π.

3.2.2 EM algorithm for GMM

The EM algorithm for GMM is very similar to the one defined for Hidden Markov Models in the previous chapter. We start with the definition of the **E-step** of the EM algorithm. The conditional expected complete data log-likelihood function given log-likelihood function by Equation (3.11) as follows:

$$Q(\theta|\theta^{(t)}) = \mathbb{E}[\ell_c(\theta|\mathbf{Y},\mathbf{Z})|\mathbf{Y},\theta^{(t)}] \tag{3.13}$$

hence inserting Equation (3.11) into (3.13) we obtain:

$$Q(\theta|\theta^{(t)}) = \sum_{i=1}^{N}\sum_{k=1}^{K} \mathbb{E}[z_{ik}|\mathbf{Y}_i,\theta^{(t)}]\left(\log \pi_k + \log \mathcal{N}(\mathbf{Y}_i|\mu_k,\Sigma_k)\right) \tag{3.14}$$

where $\theta^{(t)}$ denotes the parameter vector at the t-th iteration of the algorithm and $\mathbb{E}[z_{ik}|\mathbf{Y}_i,\theta^{(t)}]$ is the expected value of the hidden variable z_{ik} given the data $\mathbf{Y}_i$ and the parameter vector $\theta^{(t)}$. The expected value of the hidden variable z_{ik} is defined as follows:

$$\mathbb{E}[z_{ik}|\mathbf{Y}_i,\theta^{(t)}] = P(z_{ik}=1|Y_i,\theta^{(t)}) = \frac{\pi_k^{(t)}\mathcal{N}(\mathbf{Y}_i|\mu_k^{(t)},\Sigma_k^{(t)})}{\sum_{j=1}^{K}\pi_j^{(t)}\mathcal{N}(\mathbf{Y}_i|\mu_j^{(t)},\Sigma_j^{(t)})} \tag{3.15}$$

where $\pi_k^{(t)}$, $\mu_k^{(t)}$ and $\Sigma_k^{(t)}$ are the mixing proportion, mean and covariance matrix of the k-th component at the t-th iteration of the algorithm respectively. Taking a step back and looking at the Equation (3.5) we observe that the conditional expected value of z_{ik} given data $\mathbf{Y}_i$ and parameter vector $\theta^{(t)}$ is the posterior probability of the hidden variable z_{ik}.

The **M-step** of the EM algorithm then aims to maximize the conditional expected complete data log-likelihood function $Q(\theta|\theta^{(t)})$ with respect to the parameter vector θ. Such maximization is formally defined as follows:

$$\theta^{(t+1)} = \arg\max_{\theta \in \Theta} Q(\theta|\theta^{(t)}) \tag{3.16}$$

Taking partial derivatives of $Q(\theta|\theta^{(t)})$ with respect to the parameters π_k, μ_k and Σ_k and setting them to zero we obtain the following equations:

$$\hat{\pi}_k^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}[z_{ik}|\mathbf{Y}_i, \theta^{(t)}] \tag{3.17}$$

$$\hat{\mu}_k^{(t+1)} = \frac{\sum_{i=1}^{N} \mathbb{E}[z_{ik}|\mathbf{Y}_i, \theta^{(t)}]\mathbf{Y}_i}{\sum_{i=1}^{N} \mathbb{E}[z_{ik}|\mathbf{Y}_i, \theta^{(t)}]} \tag{3.18}$$

$$\hat{\Sigma}_k^{(t+1)} = \frac{\sum_{i=1}^{N} \mathbb{E}[z_{ik}|\mathbf{Y}_i, \theta^{(t)}](\mathbf{Y}_i - \mu_k^{(t+1)})(\mathbf{Y}_i - \mu_k^{(t+1)})^T}{\sum_{i=1}^{N} \mathbb{E}[z_{ik}|\mathbf{Y}_i, \theta^{(t)}]} \tag{3.19}$$

3.2.3 Markov Chain Monte Carlo

Although, EM algorithm is a very powerful tool for estimating parameters of the mixture model, it is not guaranteed to converge to the global optimum of the log-likelihood function. Therefore, we will introduce a Markov Chain Monte Carlo (MCMC) method for estimating the parameters of the mixture model by sampling from the posterior distribution of the parameters. As opposed to the previous subsection, the MCMC method also provides a measure of uncertainty of the estimated parameters.

3.2.3.1 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is a Markov Chain Monte Carlo method for sampling from a probability distribution. The algorithm is based on the idea of constructing a Markov chain that has a stationary distribution equal to the target distribution. The algorithm is defined as follows:

1. Initialize the Markov chain with an arbitrary state $\theta^{(0)}$.

2. For $t = 1, 2, \ldots$:

   (a) Sample a candidate state θ^* from a proposal distribution $q(\theta^*|\theta^{(t-1)})$.

   (b) Compute the acceptance probability $\alpha(\theta^{(t-1)}, \theta^*)$.

   (c) Sample a random number u from the uniform distribution $U(0,1)$.

   (d) If $u < \alpha(\theta^{(t-1)}, \theta^*)$ then set $\theta^{(t)} = \theta^*$, otherwise set $\theta^{(t)} = \theta^{(t-1)}$.

The last step (d) is often called *Metropolis rejection* and the acceptance probability $\alpha(\theta^{(t-1)}, \theta^*)$ in step (b) is defined as follows:

$$\alpha(\theta^{(t-1)}, \theta^*) = \min\left(1, \frac{p(\theta^*)q(\theta^{(t-1)}|\theta^*)}{p(\theta^{(t-1)})q(\theta^*|\theta^{(t-1)})}\right) \tag{3.20}$$

where $p(\theta)$ is the target distribution and $q(\theta^*|\theta^{(t-1)})$ is the proposal distribution from which we can easily sample. The second term in the minimum function of Equation 3.20 is called Hastings ratio. Metropolis rejection ensures here that the probability of accepting a candidate state θ^* is equal to $\alpha(\theta^{(t-1)}, \theta^*)$ and rejecting it with probability $1 - \alpha(\theta^{(t-1)}, \theta^*)$. Proposal distributions are usually chosen to be symmetric, i.e. $q(\theta^*|\theta^{(t-1)}) = q(\theta^{(t-1)}|\theta^*)$, so that the computation of Hastings ratio is simplified (to Metropolis ratio) and the acceptance probability is therefore following:

$$\alpha(\theta^{(t-1)}, \theta^*) = \min\left(1, \frac{p(\theta^*)}{p(\theta^{(t-1)})}\right) \tag{3.21}$$

Given the Gaussian mixture model, we can use the Metropolis-Hastings algorithm to sample from the posterior distribution of the parameters π_k, μ_k and Σ_k. The target distribution is the posterior distribution of the parameters, i.e. $p(\pi_k, \mu_k, \Sigma_k|\mathbf{Y})$. This distribution is proportional to the product of the prior distribution and the likelihood function, i.e. $p(\pi_k, \mu_k, \Sigma_k|\mathbf{Y}) \propto p(\pi_k, \mu_k, \Sigma_k)p(\mathbf{Y}|\pi_k, \mu_k, \Sigma_k)$ as follows from the Bayes' theorem. The prior distribution is the conjugate prior of the multivariate Gaussian distribution, i.e. the Normal-Wishart distribution when we want to infer both the mean and the covariance matrix of the Gaussian distribution. The Normal-Wishart distribution is defined as follows:

$$\mathcal{NW}(\mu, \Sigma|\mu_0, \kappa_0, \nu_0, \Lambda_0) = \mathcal{N}(\mu|\mu_0, (\kappa_0\Sigma)^{-1})\mathcal{W}(\Sigma|\nu_0, \Lambda_0) \tag{3.22}$$

where μ_0 is the prior mean, κ_0 is the prior precision, ν_0 is the prior degrees of freedom and Λ_0 is the prior scale matrix. The likelihood function is the product of the component Gaussian distributions given by Equation 3.9. Together with the prior distribution, the posterior distribution is given by the following:

$$p(\pi_k, \mu_k, \Sigma_k|\mathbf{Y}) \propto \prod_{i=1}^{N}\prod_{k=1}^{K}\left[\pi_k \mathcal{N}(\mathbf{Y}_i|\mu_k, \Sigma_k)\right]^{\mathbb{E}[z_{ik}|\mathbf{Y}_i, \theta^{(t)}]} \mathcal{NW}(\mu_k, \Sigma_k|\mu_0, \kappa_0, \nu_0, \Lambda_0) \quad (3.23)$$

where $\mathbb{E}[z_{ik}|\mathbf{Y}_i, \theta^{(t)}]$ is the posterior probability of the i-th observation belonging to the k-th component Gaussian distribution given the current estimate of the parameters $\theta^{(t)}$ as per Equation 3.15. The proposal distribution is often defined as a multivariate Gaussian distribution with mean $\theta^{(t-1)}$ and covariance matrix Σ which effects the convergence of the algorithm since it determines the size of the step taken in the parameter space. If Σ is too small the algorithm might not explore the parameter space sufficiently and the Markov chain might get stuck in a local optimum. On the other hand, if Σ is too large then the Markov chain might not converge at all. The choice of the proposal distribution is not unique, and it is usually determined empirically given some knowledge about the target distribution. Initialization of the Markov chain in the first step of the algorithm is also crucial since if the initial state is far from the region of high probability of the target distribution then the Markov chain might not converge at all. Lastly, notice that the samples from proposal conditional distribution are correlated since the next state of the Markov chain depends on the previous state which is a consequence of the Markov property and differs from the independent sampling of the parameters.

In summary, the Metropolis-Hastings algorithm is very flexible and simple to implement, and it is often used as a baseline for more sophisticated MCMC methods. There are several variants of the Metropolis-Hastings algorithm, e.g. Metropolis-within-Gibbs sampling, which is a special case of the Metropolis-Hastings algorithm where the proposal distribution is chosen to be the conditional distribution of the parameters given the current state of the Markov chain.

3.2.3.2 Gibbs Sampling

Chapter 4

Proposed versions of Hidden Markov Models

4.1 Gaussian HMM

4.1.1 Poisson HMM

4.1.2 Context-sensitive HMM

Chapter 5

Trading strategies based on Hidden Markov Models

5.1 Market regimes and trading strategies

5.1.1 Market states

5.1.2 Trading strategies

References

[1] *TradingView*. Accessed: 2023-05-20. 2023. URL: https://www.tradingview.com/chart/?symbol=BINANCE%3ABTCUSDT.

[2] Charles Louis Xavier Joseph de la Vallée Poussin. A strong form of the prime number theorem, 19th century.

[3] Peter Henderson. *Object-oriented specification and design with C++*. London: McGraw-Hill, 1993.

[4] C. J. Read. "A solution to the invariant subspace problem on the space l_1". In: *Bull. London Math. Soc.* 17 (1985), pp. 305–317.

Appendix A

Maximum likelihood results

Appendix B

Python code