Prague University of Economics and Business

Faculty of Finance and Accounting

Department of Banking and Insurance

MASTER THESIS

2023

Bc. Adam Pešek

Prague University of Economics and Business

Faculty of Finance and Accounting

Department of Banking and Insurance

Financial Engineering

Hidden Markov Models for Cryptocurrency Trading

Author: Bc. Adam Pešek

Supervisor: Ing. Milan Fičura, PhD.

Defense Date: September 2023

Declaration

I hereby declare that the master thesis "Hidden Markov Models for Cryptocurrency Trading" presented herein is my own work, or fully and specifically acknowledged wherever adapted from other sources. This work has not been published or submitted elsewhere for the requirement of a degree program.

In Prague on September 2023

. . . .

(author signature)

Abstract

Hidden Markov models is a statistical signal prediction model for sequential data, which has been widely used to predict economic regimes and stock prices. In this work, we introduce the application of different versions of Hidden Markov Models in cryptocurrency trading based on the market state predictions. We first introduce the basic concepts of Hidden Markov Models and the Baum-Welch algorithm for parameter estimation along with the Viterbi algorithm for state prediction. We consider several models with different assumptions on the distribution of the observations, including the Gaussian HMM, Context-Sensitive HMM and other variants. Having trained the models on historical data, we then apply the models to predict the market states of the cryptocurrency market and use these predictions alongside predefined trading strategies to trade on the Binance spot market. Lastly, we evaluate the performance of the models and trading strategies using the Sharpe ratio and MSE and compare the results with the benchmark strategy of buy-and-hold.

Table of contents

List of figures

List of tables

Chapter 1

Markov Processes

As described in (**Ross2014**), a Markov chain is a stochastic process describing a sequence of possible events where the probability of each event depends solely on the state attained in the previous event. This is termed as the 'Markov Property', which encapsulates the memorylessness of the process — the future is independent of the past given the present.

Markov chains provide a robust framework for modeling the probabilistic nature of cryptocurrency markets. The rapid fluctuation of cryptocurrency prices, driven by a myriad of factors such as market sentiment, technological advancements, regulatory news, and macroeconomic trends, may be represented effectively using Markov chains. (**Zhang2023**)

In this chapter, we start by laying down the fundamental theoretical foundations of Markov chains. This includes an explanation of key concepts such as state spaces, transition probabilities, and the difference between discrete and continuous time Markov chains. Understanding these basic building blocks is pivotal in leveraging Markov chains for cryptocurrency market analysis and trading.

Next, we focus on specific properties of Markov chains like irreducibility, periodicity, and stationarity, which play critical roles in determining the long-term behavior of the chain.

1.1 Discrete-time Markov Chains

Since we know that Markov Chain is a stochastic process, we ought to define what a stochastic process is, in itself, first.

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

Let I be a countable set and $\mathbb{S}$ a σ-algebra on I. Each $i \in I$ is called a *state* and $(I, \mathbb{S})$ a *state-space*. Therefore, we have two measurable topological spaces $(\Omega, \mathscr{F})$ and $(I, \mathbb{S})$ and a random variable $X : \Omega \to I$ assuming that X is measurable function. Thus, we call $(I, \mathbb{S})$ a state space and $(\Omega, \mathscr{F})$ an underlying space. Therefore, we may set:

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

Since we are allowing only for the discrete realizations of the random variable X, given previous assumptions and that $\sum\limits_{i \in I} \mu_X(i) = 1$, μ_X is a *probability mass function.* (**Norris2012**)

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 μ_X such that for all $t \in T$ and $i_0, i_1, \ldots, i_{t+1} \in I$ it holds that:

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

which holds for all $t \in T$ and $i_0, i_1, \ldots, i_{t+1} \in I$. (**Praskova2012**) 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. (**Haggstrom2002**) Conditional property of Markov Chains may be equivalently expressed using current state as $i \in I$ and a previous state $j \in I$ as:

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

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 (**Tolver2016**):

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

One important distinction by (**Weinan2019**) 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[1]. 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 **A** that satisfies property given by Equation 1.5 is called *stochastic matrix.* (**Gagniuc2017**) Furthermore, we ought to define a probability distribution $\mathbf{p} = \{p_i, i \in I\}$ as a vector of probabilities of observing each state at time $t = 0$ such that:

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

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.

According to (**Praskova2012**), once we have transition matrix A and initial distribution **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 **p** if and only if all finite dimensional distributions of $\{X_t, t \in \mathbb{N}_0\}$ are consistent with the following equation:

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

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 called *Chapman-Kolmogorov equation* as in (**Yin2004**). Above stated equation also holds for non-homogeneous Markov Chains with the only difference that the transition probabilities $p_{i,j}$ are time dependent:

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

Another substantial property of homogeneous Markov Chains is that their n-th order transition probabilities can be expressed as a product of their first order transition probabilities:

[1] In some sources the homogeneity with respect to time is emphasized s.t. the term is time-homogeneous or time-heterogenuous Markov Chains

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

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 $P(X_m = i) \geq 0$. (**Praskova2012**)

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:

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

using a maximum likelihood estimator of transition matrix according to (**Gagniuc2017**):

$$\hat{p}_{i,j} = \frac{\sum_{k=1}^{n-1} \mathbb{1}_{\{X_k = i, X_{k+1} = j\}}}{\sum_{k=1}^{n-1} \mathbb{1}_{\{X_k = i\}}} \tag{1.12}$$

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:

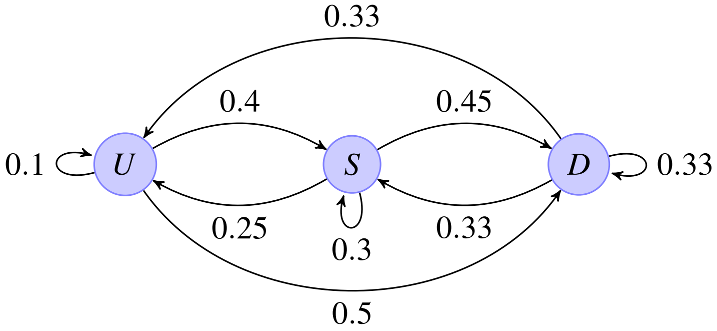


Fig. 1.1 Transition diagram of the Markov Chain given probabilities in matrix A.

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.

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**:

$$\begin{aligned}\mathbb{P}(X_{t_0}=x_0,\dots,X_{t_n}=x_n|A,p) &= \mathbb{P}(X_0=x_0)\prod_{k=1}^{4}\mathbb{P}(X_k=x_k|X_{k-1}=x_{k-1}) \\ &= p_{x_0}p_{1,2}\ p_{2,3}\ p_{3,3}p_{3,1} \\ &= 0.2*0.4*0.45*\ 0.33*0.33 \\ &= 0.002376\end{aligned} \tag{1.13}$$

Where the probability of observing state x_0 is determined by our initial distribution **p** 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 **A** and a sequence of events $\{U,S,D,D,U\}$ observed until 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.

$$\mathbb{P}(X_{t_{k+1}}|X_{t_k},\dots,X_{t_{k-4}}) = \mathbb{P}(X_{t_{k+1}}|X_{t_k}) \tag{1.14}$$

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 Chains 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:

$$i_{k+1} = \arg\max_{i\in I} P(X_{t_{k+1}} = i_{k+1}|X_{t_k} = i_k) \tag{1.15}$$

1.1.1 Classification of states

Markov Chains may be classified into several categories based on their properties. Firstly, (**Praskova2012**) and others, 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 k-th return to state $j \in I$:

$$\tau_j(k+1) = \inf\{n \geq \tau_j(k) : X_n = j\}, \quad k \in \mathbb{N}_0 \tag{1.16}$$

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 $\mathscr{F}_n = \sigma(X_0, \dots, X_n)$ generated by random variables $X_0, X_1, \dots, X_n$ is called a *recurrence time*.

Thus, (**Bremaud1999**) and (**Tolver2016**), state that $j \in I$ is *transient* if there is a non-zero probability that the process will never return to state j once it has left it, i.e.:

$$\mathbb{P}(\tau_j(1) = \infty | X_0 = j) > 0, \quad \sum_{n=0}^{\infty} p_{j,j}^{(n)} \leq \infty \tag{1.17}$$

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.:

$$\mathbb{P}(\tau_j(1) \leq \infty | X_0 = j) = 1, \quad \sum_{n=0}^{\infty} p_{j,j}^{(n)} = \infty \tag{1.18}$$

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 as (**Gebali2008**). The greatest common divisor of the number of times a state can return to itself 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 substantial since it implies that the state will be visited infinitely often and the expected time between visits is finite. Same logic applies that if all states of a Markov chain are ergodic, then the chain itself is ergodic and thus 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 by (**Bremaud1999**):

$$\pi_j = \sum_{i \in I} \pi_i p_{i,j}, \quad j \in I \tag{1.19}$$

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 Equation 1.19 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:

$$\mathbb{P}(X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n) = \mathbb{P}(X_k = i_0, X_{k+1} = i_1, \ldots, X_{k+n} = i_n) \tag{1.20}$$

and also for π_j called initial stationary probabilities:

$$p_j(n) = \mathbb{P}(X_n = j) = \pi_j, \quad j \in I \tag{1.21}$$

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 (**Praskova2012**) 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.

Important property of stationary distribution is that if we have an irreducible and aperiodic Markov Chain, its transition matrix A converges to limiting matrix π_{ij} according to (**Haggstrom2002**) as follows:

$$\lim_{n \to \infty} p_{ij}^{(n)} = \pi_{ij}, \quad i, j \in I \tag{1.22}$$

since the transition matrix is irreducible, each row entry of limiting matrix π_{ij} is equal to the stationary distribution π_j. This is called *Markov Chain convergence theorem*, and it implies that if we run Markov Chain for sufficiently long time, the distribution at time n approaches the stationary distribution π, i.e. approaches equilibrium as $n \to \infty$.

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 (**Navarro2011**) states that it 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 1.2 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 1.3.

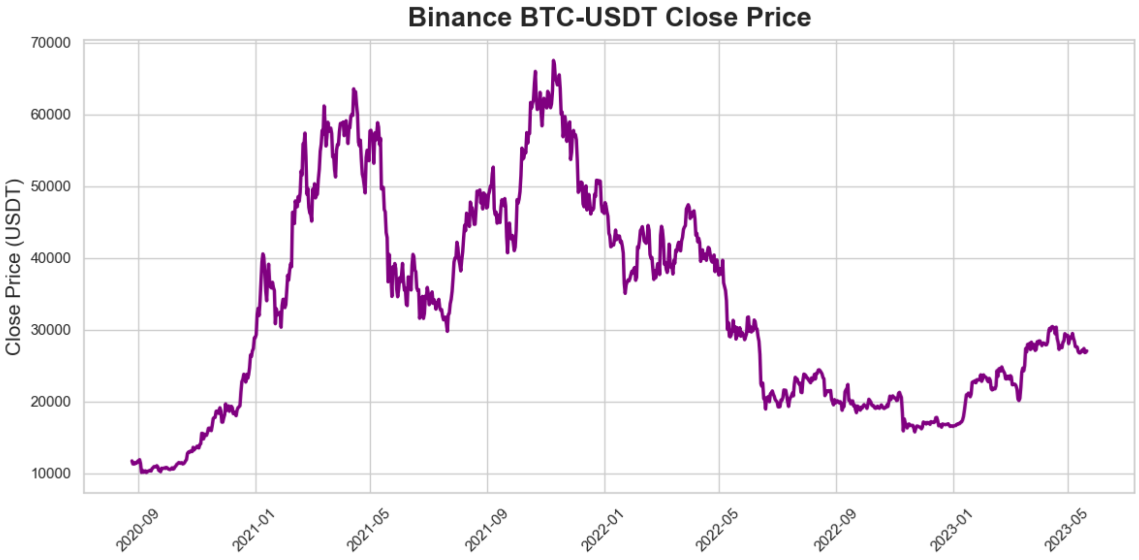


Fig. 1.2 BTC-USD daily close prices from 23rd August 2020 to 15th May 2023 obtained from Binance. (**tradingview**)

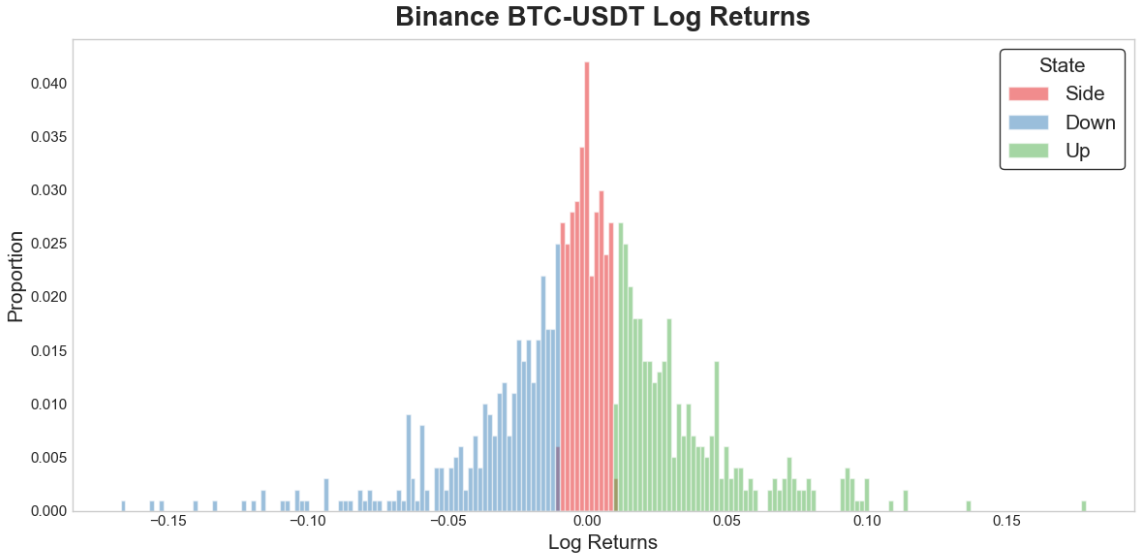


Fig. 1.3 Distribution of the log returns with respect to predefined market states. (**tradingview**)

Observing the distribution of the log-returns of the daily BTC-USDT close price in 1.3 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. (**Peters1994**)

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:

$$\mathbf{A} = \begin{pmatrix} 0.22 & 0.16 & 0.62 \\ 0.32 & 0.27 & 0.41 \\ 0.6 & 0.19 & 0.21 \end{pmatrix}, \quad \mathbf{p} = \begin{pmatrix} 0.4 & 0.19 & 0.41 \end{pmatrix} \tag{1.23}$$

where we note that the frequency of observing each state is proportional to the initial distribution $\mathbf{p}$, and we assume that state space is $I = \{U, S, D\}$ By the definition of the transition matrix $\mathbf{A}$, such matrix is also a stochastic matrix since it satisfies the properties given by Equation (1.5). Each state of the transition matrix $\mathbf{A}$ 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 $\mathbf{p}$ using Equation 1.19. Finally, we may also calculate the expected return time for each state as follows:

$$E[\tau_U(1)|X_0 = U] = \frac{1}{\pi_U} = \frac{1}{0.4} = 2.5 \tag{1.24}$$

$$E[\tau_S(1)|X_0 = S] = \frac{1}{\pi_S} = \frac{1}{0.19} = 5.26 \tag{1.25}$$

$$E[\tau_D(1)|X_0 = D] = \frac{1}{\pi_D} = \frac{1}{0.41} = 2.5 \tag{1.26}$$

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 $\{X(t), t \geq 0\}$ on a probability space $(\Omega, \mathscr{F}, \mathbb{P})$ s.t. for all $t \in \mathbb{R}_0^+$ and $i_0, i_1, \ldots, i_{t+1} \in I$ it holds that:

$$\mathbb{P}(X(t) = j | X(s) = i, X(t_n) = i_n, \ldots, X(t_1) = i_1) = \mathbb{P}(X(t) = j | X(s) = i) \tag{1.27}$$

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 Equation 1.2 with the only difference of continuous transition period. (**Tolver2016**)

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

$$p_{i,j}(s, s+t) = p_{i,j}(t), \quad i, j \in I \tag{1.28}$$

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:

$$p_{i,j}(s+t) = \sum_{k \in I} p_{i,k}(s) p_{k,j}(t), \quad i, j \in I \tag{1.29}$$

Here we also assume (**Gallager2013**) stating following:

$$\lim_{t \to 0_+} p_{i,j}(t) = \delta_{i,j} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases} \tag{1.30}$$

where $\delta_{i,j}$ is a *Kronecker delta function*, i.e. the transition probabilities $p_{ij}(t)$ are right continuous at $t = 0$. (**Norris2012**) says that if such additional conditions are 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 càdlàg almost surely. Such version allows us to infer certain properties of the stochastic process.

For example (**Gidi2018**), one important property of such Markov Chain following from *Doob's martingale convergence theorem* for $s \geq 0$ and $h > 0$ is:

$$\mathbb{P}(X(t) = i | X(s) = i, s \leq t \leq s + h) = \exp(-q_i h) \tag{1.31}$$

which means that the probability of staying in state i for time h is equal to $e^{-q_i h}$. Non-negative real elements of $q_{i,j}$ are called *transition rates* from state i to state j and absolute transition rate $q_i = \sum_{j \neq i} q_{i,j}$ respectively. Let us denote $\mathbf{Q}$ as transition rates matrix with entries $\{q_{i,j} : i, j \in I\}$ where each row sums to zero, i.e. $\sum_{j \in I} q_{i,j} = 0$. This also means that the diagonal entries of $\mathbf{Q}$ are equal to $q_i = -q_{i,i}$ and $\sum_{j \neq i} q_{i,j} = -q_{i,i}$. 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. (**Praskova2012**)

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. (**Norris2012**)

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:

$$T_i = \inf\{t \geq 0 : X(t) \neq i | X(0) = i\} \tag{1.32}$$

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

$$f(x) = \begin{cases} q_i e^{-q_i x}, & x \geq 0 \\ 0, & \text{elsewhere} \end{cases} \tag{1.33}$$

According to (**Praskova2012**) and the properties of the transition rates, such time-homogeneous continuous Markov Chain should satisfy following equations:

$$\begin{aligned}\mathbb{P}(X_{t+h} = i|X_t = i) &= 1 - q_i h + o(h)\\ \mathbb{P}(X_{t+h} = j|X_t = i) &= q_{i,j} h + o(h), \quad i \neq j\end{aligned} \tag{1.34}$$

where $o(h)$ is a function of h such that $\lim_{h\to 0} \frac{o(h)}{h} = 0$.

Such intensities resemble the probability functions of Poisson process, and indeed according to (**Norris2012**) it is a special case of homogeneous continous-time Markov Chain 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:

$$\mathbf{Q} = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \dots \\ 0 & -\lambda & \lambda & 0 & \dots \\ 0 & 0 & -\lambda & \lambda & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{1.35}$$

1.2.1 Cryptocurrency market movements II.

1.3 Markov Renewal process

Poisson process as a counting process serves as model to represent the number of events occurring in a given time interval $[0,t]$. The inter-arrival times between events are exponentially distributed iid random variables with parameter $\lambda > 0$ and the number of events in all disjoint intervals is independent. In other words, the probability of observing n events in interval $[0,t]$ is given by Poisson distribution with parameter λt. But we would like to also study instances where the inter-arrival times are not exponentially distributed, but rather follow some other distribution.

Therefore, we introduce a concept of *renewal process*, as stated by (**Praskova2012**) and (**Mitov2014**), which can be defined using a sequence of non-negative random variables $\{X_n, n \in \mathbb{N}_0\}$ that are identically distributed and independent with distribution function $F(0) < 1$ that satisfies following equation:

$$S_n = \sum_{k=0}^{n} X_k, \quad n \in \mathbb{N}_0 \tag{1.36}$$

where S_n is a random variable that denotes the inter-arrival time between the n-th $n-1$-th renewal such that the renewal process is then:

$$N(t) = \sup\{n : S_n \leq t\} = \sum_{n=1}^{\infty} \mathbb{1}_{\{S_n \leq t\}} \tag{1.37}$$

where $N(t)$ is a random variable that denotes the number of events in interval $[0,t]$. Furthermore, we could also note that the renewal function is derived as the expectation $m(t) = \mathbb{E}[N_t]$.

Our interest lies in the special case of the renewal process called *Markov renewal process* which is defined as a renewal process with the additional property that the sequence of random variables $\{X_n, n \in \mathbb{N}_0\}$ denoting states still constitutes a Markov Chain, in this case discrete-time Markov Chain, and the transition intervals $\tau_n = T_n - T_{n-1}$ follow any distribution with finite mean and its parameter may depend on the previous state X_{n-1} and also current state X_n of the Markov Chain. This process is also known as *semi-Markov* since the transitions between states are still occurring according the Markov Chain, but the time between transitions is a random variable with some distribution that again may depend on the previous and current state of such Markov Chain. Thus, semi-Markov process serves as a way of describing the underlying Markov renewal process. (**Medhi2012**)

Take a process $\{X_n, n \in \mathbb{N}_0\}$ defined on state space I and let the transitions occur at certain epochs $t_0 = 0, \dots, t_n, t_{n+1}$. If for such process Markov memoryless property holds:

$$\mathbb{P}(X_{n+1} = j, \tau_{n+1} \leq t | X_n = i, \dots, X_0 = i_0, t_n, \dots, t_0) = \mathbb{P}(X_{n+1} = j, \tau_{n+1} \leq t | X_n = i) \quad (1.38)$$

then the $\{X_n, \tau_n\}$ for $n \in \mathbb{N}_0$ constitute a *Markov renewal process*, as stated by (**Cinlar1969**) and (**Barbu2008**), and the process is clearly time-homogeneous since the transition probabilities are independent of time:

$$\mathbb{P}(X_{n+1} = j, \tau_{n+1} \leq t | X_n = i) = \mathbf{Q}_{i,j}(t) \quad (1.39)$$

where $\mathbf{Q}_{i,j}(t)$ is, according to (**Medhi2012**), called a *semi-Markov kernel* of the Markov renewal process. Limiting behavior of the kernel $\mathbf{Q}_{i,j}(t)$ as $t \to \infty$ is approaching the transition matrix $\mathbf{A}$ of the Markov Chain since the distribution function of waiting times is equal to 1 for $t \to \infty$:

$$\lim_{t \to \infty} \mathbf{Q}_{i,j}(t) = \mathbf{A}_{i,j} \quad (1.40)$$

Moreover, for continuous process $\{Y(t)\}$ defined on a state space I, we say that it is called *semi-Markov process* if:

$$Y(t) = X_n, \quad t \in [t_n, t_{n+1}) \quad (1.41)$$

where X_n is an embedded Markov Chain of such continuous process. Clearly, we see that at each epoch t_n the process $\{Y(t)\}$ jumps to a new state and stays in such state until the next epoch t_{n+1} as is visible on Figure 1.4. Therefore, the process $\{Y(t)\}$ is a semi-Markov process with semi-Markov kernel $\mathbf{Q}_{i,j}(t)$. From the definition of the continuous-time Markov Chain, it follows that the semi-Markov process with exponential holding times is a continuous-time Markov Chain (**Sahner1996**):

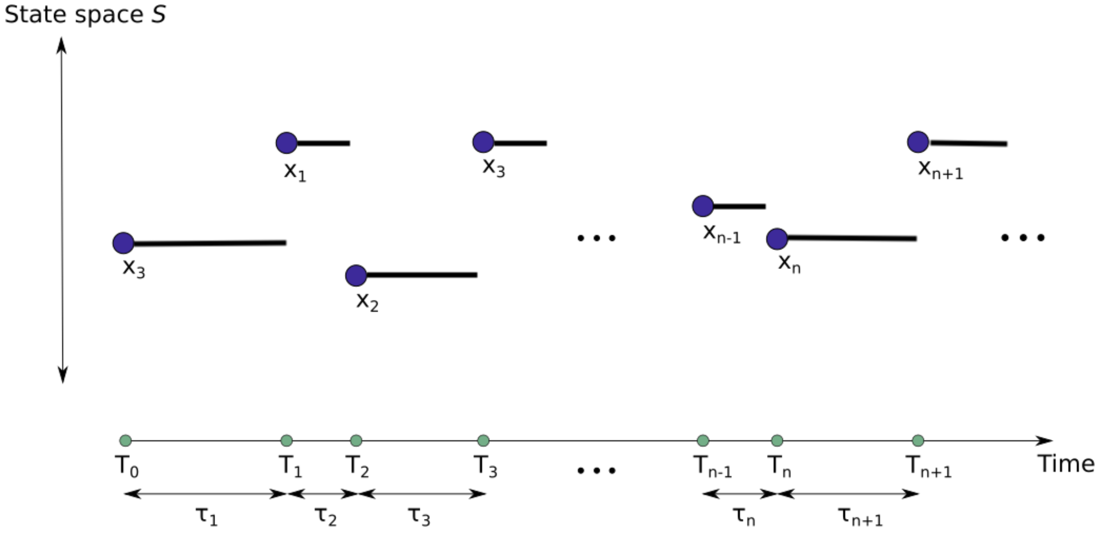
$$\begin{aligned}\mathbb{P}(X_{n+1} = j, \tau_{n+1} \leq t | X_n = i) &= \mathbb{P}(X_{n+1} = j | X_n = i)\mathbb{P}(\tau_{n+1} \leq t | X_{n+1} = j, X_n = i) && (1.42)\\ &= \mathbb{P}(X_{n+1} = j | X_n = i)\mathbf{W}_{i,j}(t) && (1.43)\\ \mathbf{Q}_{i,j}(t) &= a_{i,j}(1 - e^{-\lambda_i t}) && (1.44)\end{aligned}$$

where $\mathbf{W}_{i,j}(t)$ is a probability that the process $\{Y(t)\}$ stays in state j for time t given that it is in state i, i.e. waiting time. In other words, it is a conditional distribution function of the random variable τ conditioned on the events $X_{n+1} = j, X_n = i$. Such cumulative distribution function is again a random variable we call *conditional sojourn time* and denote with $T_{i,j}$:

$$\mathbf{W}_{i,j}(t) = \mathbb{P}(T_{i,j} \leq t) = \mathbb{P}(\tau_{n+1} \leq t | X_{n+1} = j, X_n = i) \tag{1.45}$$

we can also find the relation between waiting times and transition probabilities as follows from Equation 1.46 which holds for all $a_{i,j} > 0$ s.t. if transition probability to state j is zero, i.e. $a_{i,j} = 0$, then the waiting time is assumed to be 1:

$$\mathbf{W}_{i,j}(t) = \frac{\mathbf{Q}_{i,j}(t)}{a_{i,j}} \tag{1.46}$$



Source: *Wikipedia page on Markov renewal process*

Fig. 1.4 Graphical representation of a semi-Markov process Y(t).

Thus, we can observe that the there is a valuable benefit of using more generalized Markov Chain such as semi-Markov process since it allows us to model the waiting times between transitions as a random variable with some distribution. In other words, we can model the time between transitions as a random variable with some distribution, which is not possible with the standard (discrete-time) Markov Chain that assumes that transitions occur at each time step and the time spent in each state (or inter-arrival time) is geometrically, or in continuous case exponentially, distributed.

This property is especially useful in modeling the cryptocurrency market since the time between transitions is not constant, but rather follows some distribution. We can expect that if the market is in bull or bear state it will stay in such state for some time, but the time spent in such state could be very different from stagnant market state.

Chapter 2

Hidden Markov Models

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. These "visible" observations are labeled as emissions emitted by the respective hidden state. Thus, the observations are assumed to be generated by certain hidden stochastic process, i.e. a Markov Chain.[1] In general, we could assume that the state and emission space is not necessarily finite, but we will follow the classical definition by (**Rabiner1989**) and (**Elliott1995**) of Hidden Markov Model (hereinafter "HMM") and assume that at least state space is finite.

2.1 Discrete Hidden Markov Model

A Discrete Hidden Markov Model is one where the observations are assumed to be a sequence of discrete symbols or categorical variables. It defines a family of models that are commonly used in statistical signal processing, speech recognition, computational biology, and other fields. (**Rabiner1989**)

There are two main models under the umbrella of Discrete Hidden Markov Models, namely *Categorical Hidden Markov Model* and *Multinomial Hidden Markov Model*. The difference between them is in the way we are interpreting the emissions. In case of Categorical HMM, we assume that the emissions are categorical variables, i.e. they are represented by a single integer value, meaning that each emission belongs to one of the M possible categories. On the other hand, in case of Multinomial HMM, we assume that the emissions at each time step are a result of a multinomial experiment, i.e. certain number of trials with M possible

[1] Models that comprise unobserved random variables, e.g. Hidden Markov Models, are called **latent variable models**, **missing data models**, or also **models with incomplete data**

outcomes. In other words, each emission is represented by a vector of length M with only one element equal to 1 and the rest 0 since the number of trials is fixed to 1. Such vector definition would change if we had more than one trial in single time step, but we will not consider such case in this work since it is not applicable to the problem at hand. We always assume that the number of trials is fixed to 1 and thus each observation might be represented by a vector of length M with only one element equal to 1 and the rest 0 or by a single integer value for the purpose of Categorical HMM. This vector representation is also called *1-of-M encoding*, but we will use the representation of integer values since it is easier to interpret and more efficient in our case.

Revisiting previous section, we have defined a transition matrix **A** and initial distribution **p** for a homogeneous Markov Chain. Both of these parameters are used to describe the hidden stochastic process. In order to describe the emissions, we will define an $N \times M$ *emission matrix*[2] **B** as follows:

$$\mathbf{B} = \begin{pmatrix} b_1(y_1) & b_1(y_2) & \dots & b_1(y_M) \\ b_2(y_1) & b_2(y_2) & \dots & b_2(y_M) \\ \vdots & \vdots & \ddots & \vdots \\ b_N(y_1) & b_N(y_2) & \dots & b_N(y_M) \end{pmatrix} \tag{2.1}$$

where N represents number of states and M number of possible emission symbols.

Probabilistically each element of the matrix represents conditional probability of emitting symbol k for all $k = 1, 2, \dots, M$, given state i:

$$b_i(k) = \mathbb{P}(Y_t = k | X_t = i) \tag{2.2}$$

As stated at the beginning of this section, observer does not have access to the hidden states, but only to the emissions emitted by the hidden states. Thus, observing only another stochastic process $\{Y_t, t \in \mathbb{N}_0\}$ linked to hidden Markov Chain s.t. it governs the distribution of Y_t. In other words, entire statistical inference, even in terms of the hidden Markov process, is based on the observed sequence of emissions $\{Y_t\}$.

Considering the discrete time index t, Hidden Markov model is bivariate discrete-time stochastic process $\{X_t, Y_t\}$, where $\{Y_t\}$ is a sequence of conditionally independent and identically distributed random variables with a probability distribution determined by the hidden state i at time t. (**Rabiner1989**)

Note that there are three common types of Hidden Markov Models depending on structure of the underlying hidden Markov Chain according to (**Nelwamondo2006**). These are namely

[2] Also referred to as **observation matrix** and **observations** respectively

(a) left-to-right, (b) two-parallel left-to-right and (c) ergodic. First model bears a property that the next state index is always greater than or equal to the current state index s.t. the final state is absorbing. Two-parallel left-to-right model allows for two parallel paths taken by the Markov Chain and lastly in ergodic model, all states are connected. For the purpose of this work we will only consider models with ergodic property as shown in Figure 2.2.

Above we specified an emission matrix **B** as a discrete probability distribution of the emissions, given the hidden state, that take on values from a finite set of M possible emissions. $\{Y_t\}$ is then a sequence of conditionally independent and identically distributed discrete random variables that follow a categorical distribution with M possible outcomes, i.e. each emission symbol k is given a probability of being emitted by the hidden state i as stated by (**Paisley2009**):

$$Y|X = i \sim \mathrm{Cat}(M; b_i(y_1), \ldots, b_i(y_M)), \quad \forall i \in I \tag{2.3}$$

$$f(y|x = i) = \prod_{k=1}^{M} b_i(k)^{\mathbb{1}\{y=k\}} \tag{2.4}$$

Important property of the categorical distribution in Bayesian statistics is that it is a generalization of the Bernoulli distribution for $M > 2$ outcomes and also a conjugate prior for the Dirichlet distribution. In some instances, as expressed by (**Paisley2009**), we also use Dirichlet distribution with parameters $\alpha_1 = \ldots = \alpha_M$ as it is a special case called uniform prior for the categorical distribution. The same approach is also applicable other parameters of the model, i.e. transition matrix **A** and initial distribution **p**.

Usually, as mentioned in (**Agresti2007**), we generalize the categorical distribution to a multinomial distribution in which case the number of samples is fixed to 1. In other words, the emission space is not 1-to-M encoded but rather 1-of-M encoded s.t. the probability mass function is:

$$Y|X = i \sim \mathrm{Multinomial}(1; M; b_i(y_1), \ldots, b_i(y_M)), \quad \forall i \in I \tag{2.5}$$

$$f(\mathbf{y}, n = 1|x = i) = \prod_{k=1}^{M} b_i(k)^{y_k}, \quad \sum_{k=1}^{M} y_k = 1 \tag{2.6}$$

where **y** is a vector of length M with only one element equal to 1 and the rest 0.

HMM with finite state and emission space as described above is also called *discrete Hidden Markov Model*. Same definition is also applicable for continuous emission space, where the probability density function is often a Gaussian distribution with mean vector μ and covariance matrix Σ or Poisson distribution with parameter vector λ.

An important note, as per (**Bishop2006**) and (**Ramaprasad2004**), is that the emissions are not necessarily independent, but only conditionally independent given the hidden state, which implies that the process $\{Y_t\}$ is not a Markov Chain, as opposed to process $\{X_t\}$, but we may view it as an extension of Mixture Models with certain dependencies between the emissions. For example, in a case where we have a Markov Chain with finite N hidden states and emissions modelled by Gaussian distributions, we refer to such model as *Gaussian Hidden Markov Model*[3] s.t. marginal distribution of Y_t is a mixture of Gaussian distributions.

Since HMM is categorized as a sequence model, we are generally interested in a joint probability distribution of the hidden states and the emissions when examining time series data or any other sequential data. Such joint probability distribution given parameters vector $\theta = (\mathbf{A}, \mathbf{B}, \mathbf{p})$ is, as described by (**Rabiner1989**), expressed by the following equation:

$$\begin{aligned}\mathbb{P}(\mathbf{X}, \mathbf{Y}|\theta) =& \mathbb{P}(X_0 = x_0)\mathbb{P}(Y_0 = y_0|X_0 = x_0) \\ & \prod_{t=1}^{T} \mathbb{P}(X_t = x_t|X_{t-1} = x_{t-1})\mathbb{P}(Y_t = y_t|X_t = x_t) \qquad (2.7) \\ =& p_{x_0} b_{x_0}(y_0) \prod_{t=1}^{T} a_{x_t, x_{t-1}} b_{x_t}(y_t) \qquad (2.8)\end{aligned}$$

where $\mathbf{X} = \{x_0, x_1, \ldots, x_T\}$ and $\mathbf{Y} = \{y_0, y_1, \ldots, y_T\}$ are the sequences of hidden states and emissions respectively.

One possibility of graphically representing such joint probability distribution is by using a graphical model as shown in Figure 2.1.

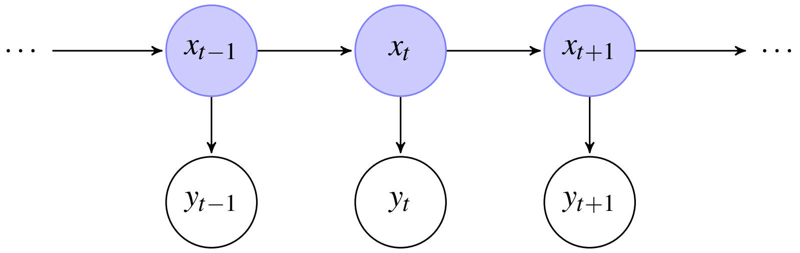


Fig. 2.1 An HMM with 4 hidden states and 2 discrete emissions denoted by x_1 and x_2.

We may also visualize conditional relationship with a graphical model in Figure 2.2 representing a structure of HMM with 3 hidden states $\{S_1, S_2, S_3\}$ and 2 emission symbols

[3] In some literature as (**Capp2005**) also **normal Hidden Markov Model**

$\{E_1, E_2\}$ where the nodes represent the relationship imposed by the transition matrix **A** and emission matrix **B**:

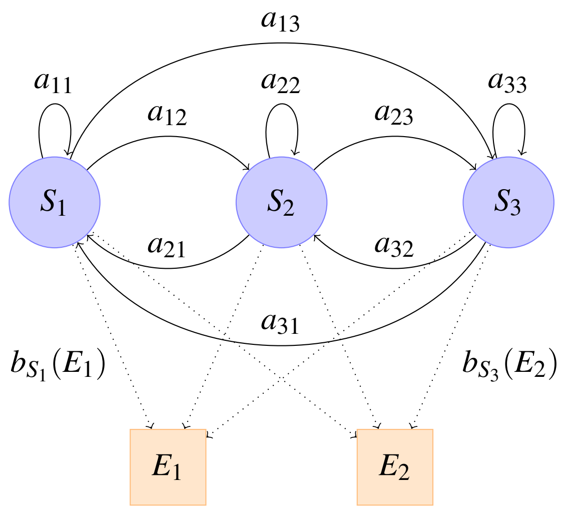


Fig. 2.2 An HMM with 3 hidden states and 2 emission symbols denoted by E_1 and E_2.

There are 3 main assumptions of Hidden Markov Models as a consequence of the properties of Markov processes, as suggested by (**Oliver2013**) and (**Capp2005**):

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

$$\mathbb{P}(X_{t+1} = j | X_t = i, X_{t-1} = i_{t-1}, \dots, X_0 = i_0) = \mathbb{P}(X_{t+1} = j | X_t = i) \quad \forall i, j \in I \tag{2.9}$$

   It is also possible to assume that the states in HMM are dependent beyond the current state therefore giving rise to *k-order HMM* where the conditional distribution of the next state depends on the current state and the previous k states $X_{t-k}, X_{t-k+1}, \dots, X_{t-1}$. (**Capp2005**)

2) **Stationary assumption** - The transition matrix **A** is time invariant s.t. transition probabilities only depend on the time interval between the transitions, thus for all $t \neq s$:

$$\mathbb{P}(X_{t+1} = j | X_t = i) = \mathbb{P}(X_{s+1} = j | X_s = i) \quad \forall i, j \in I \tag{2.10}$$

However, when we abstract from this assumption, we may consider a *non-stationary HMM* where the transition matrix is time-dependent, i.e. depends on time index t.

3) **Emission independence assumption** - Emissions are independent, s.t. if we have an emission sequence $\mathbf{Y} = \{y_1, y_2, \dots, y_T\}$ then:

$$\mathbb{P}(\mathbf{Y}|X_1 = x_1, \dots, X_T = x_T) = \prod_{t=1}^{T} \mathbb{P}(Y_t = y_t | X_t = x_t) \tag{2.11}$$

When there is a dependence between the emissions in a way that the emission at time t depends on the previous emissions, conditionally on the hidden state sequence $\{X_k\}$, $\{Y_k\}$ forms a (non-homogeneous) Markov Chain, therefore jointly representing a generalization of HMM called *Markov-switching Models*[4]. (**Capp2005**)

There are mainly 3 fundamental problems in HMM that need to be resolved as depicted by (**Oliver2013**) and (**Ching2005**):

1. **Evaluation problem** - Given a model denoted as $\theta = (\mathbf{A}, \mathbf{B}, \mathbf{p})$ and an emission sequence $\mathbf{Y} = y_1, y_2, \dots, y_T$, how to efficiently compute the probability that the model generated the observation sequence, in other words, what is $\mathbb{P}(\mathbf{Y}|\theta)$?

2. **Decoding problem** - What is the most likely sequence of hidden states that could have generated the emission sequence $\mathbf{Y}$? Thus, we would like to find $\mathbf{X} = \arg\max_{\mathbf{X}} \mathbb{P}(\mathbf{Y}, \mathbf{X}|\theta)$, where $\mathbf{X}$ is the hidden state sequence.

3. **Learning problem** - Given a set of emission sequences find θ that best explains the emission sequence $\mathbf{Y}$. Thus, find the vector of parameters θ that maximizes $\mathbb{P}(\mathbf{Y}|\theta)$.

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 solved by **Forward-Backward algorithm**[5], 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. All three algorithms are described in the following chapter.

2.2 Continuous HMM

If we were to consider a continuous emission space, we could plot the joint probability distribution of the hidden states and the emissions to see that the marginal distribution of the

[4] Also, **Markov jump system**.

[5] Focusing on the most widely used implementation called **alpha-beta algorithm**.

emissions is not that easily tractable and may not necessarily be unimodal. For example, in *Gaussian Hidden Markov Model* (GHMM) setting, marginal distribution of the emissions is a mixture of Gaussian distributions that are either univariate or multivariate depending on the feature space. (**Bishop2006**)

Although, most of the time we consider unimodal distributions for our data it might not be the case for incomplete data, i.e. data with missing values. For example, statistical inferences about the subpopulations within the overall population require such tools in cases where the subpopulations significantly differ and may be interpreted the same way as hidden variables in case of Hidden Markov Models. With the extension of Mixture Models we could possibly consider any distribution as a component of the mixture, but since Gaussian distributions are able to approximate any density arbitrarily well, we will focus on them in the following section.

2.2.1 Mixture Distributions

Let $\mathbf{Y}$ denote a p-dimensional random vector with probability density function $f_{\mathbf{Y}}(y_1,\ldots,y_p)$ on $\mathbb{R}^p$. This probability density function is defined as a convex combination of M component probability densities as follows from (**McLachlan2000**):

$$f_{\mathbf{Y}}(y_1,\ldots,y_p) = \sum_{i=1}^{M} \pi_i f_i(y_1,\ldots,y_p) \tag{2.12}$$

where f_i is a component density of the mixture and π_i a mixing proportion or mixture weight with following properties:

$$0 \leq \pi_i \leq 1 \quad \forall i \in \{1,2,\ldots,M\} \tag{2.13}$$

and

$$\sum_{i=1}^{M} \pi_i = 1 \tag{2.14}$$

Therefore, the probability density $f_{\mathbf{Y}}$ given by Equation 2.12 is referred to as a g-component finite mixture density, conversely $F_{\mathbf{Y}}$ as a g-component finite mixture distribution.

Equation 2.12 also assumes that the number of components M is fixed but in many applications this is not the case, and we have to infer the number of components which are required to adequately describe the data. One way we can avoid this problem is by assuming that the number of components is sufficiently large or infinite, and then use a model selection criterion to select the number of components that best fits the data as suggested by (**Sammut2011**) and (**Rasmussen1999**). Furthermore, mixing proportions π_i are also unknown and have to be estimated along with the respective parameters of the component densities.

Analogously to previous definition we can define a finite mixture of random vector **Y** as (**Bishop2006**):

$$f(y,x) = g(x)f(y|x) \tag{2.15}$$

where X is defined as a random variable following a multinomial distribution with a vector of parameters $\pi = \{\pi_1, \dots, \pi_M\}$ and $g(x)$ as its probability density function. Summing over all possible values of X we obtain the marginal density of random vector **Y**:

$$f_{\mathbf{Y}}(y_1,\dots,y_p) = \sum_{i=1}^{M} \mathbb{P}(X=i) f_{\mathbf{Y}}(y_1,\dots,y_p|X=i) \tag{2.16}$$

where $\mathbb{P}(X=i) = \pi_i$ and $f_{\mathbf{Y}}(y_1,\dots,y_p|X=i) = f_i(y_1,\dots,y_p)$ are the mixing proportions and component densities respectively thus arriving at the same definition as in Equation 2.12.

2.2.2 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 (**Bishop2006**):

$$f(y|\mu,\Sigma) = \sum_{k=1}^{K} \pi_k \mathscr{N}(y|\mu_k,\Sigma_k) \tag{2.17}$$

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

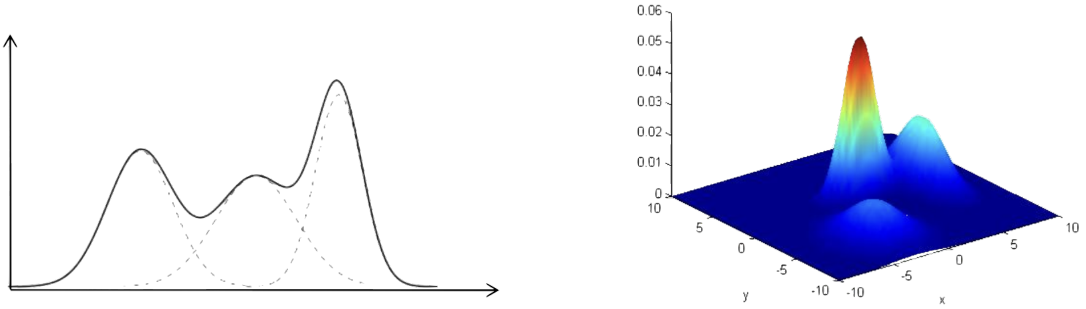
Thus, the vector of parameters is $\theta = \{k \in \mathbb{N} : (\pi_k, \mu_k, \Sigma_k)\}$. Probability density function of each component of GMM depends on the dimensionality of the data. Let the dimensionality of the data be $d > 1$ then the random vector $\mathbf{Y} = (y_1, \dots, y_d)$ has the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

$$f_k(\mathbf{Y}) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{Y}-\mu_k)^T \Sigma_k^{-1}(\mathbf{Y}-\mu_k)\right) \tag{2.18}$$

where $|\Sigma_k|$ denotes the determinant of d x d covariance matrix Σ_k and μ_k is a d-dimensional mean vector. Therefore, the conditional distribution of Y given component k is:

$$\mathbf{Y}|k \sim \mathcal{N}_d(\mu_k, \Sigma_k) \tag{2.19}$$

If we consider $d = 1$ we say that the random variable Y has a univariate Gaussian distribution with mean μ_k and variance σ_k^2 and there the mixture model is called a *univariate normal (Gaussian) mixture model* as in Fig 2.3a with number of components $M = 3$. In case of $d > 1$ we refer to the model as *multivariate normal (Gaussian) mixture model*, e.q. in Fig 2.3b where $d = 2$ and $M = 3$.



(a) d = 1

(b) d = 2

Source: *A short tutorial on Gaussian Mixture Models by Mohand Saïd Allili*

Fig. 2.3 Probability distribution of Gaussian Mixture Model with $M = 3$ components for $d = 1$ and $d = 2$ respectively.

The motivation for using the Gaussian mixture model is that it is a universal approximation of densities, i.e. it can approximate any density arbitrarily well as stated by (**Bishop2006**). In a setting of possible stock price prediction we may consider looking at asset returns as a mixture of Gaussian distributions, where each component represents a different regime of the market. This was first considered by (**Fama1965**) in estimating densities of asset returns and later further developed by other authors. Indeed, the financial markets were shown to be characterized by different regimes that manifest themselves in the implication on the mean and variance of the returns. (**Hamilton1989**)

We aim to briefly explain only one regime that deviates from the "usual" view of the asset returns distribution, which was first introduced by (**Black1976**) and later by (**Christie1982**). This regime or effect is called *leverage effect*, and it is defined as a negative correlation between the returns and the volatility of the asset. Such effect is often observed in the financial markets and is usually attributed to the fact that the expected asset returns tend to increase during pronounced market downturns, as well as during periods of higher volatility. (**Aydemir2007**)

Such behavior might be captured by a 2-component mixture of Gaussian distributions, where the first component represents the "usual" regime of the market and the second component represents the "leverage" regime. The former component often exhibits near zero mean, while the latter is accompanied by negative mean and much higher variance and correlations to address the existence of the asymmetry and leptokurtosis in the distribution of asset returns. (**Paolella2015**)

This is providing us with a motivation to consider a Gaussian mixture model as a generative model for the asset returns and volatility which we will further use in the context of Hidden Markov Models in prediction of the market states.

As for the estimation of the parameters of the Gaussian mixture model, we will use the Expectation-maximization (EM) algorithm which is a general approach to maximum likelihood estimation in the presence of missing or hidden data. The details of the algorithm are described in the following chapter focused on the parameter estimation. (**Dempster1977**)

2.2.3 Gaussian HMM

As we have shown in the previous section, Hidden Markov Models are a generalization of Mixture Models with dependencies between the hidden states. In section dedicated to *Discrete HMM* we have shown that the marginal distribution of the emissions given state is categorical distribution. Since a suitable choice for the emission distribution is a Gaussian distribution, we may distinguish between two types of Gaussian Hidden Markov Models, namely *Gaussian-Mixture HMM* and *Gaussian HMM*. The former is a generalization of the latter, where the marginal distribution of the emissions is a mixture of Gaussian distributions. (**Bishop2006**)

In case of Gaussian HMM, the marginal distribution of the emissions is a d-dimensional Gaussian distribution with $N \times d$ mean vector μ and vector Σ of length N of $d \times d$ covariance matrices, thus the number of components $M = 1$. Depending on the dimensionality of the data, we may consider univariate or multivariate Gaussian distribution. The probability density function of the d-dimensional Gaussian distribution for each hidden state i is defined as follows:

$$b_i(y_t) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(y_t - \mu_i)^T \Sigma_i^{-1} (y_t - \mu_i)\right) \tag{2.20}$$

where y_t is a d-dimensional vector of emissions at time t. Parameters of such HMM model are therefore $\theta = \{\mathbf{A}, \mathbf{p}, \mu, \Sigma\}$.

Taking into account $M > 1$ components, we arrive at the Gaussian-mixture HMM, where the marginal distribution of the emissions is a mixture of Gaussian distributions. The probability density function of the d-dimensional Gaussian mixture distribution for each hidden state i is defined as follows:

$$b_i(y_t) = \sum_{k=1}^{M} \frac{\pi_{i,m}}{(2\pi)^{d/2}|\Sigma_{i,k}|^{1/2}} \exp\left(-\frac{1}{2}(y_t - \mu_{i,k})^T \Sigma_{i,k}^{-1}(y_t - \mu_{i,k})\right) \tag{2.21}$$

where $\pi_{i,k}$ is the mixing proportion of the k-th component of the mixture and $\mu_{i,k} \in \mathbb{R}^d$ and $\Sigma_{i,k} \in \mathbb{R}^{d\times d}$ are the mean and covariance matrix of the k-th component respectively. Suddenly the number of parameters increases linearly with the number of components M and thus the model becomes more complex. Parameters of such HMM model are therefore $\theta = \{\mathbf{A}, \mathbf{p}, \pi, \mu_{i,k}, \Sigma_{i,k}\}$ for $i = 1,\dots,N$ and $k = 1,\dots,M$.

One convenient way to represent this model, according to (**Yu2015**), is to view it as a generative model producing a sequence of emissions $\mathbf{Y} = \{y_1,\dots,y_T\}$ as follows:

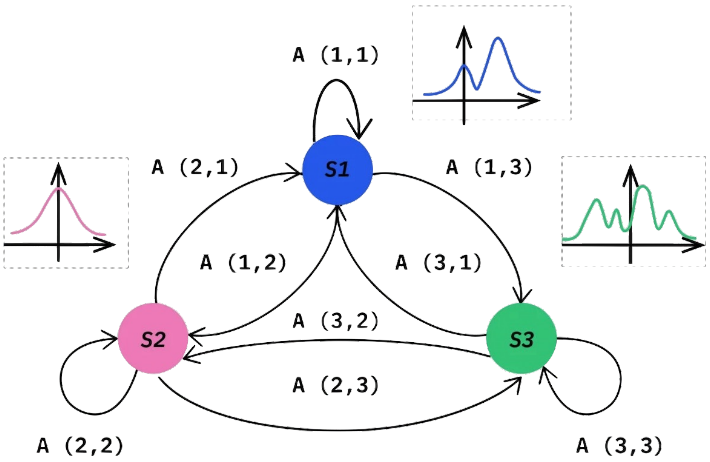
$$y_t = \mu_i + r_t(\Sigma_i) \tag{2.22}$$

where the state i is determined by the transitions in the Markov Chain and r_t is a random d-dimensional vector drawn from the Gaussian distribution with zero mean and covariance matrix Σ_i. Below you can see a graphical representation of such models:

2.3 Hidden Semi-Markov Model

One of the limitations mentioned in the chapter dedicated to Markov Renewal Processes is the fact that the time spent in a state is geometrically distributed which is not always the case in real-world applications. In order to address this issue, we may consider a generalization of the Hidden Markov Model called *Hidden Semi-Markov Model* (HSMM) where the time spent in a state follows a general distribution, e.g. Poisson, Negative binomial. (**Yu2015**)

In classical HMM setting, each observation at time t is generated by the hidden state X_t. After emitting the observation, the model transitions to a new state X_{t+1} according to the transition probabilities. In case of HSMM, the model stays in the same state i for a random number of time steps before transitioning to state j, which is called waiting time and is denoted by $W_{i,j}t$. This means that the original observation sequence $\mathbf{Y} = \{y_1,\dots,y_T\}$ now consists of multiple subsequences $\mathbf{C}_1,\dots,\mathbf{C}_L$ where each element in subsequence $\mathbf{C}_l$ is generated by the same hidden state. (**Dasu2011**) The number of subsequences L as well as



Source: *Improving Hidden Markov Models – Tooploox at NeurIPS 2022*

Fig. 2.4 Graphical representation of Gaussian-Mixture HMM. Each hidden state S1, S2 and S3 is associated with a Gaussian mixture distribution with M components. A(i, j) denotes the transition probability from state i to state j.

the length of each subsequence C_l is governed by the waiting time distribution, which is why this model is sometimes referred to as *Explicit-duration HMM*. (**Yu2015**)

This is an important distinction from the classical model assumption of the HMM where each state can only generate one observation per time step. Thus, the number of observations produced by the state i is determined by the time spent in such state, i.e. duration d. This relationship is depicted in Figure 2.5 where we clearly see that e.g. the number of observations generated by the state x_{l-1} is equal to the duration $d = 3$.

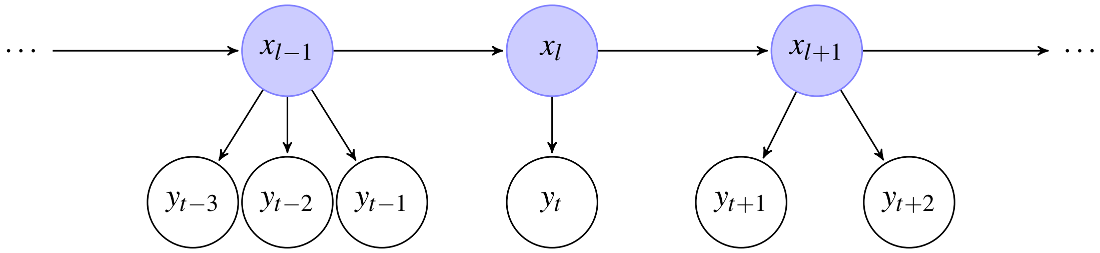


Fig. 2.5 HSMM with left-to-right state and emission dependence. The number of observations generated by the state x_l is equal to the duration d. State $x_{l-1} \neq x_l$ and also $x_l \neq x_{l+1}$.

It is obvious that the only difference between the classical discrete HMM and HSMM is the way the observations are generated. In case of HSMM, the observations are generated by the same state for a random number of time steps and therefore the transition probabilities given by matrix **A** ought to be adjusted accordingly, but emission probabilities remain the same. In order to account for the duration of the chain in each state, we will introduce, as suggested by e.g. (**Bulla2013**), probability distribution called *sojourn time distribution* $d_j(u)$ which is defined as follows:

$$d_i(u) = \mathbb{P}(X_{t+u+1} \neq i, X_{t+u} = i, \dots, X_{t+2} = i | X_{t+1} = i, X_t \neq i) \tag{2.23}$$

where the sojourn time distribution $d_i(u)$ is the probability of staying in state i for u time steps and then transitioning to another state. There is a possibility of using discrete or continuous valued random variable for duration but in case of HSMM we assume that the sojourn time distribution $d_i(u)$ is discrete valued, i.e. $u = \{1, 2, 3, \dots, D\}$ where D is the maximum possible duration in respective state, as stated in (**Yu2010**). The most popular choices of the sojourn time distribution are Poisson, negative binomial or geometric distribution depending on the application. As for the transition probabilities, we are interested in the probability of transitioning from state i to state j but only if $i \neq j$ since the probability of staying in the same state is not governed by the geometric distribution anymore. (**Abdullah2022**) Hence, the transition probabilities are defined as follows:

$$a_{i,j} = \mathbb{P}(X_{t+1} = j | X_{t+1} \neq i, X_t = i) \tag{2.24}$$

where $p_{i,i} = 0$, thus the transition matrix **A** has its diagonal elements equal to zero and is valid stochastic or Markov matrix since $\sum_{i \neq j} a_{i,j} = 1$.

2.4 Models with context

In past years, different versions of Hidden Markov Models have proven to be a powerful tool in modelling short-term dependencies between adjacent symbols. To account for the variability in the data and thus improve the performance of the model, we use standard techniques such as increasing the number of hidden states or increase number of components in Gaussian Mixtures. However, these techniques are not always sufficient to capture the dependencies in the data and lead to overfitting. (**Yoon2006**) However, they are not able to capture for example long-term dependencies or exogenous variables, which are common problems in many applications. (**Yoon2006**)?

In order to address these issues, there exist several extensions of the classical Hidden Markov Model, in some literature under the term *Conditional HMM*. Although, such models might not necessarily satisfy the Markov property, they are still referred to as Hidden Markov Models. The variations of such models arise in the way the transition and emission probabilities are defined. These are often conditioned on the previous k states or emissions, on the entire sequence of states or emissions or even on certain external variables. There is a wide range of extensions, but this work will aim to mainly focus on the subset of them that are relevant to the problem at hand.

2.4.1 Parametric Hidden Markov Model

As we have mentioned the condition of the external variables, models suitable for such a case are called *Parametric Hidden Markov Models* (PHMM) as in (**Bobick1999**) and later in (**Radenen2014**). The name is trivially derived from the parametric dependence of the transition or emission probabilities on the external variables. These variables are beneficial in cases in which we have some additional information about the data, e.g. in case of stock price prediction we may consider the macroeconomic variables such as interest rates, inflation, unemployment rate, etc. This implies that considering different countries might not be a problem since we may use the same model and only change the external variables to account for the differences in respective economics.

In its simplest form the PHMM constructs a dependence between the mean of the Gaussian distribution and the external variables. The usual choice of Gaussian distribution is due to its ability to approximate any distribution by considering the Gaussian mixture model. (**Bishop2006**)

The dependence between the mean of the Gaussian distribution and the external variables ϕ is usually modelled by a linear transformation, but it is also possible to consider non-linear transformations. In case of linear transformation, the mean of the Gaussian distribution is defined as follows:

$$\hat{\mu}^i(\theta) = \mathbf{V}^i\theta + \bar{\mu}^i \tag{2.25}$$

As (**Radenen2014**) points out, $\bar{\mu}^i$ is a $d \times 1$ vector that may be interpreted as an average mean vector of the i-th state modified by linear transformation of the c-dimensional augmented vector of external variables θ. The augmentation of the vector of external variables is done by adding a constant 1 to the vector θ, i.e. assuming we have $c-1$ external variables so that $\theta = [1, \phi]$. The matrix $\mathbf{V}^i$ is a d x c matrix of parameters that defines the linear transformation of the external variables. (**Radenen2014**) The dimensionality of the mean vector μ^i is equal to the dimensionality of the data d s.t. $d = 1$ in case of univariate Gaussian distribution and $d > 1$ in case of multivariate Gaussian distribution.

We are not limited to consider only the parametrization of the mean of the Gaussian distribution, but also the covariance matrix. Works by (**Bobick1999**) and (**Radenen2014**) define diagonal and full covariance matrix parametrization respectively. Let us focus on the latter where each element of the estimated covariance matrix a linear transformation of the external variables:

$$\hat{\Sigma}^i_{u,v} = D^i_{u,u}(\theta) \times D^i_{v,v}(\theta) \times \bar{\Sigma}^i_{u,v} \tag{2.26}$$

where $\bar{\Sigma}^i$ is a valid covariance matrix independent of θ and $D^i(\theta)$ is the diagonal matrix with analogous definition to the one of the mean vector previously:

$$D^i(\theta) = diag(exp(\mathbf{Z}^i\theta)) \tag{2.27}$$

Exponential function above is applied element-wise and ensures that the diagonal elements of the covariance matrix are strictly positive. The matrix $\mathbf{Z}^i$ is a $d \times c$ matrix of parameters that defines the linear transformation of the external variables s.t. $Z^i = [U^i, \widetilde{\Sigma}^i]$ where $\widetilde{\Sigma}^i$ is $d \times 1$ offset vector. Note that both $\bar{\mu}^i$ and $\bar{\Sigma}^i$ may be instances of estimated parameters learned in classical HMM. (**Radenen2014**)

We may also consider the parametrization of the $N \times N$ transition matrix $\mathbf{A}$ which is again analogous to the above parametrization:

$$\hat{a}_{i,j} = \frac{exp(\log \bar{a}_{i,j} + \mathbf{w}_{i,j}\theta)}{\sum_{k=1}^{N} exp(\log \bar{a}_{i,k} + \mathbf{w}_{i,k}\theta)} \tag{2.28}$$

where weight vector $\mathbf{w}$ has the same dimensionality as the augmented vector of external variables θ and $\bar{a}_{i,j}$ are the original transition probabilities. (**Radenen2014**)

From the Equations 2.25, 2.26, 2.27 and 2.28 we can see that setting all elements of parameter matrices and vectors $\mathbf{V}$, $\mathbf{Z}$ or $\mathbf{w}$, $\widetilde{\Sigma}^i$ to zero results in the classical HMM. Therefore, the PHMM is a generalization of the classical HMM. (**Radenen2014**)

Altough, for a while we have assumed that the external variables are fixed, it is also possible that the external variables are time-varying, i.e. dynamic context is present as depicted by Figure 2.6. If transition probabilities are conditioned on the time-varying external variables, the hidden stochastic process is referred to as *heterogeneous Markov Chain*. Practically, in a spoken language we take certain external variables as fundamentally fixed, e.g. age, gender, etc., because they do not change over time, however, there are also external variables that are dynamic, e.g. facial expression, intonation, etc., and change throughout the conversation. Fortunately, the PHMM is able to handle this quite easily by slicing the $d \times T$ matrix of external variables Φ into $d \times 1$ vectors ϕ_t. For the purpose of this work it will be advantageous to acknowledge that extension to a Gaussian Mixture Model is also possible in the PHMM setting. (**Radenen2014**)

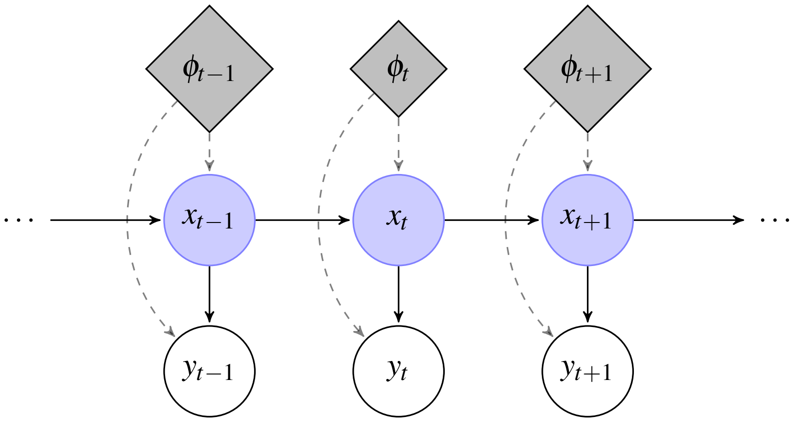


Fig. 2.6 An extended HMM with external variables influencing the hidden states and emissions.

Chapter 3

Parameter Estimation for Hidden Markov Models

3.1 Expectation–Maximization algorithm

Also abbreviated as **EM algorithm** is an iterative approach for maximum likelihood estimates of model parameters. 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. EM algorithm introduce further below is mainly based on the work of (**Dempster1977**) and (**McLachlan2008**).

In other words, we want to find the best estimate of the parameters for which the observed sequence of emissions is the most likely. This is of great importance and efficiency in Hidden Markov models where we have a space of emissions and states, where emissions are observed and states are hidden. We also know that the state space is, in our case, finite, and therefore we can enumerate all possible states since the emissions are independent but not identically distributed, s.t. the distribution of emissions depends on the hidden state.

Let us also denote $\theta \in \Theta$ which is a vector of parameters belonging to parameter space Θ. The aim of EM algorithm is essentially to find the "best" estimate of θ that maximizes the likelihood function $L(\theta|\mathbf{Y})$, this is well known as Maximum likelihood estimate (MLE) that leads to $\hat{\theta}_{MLE}$.

Complete data

If we assume that the data are complete, i.e. we have a complete set of observations $\mathbf{Y} = \{y_1, \ldots, y_T\}$, and we discard the hidden states, then MLE results in the following optimization problem:

$$\hat{\theta}_{MLE} = \underset{\theta \in \Theta}{\arg\max} L(\theta|\mathbf{Y}) = \underset{\theta \in \Theta}{\arg\max} \prod_{t=1}^{T} f(y_t|\theta) \tag{3.1}$$

where $f(y_t|\theta)$ is a probability density function of a random variable Y_t given a parameter θ.

To apply the approach for complete data, let us consider daily log-returns of BTC/USDT trading pair in past 5 years. Plotting the histogram we conclude that the log-returns may asymptotically follow normal distribution. Since the probability density function in such case is unimodal and has only one global maximum, the logarithmic transformation of likelihood function conveniently converts multiplication to addition 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 $\mathcal{N}(\mu, \sigma^2)$ given an observed sequence of log-returns $\mathbf{Y} = \{y_1, \ldots, y_T\}$ where $T \in \mathbb{N}$:

$$L(\mu, \sigma^2|\mathbf{Y}) = \prod_{t=1}^{T} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(y_t-\mu)^2}{\sigma^2}} \tag{3.2}$$

$$\ell(\mu, \sigma^2|\mathbf{Y}) = \sum_{t=1}^{T} \ln L(\mu, \sigma^2|y_t) \tag{3.3}$$

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.

$$\ell(\mu, \sigma^2|\mathbf{Y}) = -\frac{T}{2}\ln(2\pi) - \frac{T}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{t=1}^{T}(y_t - \mu)^2 \tag{3.4}$$

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 parameters as follows:

$$
\begin{aligned}
\frac{\partial \ell(\mu, \sigma^2|\mathbf{Y})}{\partial \mu} &= \frac{1}{\sigma^2}(\sum_{t=1}^{T} y_t - T\mu) \\
0 &= \frac{1}{\sigma^2}(\sum_{t=1}^{T} y_t - T\mu) \\
\hat{\mu}_{MLE} &= \frac{\sum_{t=1}^{T} y_t}{T} \qquad (3.5)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \ell(\mu, \sigma^2|\mathbf{Y})}{\partial \sigma^2} &= -\frac{T}{\sigma} + \frac{1}{\sigma^3}\sum_{t=1}^{T}(y_t - \mu)^2 \\
0 &= -\frac{T}{\sigma} + \frac{1}{\sigma^3}\sum_{t=1}^{T}(y_t - \mu)^2 \\
\hat{\sigma}^2_{MLE} &= \frac{\sum_{t=1}^{T}(y_t - \mu)^2}{T} \qquad (3.6)
\end{aligned}
$$

Inserting observed data **Y** into Equation 3.5 and 3.6, ML estimate of μ is 0.003 and σ^2 0.042 respectively. Histogram with fitted curve TODO.

Incomplete data

Although the assumption of complete data simplifies the analytical procedure of calculating the ML estimate in closed form, it is not applicable for incomplete data, i.e. when certain information is latent. This is particularly applicable for Hidden Markov Models. The alternative for the likelihood function of the complete data is therefore to use a joint probability of observed and hidden part of the data to obtain a marginal probability by summing over all possible hidden states $i \in I$, as per (**Jurafsky2008**) and (**Devavrat2014**):

$$
\begin{aligned}
\ell(\theta|\mathbf{Y}) = \sum_{t=1}^{T} \ln \mathbb{P}(Y_t = y_t|\theta) &= \sum_{t=1}^{T} \ln \sum_{i=1}^{N} \mathbb{P}(Y_t = y_t, X_t = i|\theta) \qquad (3.7) \\
&= \sum_{t=1}^{T} \ln \sum_{i=1}^{N} \mathbb{P}(Y_t = y_t|X_t = i, \theta)\mathbb{P}(X_t = i|\theta) \qquad (3.8)
\end{aligned}
$$

Since the natural logarithm is strictly monotonic increasing function the value of θ maximizes the log-likelihood as well as the likelihood function. Simply, the EM algorithm iterates over possible values of θ to find the best estimate in terms of log-likelihood difference, i.e. until convergence criterion, in (**McLachlan2008**), is satisfied:

$$|\ell(\theta^{(k)}|\mathbf{Y}) - \ell(\theta^{(k-1)}|\mathbf{Y})| \leq \varepsilon \tag{3.9}$$

where the current estimate of θ for k-th iteration is denoted with superscript (k) and the convergence threshold ε.

Since the log-likelihood function in Equation 3.7 is not analytically tractable, according to (**Bishop2006**), and involves logarithm of a sum, we need to find an alternative approach to estimate the parameters. It is, however, possible to introduce several assumptions that will eventually suffice in providing direct iterative solution to the optimization problem. Let us first construct a lower bound for the marginal likelihood in Equation 3.7.

We introduce density function $q(x)$ called "averaging distribution" and start by multiplying the joint likelihood by $\frac{q(x)}{q(x)}$. Such expression will allow for a construction of artificial weights and with the use of Jensen's inequality, as suggested by (**Gu2008**):

$$\sum_{t=1}^{T} \ln \sum_{i=1}^{N} q(X_t = i) \frac{p(Y_t = y_t, X_t = i|\theta)}{q(X_t = i)} \geq \sum_{t=1}^{T} \sum_{i=1}^{N} q(X_t = i) \ln \frac{p(Y_T = y_t, X_t = i|\theta)}{q(X_t = i)} \tag{3.10}$$

$$\ell(\theta|\mathbf{Y}) \geq L(\theta, q|\mathbf{Y}), \quad \forall q \in Q \tag{3.11}$$

where Q is a set of all possible probability distributions.

The constructed lower bound $L(\theta, q|\mathbf{Y})$ can be factorized into the expectation of the joint log-likelihood w.r.t to distribution $q(x)$ and entropy $H(q)$:

$$L(\theta, q|\mathbf{Y}) = E_{q(x)}[\ln p(\mathbf{X}, \mathbf{Y}|\theta)] + H(q) \tag{3.12}$$

where $E_{q(x)}[\ln p(X, Y|\theta)]$ is called the **expectation of complete data log-likelihood function** (or Q-function). Moreover, $H(q)$ is a constant and does not depend on the parameter θ and therefore can be omitted from the optimization problem. Therefore, decoupling the log-likelihood function into two parts, we obtain the following expression:

$$\ell(\theta|\mathbf{Y}) = E_{q(x)}[\ln p(\mathbf{X}, \mathbf{Y}|\theta)] + D_{KL}(q(\mathbf{X})||p(\mathbf{X}|\mathbf{Y}, \theta)) \tag{3.13}$$

where D_{KL} is the *Kullback-Leibler divergence* between the distribution $q(x)$ and the posterior distribution $p(\mathbf{X}|\mathbf{Y}, \theta)$. The Kullback-Leibler divergence is a measure of dissimilarity between two probability distributions, and we may interpret it as a geometrical statistical distance. It is also asymmetric and non-negative, and it is zero if and only if the two probability measures are equal which is a result of *Gibb's inequality*. (**Csiszar1975**) These properties are of great importance in the EM algorithm because in order to minimize the distance between the two distributions as described above, (**Bishop2006**) states to set the distribution $q(x)$ equal to the posterior distribution $p(\mathbf{X}|\mathbf{Y},\theta)$.

This, first step of the algorithm abbreviated as **E-step** results in estimating the function q for a θ^k, given the k-th iteration of EM algorithm, that maximizes the lower bound $L(\theta,q|\mathbf{Y})$. s.t. the "distance" from the lower bound to log-likelihood function $\ell(\theta|\mathbf{Y})$ is minimized:

$$q^{(k+1)} = \underset{q\in Q}{\arg\max} L(\theta^{(k)},q|\mathbf{Y}) = p(\mathbf{X}|\mathbf{Y},\theta^{(k)}) \tag{3.14}$$

In other words, we want to minimize distance between the complete data log-likelihood function $L(\theta,q|\mathbf{Y})$ and incomplete data $\ell(\theta|\mathbf{Y})$ with respect to function $q(x)$. Kullback-Leibler divergence (hereafter KL divergence) 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.(**Balesdent2016**) Substantially, the measure is a difference of cross-entropy denoted by $H(p,q)$ and entropy by $H(p)$ which is also always non-negative as a result of Gibb's inequality.

$$\begin{aligned}
D_{KL}(p||q) &= \sum_{X} p(x)\ln\frac{p(x)}{q(x)} && (3.15)\\
&= \sum_{x} p(x)\ln\frac{1}{q(x)} - \sum_{x} p(x)\ln\frac{1}{p(x)} && (3.16)\\
&= H(p,q) - H(p) && (3.17)
\end{aligned}$$

More rigorously, we have only defined the lower bound for the complete data log-likelihood function using arbitrary probability function $q(x)$, but the deeper examination of the lower bound with the use of KL divergence directly yields the choice of marginal posterior distribution $p(\mathbf{X}|\mathbf{Y},\theta)$ for the probability function $q(x)$:

$$L(\theta,q|\mathbf{Y}) = \sum_{i=1}^{N} q(X=i)\ln\frac{p(X=i,\mathbf{Y}|\theta)}{q(X=i)} \tag{3.18}$$

$$= \sum_{i=1}^{N} q(X=i)\ln\frac{p(X=i|\mathbf{Y},\theta)p(\mathbf{Y}|\theta)}{q(X=i)} \tag{3.19}$$

$$= \sum_{i=1}^{N} q(X=i)\ln p(\mathbf{Y}|\theta) + \sum_{i=1}^{N} q(X=i)\ln\frac{p(X=i|\mathbf{Y},\theta)}{q(X=i)} \tag{3.20}$$

$$= \ell(\theta|\mathbf{Y}) - D_{KL}(q(\mathbf{X})||p(\mathbf{X}|\mathbf{Y},\theta)) \tag{3.21}$$

Now we see that the statistical distance between the two likelihood functions is determined solely by the KL divergence. Similarly, it can be shown that Equation 3.14 is a minimization problem of KL divergence between the two distributions q and p:

$$q^{(k+1)} = \arg\min_{q\in Q}(\ell(\theta|\mathbf{Y}) - L(\theta,q|\mathbf{Y})) = \arg\min_{q\in Q} D_{KL}(q(\mathbf{X})||p(\mathbf{X}|\mathbf{Y},\theta^{(k)})) \tag{3.22}$$

To visualize the EM algorithm, refer to Figure 3.1 where red curve is the log-likelihood function $\ell(\theta|\mathbf{Y})$ and notice that the blue curve already represents the lower bound after the first iteration of E-step yielding $q^{(k)}$ as a solution. Afterwards parameter estimates of $\theta^{(k)}$ are obtained as part of the M-step s.t. the next iteration of E-step can be performed, and the lower bound is maximized again represented by the green curve.

Until now, we have considered the optimal manner in which we compute the conditional expectation $E_{\mathbf{X}|\mathbf{Y},\theta^{(k)}}[\ell(\theta|\mathbf{Y})]$ by finding the appropriate function q. Next step is, as typical for coordinate descent, selection of parameter $\theta^{(k+1)}$, with already estimated $q^{(k+1)}$, that maximizes the lower bound $L(\theta,q|\mathbf{Y})$. The goal of M-step is according to (**Dempster1977**) and (**Gu2008**):

$$\theta^{(k+1)} = \arg\max_{\theta\in\Theta} L(\theta,q^{(k+1)}|\mathbf{Y}) \tag{3.23}$$

Thus, as a summarization, the EM algorithm involves two steps:

1) **Expectation step (E-step)** - choose a function q, i.e. probability distribution, that maximizes $L(\theta,q|\mathbf{Y})$, which may be also viewed as computing the posterior distribution $p(\mathbf{X}|\mathbf{Y},\theta^{(k)})$ based on $\theta^{(k)}$:

$$Q(\theta|\theta^{(k)}) = E_{\mathbf{X}|\mathbf{Y},\theta^{(k)}}[\ell(\theta|\mathbf{Y})] \tag{3.24}$$

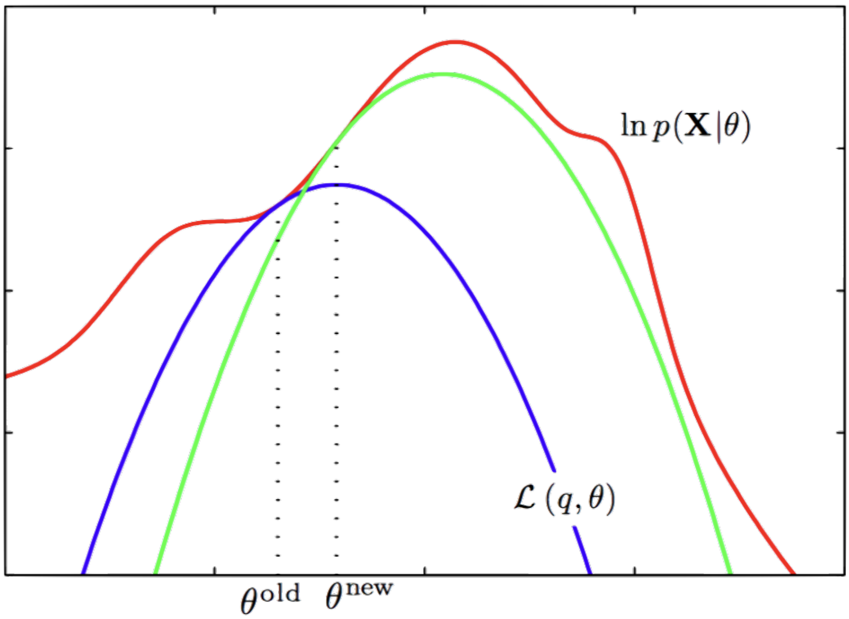


Fig. 3.1 *E-step as a problem of minimization of KL divergence represented as a gap between two likelihood functions, reprinted from (***Bishop2006***)*

2) **Maximization step (M-step)** - estimate the parameter θ that maximizes the conditional expectation $E_{\mathbf{X}|\mathbf{Y},\theta^{(k)}}[\ell(\theta|\mathbf{Y})]$.

$$\hat{\theta}^{(k+1)} = \arg\max_{\theta\in\Theta} Q(\theta|\theta^{(k)}) \tag{3.25}$$

Both of these steps are repeated until convergence criterion given by Equation 3.9 is attained assuming fixed tolerance ε.

There is also generalized version of EM algorithm called **Generalized EM** (GEM) algorithm that allows for a more flexible approach in the sense that it does not require the lower bound to be maximized in each iteration of E-step. Instead, at each iteration the expected log-likelihood is increased by a certain amount but not locally maximized, i.e. is a form of gradient ascent. Such approach is particularly useful in cases where the lower bound is not analytically tractable, or the maximization problem is computationally infeasible, e.g. in case of cHMM with parametrized covariance matrix. However, it is important to note that the GEM algorithm does not guarantee convergence to a local maximum of the log-likelihood function and might converge after fewer iterations yielding a worse estimate of the parameters. (**Bishop2006**)

3.1.1 Forward and Backward algorithm

While given a sequence of emissions denoted by $\mathbf{Y}, = \{y_0, y_1, \ldots, y_T\}$ and a model parameter vector $\theta = (\mathbf{A}, \mathbf{B}, \mathbf{p})$, we need to express the likelihood function under the model constraints as follows:

$$\mathscr{L}(\theta|\mathbf{Y}) = \prod_{t=0}^{T}\sum_{i=1}^{N}\mathbb{P}(Y_t = y_t, X_t = i|\theta) \tag{3.26}$$

$$= \prod_{t=0}^{T}\sum_{i=1}^{N}\mathbb{P}(Y_t = y_t|X_t = i, \theta)\mathbb{P}(X_t = i|\theta) \tag{3.27}$$

where N denotes a number of hidden states. As in the previous chapter, we need to rather work with log-likelihood function which is more analytically convenient and allows us to avoid numerical underflow, as in Equation 3.7, where summation is nested inside the logarithm, and therefore we cannot simply swap the order of summation and logarithm. Therefore, we need to find a way to compute the log-likelihood function in a more efficient way. Our goal is to estimate the posterior distribution of the hidden states given the observed sequence of emissions $p(\mathbf{X}|\mathbf{Y}, \theta)$ that helps us to compute the expectation of the complete data log-likelihood function.

The summation in Equation 3.7 refers to all possible permutations of the sequence of hidden states $\mathbf{X}$ which implies that we would have N^T possible sequences as seen in Figure 3.2. Furthermore, in order to calculate $\mathscr{L}(\theta|\mathbf{Y})$ we have $2TN^T$ operations which is also exponential in T and not feasible for real application. As opposed to brute-force infeasible computation described above, we take a use of **Forward** and **Backward** pass to address **E-step** of the EM algorithm, as shown by (**Bishop2006**) and (**Rabiner1989**). Resulting algorithm requires time complexity of $O(TN^2)$[1] for classical HMM and $O(TN(D+N))$ for HSMM which is a significant improvement.

Forward algorithm

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 Figure 3.2. With a help of such diagram we may imagine recording the probability of distinct subsequences at each time step t. In other words, we

[1] The O(n) symbol represents Big O notation, a convention in computer science to describe how an algorithm's running time or space needs grow as the input size increases. O(n) indicates that the growth is linear in relation to the input size n. (**Mohr2014**)

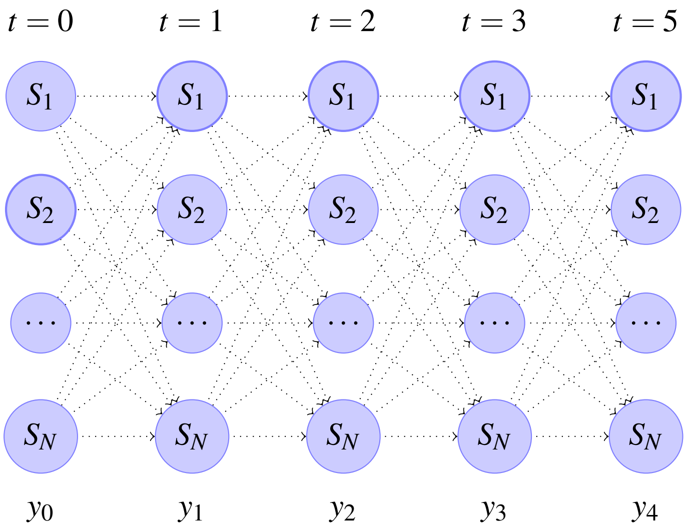


Fig. 3.2 Trellis of the emission sequence $\{y_0,\dots,y_4\}$ for T=4. Each node represents a state S_i and each arc represents a transition from one state to another. The number of possible paths is N^T.

wish to compute a joint probability while taking advantage of the conditional independence of Y_t given X_t with respect to the remaining elements of the emission sequence that happen after time t. Moreover, due to Markov memoryless property, X_t depends only on X_{t-1}. Let us now define the forward probability variable $\alpha_t(i)$ as in (**Bishop2006**) and (**Rabiner1989**):

$$\alpha_t(i) = \mathbb{P}(Y_t = y_t, Y_{t-1} = y_{t-1}, \dots, Y_0 = y_0, X_t = i|\theta) \tag{3.28}$$

Which we may also define as the probability of being in state i at time t after having observed the sequence $\{y_0, x_1, \dots, y_t\}$. The calculation therefore results in recursively summing the incoming arcs at trellis nodes:

$$\begin{aligned}\alpha_t(i) &= \mathbb{P}(Y_t = y_t, Y_{t-1} = y_{t-1}, \dots, Y_0 = y_0, X_t = i|\theta) \\ &= \mathbb{P}(Y_t = y_t|X_t = i, \theta)\mathbb{P}(Y_{t-1} = y_{t-1}, \dots, Y_0 = y_0, X_t = i|\theta) \\ &= \mathbb{P}(Y_t = y_t|X_t = i, \theta) \sum_{j=1}^{N} \mathbb{P}(X_t = i|X_{t-1} = j, \theta)\alpha_{t-1}(j) \\ &= b_i(y_t) \sum_{j=1}^{N} a_{ji}\alpha_{t-1}(j)\end{aligned} \tag{3.29}$$

In case the underlying hidden stochastic process is not strictly Markov but semi-Markov, the forward algorithm given by Equation 3.29 is no longer valid since it assumes that the transitions are geometrically distributed. Let us use derivation from (**Yu2013**) and (**Narimatsu2017**), and estimate parameters of a process that stays in state i until time $t+d-1$ and then transits to another state j at $t+d$. In order to account for the fact that the transitions are not geometrically distributed, we need to modify the forward variable $\alpha_t(i)$ s.t. the duration of the state follows some sojourn time distribution:

$$\alpha_t(i,u) = \mathbb{P}(Y_t = y_t, Y_{t-1} = y_{t-1}, \dots, Y_0 = y_0, X_t = i, \tau_t = u|\theta) \tag{3.30}$$

Forward step in the algorithm can be defined in three steps:

1. **Initialization step**: For each $i \in I$ set value at time $t = 0$ to:

   $$\alpha_0(i) = p_i b_i(y_0) \tag{3.31}$$

   In case of semi-Markov process, we need to account for the sojourn time distribution $d_i(u)$ of the state i at time u:

   $$\alpha_0(i,u) = p_i b_i(y_0) d_i(u) \tag{3.32}$$

2. **Induction step**:

   $$\alpha_t(i) = b_i(y_t) \sum_{j=1}^{N} a_{ji} \alpha_{t-1}(j) \tag{3.33}$$

   Again, in case of semi-Markov process, we have the following recursive relationship, as proposed by (**Oliver2013**), that holds for all $d \geq 1$:

   $$\alpha_t(i,u) = \alpha_{t-1}(i,u+1) b_i(y_t) + \left(\sum_{i \neq j} \alpha_{t-1}(j,1) a_{ji} \right) b_i(y_t) d_i(u) \tag{3.34}$$

   where $d_i(u)$ is the duration of the state i at time u which is referred to as the sojourn time.

   Afterwards, we recursively update current time index t to t+1 as long as $t \leq T$ starting at $t = 1$.

3. **Termination step**: once the iterative procedure is exhausted, i.e. when t=T we have the estimate of marginal likelihood of the observed sequence $\mathbf{Y}$ under the model θ denoted as $\mathscr{L}(\theta|\mathbf{Y})$:

$$\mathscr{L}(\theta|\mathbf{Y}) = \sum_{i=1}^{N} \alpha_T(i) \tag{3.35}$$

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 β variable as:

$$\begin{aligned}
\beta_t(i) &= \mathbb{P}(Y_{t+1} = y_{t+1}, Y_{t+2} = y_{t+2}, \dots, Y_T = y_T | X_t = i, \theta) \qquad (3.36)\\
&= \sum_{i,j}^{N} \mathbb{P}(Y_{t+1} = y_{t+1}, Y_{t+2} = y_{t+2}, \dots, Y_T = y_T, X_{t+1} = j | X_t = i, \theta)\\
&= \sum_{i,j}^{N} \mathbb{P}(Y_{t+1} = y_{t+1} | X_{t+1} = j)\mathbb{P}(X_{t+1} = j | X_t = i, \theta)\\
&\mathbb{P}(Y_{t+2}, \dots, Y_{t-1}, Y_t | X_{t+1} = j)\\
&= \sum_{j=1}^{N} b_j(y_{t+1}) a_{ij} \beta_{t+1}(j)
\end{aligned}$$

As for forward algorithm, the recursive computation is very similar, the only differences are in the fact that we are going backwards in time using our emission sequence while also accounting for the emission probabilities at previous step.

However, there is a slight difference in case of semi-Markov process since we also need to condition by the sojourn time in respective state i at time u:

$$\beta_t(i) = \mathbb{P}(Y_{t+1} = y_{t+1}, Y_{t+2} = y_{t+2}, \dots, Y_T = y_T | X_t = i, \tau_t = u, \theta) \tag{3.37}$$

Effectively, there are 3 main steps, as pointed out by (**Oliver2013**):

1. **Initialization step**: with default value of $\beta_T(i) = 1$ or $\beta_T(i,u) = 1$ for all $i \in I$.

2. **Induction step**:

$$\beta_t(i) = b_j(y_{t+1}) a_{ij} \beta_{t+1}(j) \tag{3.38}$$

For semi-Markov process, we have the following version of the induction step, as follows from (**Yu2013**):

$$\beta_t(i,u) = \begin{cases} b_i(y_{t+1})\beta_{t+1}(i,u-1) & u>1 \\ \sum\limits_{j\neq i} a_{i,j} b_j(y_{t+1}) \left(\sum\limits_{u'\geq 1} d_j(u')\beta_{t+1}(j,u') \right) & u=1 \end{cases} \tag{3.39}$$

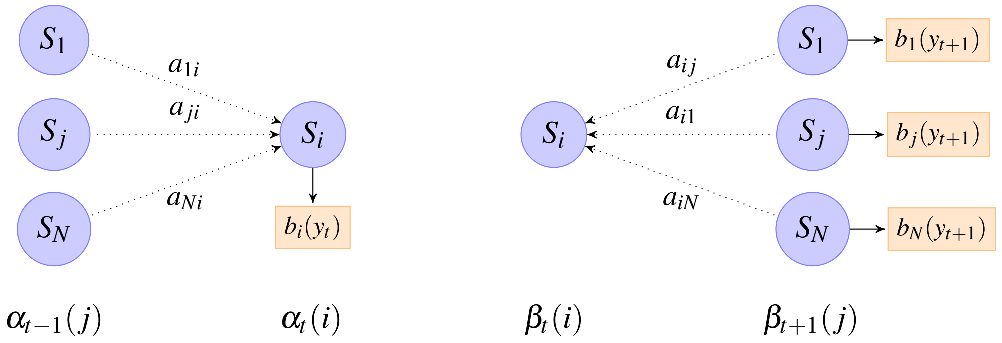
Afterwards, we recursively update current time index t to t-1 as long as $t \geq 0$.

3. **Termination step**: once the iterative procedure is exhausted, i.e. when t=0 we have the estimate of $\mathscr{L}(\theta|\mathbf{Y})$ as:

$$\mathscr{L}(\theta|\mathbf{Y}) = \sum_{i=1}^{N} p_i b_i(y_0)\beta_0(i) = \sum_{i=1}^{N} \alpha_0(i)\beta_0(i) \tag{3.40}$$

Above expression is directly equal to the resulting likelihood computed by the forward algorithm as in Equation 3.35.

Both forward and backward algorithm are used to compute the joint probability of the observed sequence and the hidden states at each time step but in different directions. These directions are also known as filtering and smoothing respectively and represented in Figure 3.3 below.



(a) Filtering using Forward algorithm

(b) Smoothing using Backward algorithm

Fig. 3.3 Figure (a) interprets recursive computation of alpha variable and (b) beta variable.

Getting everything together we show that the forward and backward algorithms coincide in solving joint and marginal posterior distribution of hidden states in the upcoming section dedicated to Baum-Welch algorithm.

3.1.2 Baum-Welsch algorithm

At the beginning of the 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 for which we can use *Baum-Welsch algorithm* as a special case EM algorithm. We will show that former step is easily computed given variables α and β established in Forward-Backward algorithm. Those will result in estimating the posterior distribution of hidden states given our emission sequence $\mathbf{Y}$, i.e. $\mathbb{P}(X_t = i|\mathbf{Y},\theta)$. (**Rabiner1989**)

In order to estimate marginal posterior distribution we might use Bayes Theorem given each subsequent time index t and denote such quantity as γ, s.t.:

$$\gamma_t(i) = \mathbb{P}(X_t = i|\mathbf{Y},\theta) \propto \mathbb{P}(\mathbf{Y}|X_t = i,\theta)\mathbb{P}(X_t = i|\theta) = \alpha_t(i)\beta_t(i) \tag{3.41}$$

where the numerator includes the joint 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. Normalizing constant is the marginal probability of the observed sequence which is equal to the likelihood function $\mathscr{L}(\theta|\mathbf{Y})$ as seen in Equation 3.35.

There is also another quantity that we need to estimate in order to complete the E-step and that is the joint posterior distribution of hidden states which is denoted as ξ. Such quantity expresses the probability of being in state i at time t and state j at time $t+1$ given the observed sequence and the model parameters θ:

$$\begin{aligned}\xi_t(i,j) &= \mathbb{P}(X_t = i, X_{t+1} = j|\mathbf{Y},\theta) \\ &\propto \mathbb{P}(\mathbf{Y}|X_t = i, X_{t+1} = j, \mathbf{Y}|\theta) \\ &= \alpha_t(i)a_{ij}b_j(y_{t+1})\beta_{t+1}(j)\end{aligned} \tag{3.42}$$

Trivially, there is a connection between γ and ξ:

$$\gamma_t(i) = \sum_{j=1}^{N} \mathbb{P}(X_t = i, X_{t+1} = j|\mathbf{Y},\theta) = \sum_{j=1}^{N} \xi_t(i,j) \tag{3.43}$$

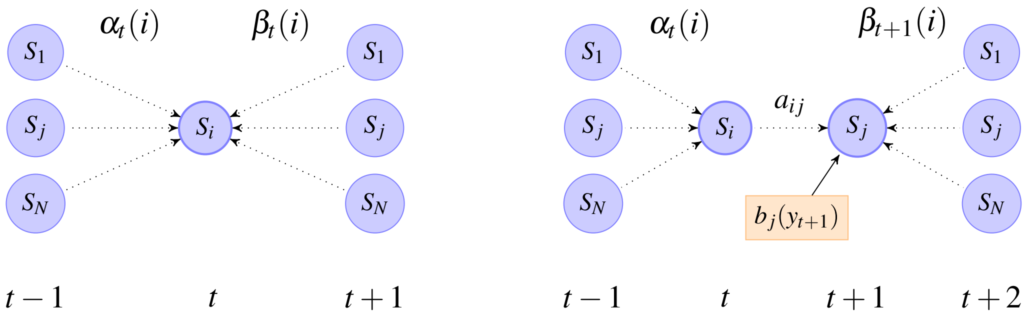
Again we ought to mention the difference in definition of the posterior distribution of hidden states in case of semi-Markov process. Let us start with the definition of ξ since there is clear analytical expression derived from forward and backward variables:

$$\xi_t(i,j) = \alpha_{t-1}(i,1)a_{i,j}b_j(y_{t+1})\left(\sum_{u\geq 1} d_j(u)\beta_{t+1}(j,u)\right) \tag{3.44}$$

However, the definition of γ is not as straightforward as in the case of Markov process, and we ought to use the backward recursion with respect to the ξ variable:

$$\gamma_t(i) = \gamma_{t+1}(i) + \sum_{j\neq i}\left(\xi_{t+1}(i,j) - \xi_{t+1}(j,i)\right) \tag{3.45}$$

where the initial condition for $t = T$ is given by $\gamma_T(i) = \sum_{u\geq 1} \alpha_T(i,u)$.



(a) Gamma variable as in Equation 3.41

(b) Xi variable as in Equation 3.42

Fig. 3.4 Trellis describing (a) $\gamma_t(i)$ variable as a product of $\alpha_t(i)$ and $\beta_t(i)$ variables and (b) $\xi_t(i,j)$ variable as a product of $\alpha_t(i)$, a_{ij}, $b_j(y_{t+1})$ and $\beta_{t+1}(j)$ variables.

E-step

First, we need to specify the equation for complete data log-likelihood function for the Hidden Markov Model (Equation 3.26), which is a joint probability distribution of observed and hidden data given our model parameters, i.e. $\ln p(\mathbf{X},\mathbf{Y}|\theta)$. Also note that here θ is a vector of parameters containing initial distribution of states, transition and emission matrices, s.t. $\theta = \{\mathbf{p},\mathbf{A},\mathbf{B}\}$, and is time invariant therefore:

$$L(\theta|\mathbf{Y}) = p_{x_1}\left[\prod_{t=2}^{T} a_{x_t,x_{t-1}}\right]\prod_{t=1}^{T} b_{x_t}(y_t) \tag{3.46}$$

where x_t is a hidden state at time t and y_t is an observed emission at time t. What is missing in the above equation is the information about the hidden states, therefore we need to marginalize over all possible hidden states at each time step. This is where we will use γ variable defined in Equation 3.41 and ξ variable defined in Equation 3.42 as a consequence of Forward-Backward algorithm. Firstly, we need to express the log-likelihood function in terms of model parameters:

$$\ell(\theta|\mathbf{Y}) = \ln p_{x_1} + \sum_{t=2}^{T} \ln a_{x_t,x_{t-1}} + \sum_{t=1}^{T} \ln b_{x_t}(y_t) \tag{3.47}$$

Next step according to EM algorithm is to take the expectation of the log-likelihood function above with respect to marginal and joint posterior distribution of the latent variable.

$$\begin{aligned} Q(\theta, \theta^{(k)}) &= \mathbb{E}_{\mathbf{X}|\mathbf{Y},\theta^{(k)}}[\ell(\theta|\mathbf{Y})] \\ &= \sum_{i=1}^{N} \gamma_1(i) \ln p_i + \sum_{i,j=1}^{N} \sum_{t=2}^{T} \xi_t(i,j) \ln a_{i,j} + \sum_{i=1}^{N} \sum_{t=1}^{T} \gamma_t(i) \ln b_i(y_t) \end{aligned} \tag{3.48}$$

To summarize, the E-step of the EM algorithm in case of HMM lies in the estimation of marginal and joint posterior distribution $\gamma_t(i)$ and $\xi_t(i)$ and deriving Equation 3.48 above. Note that since we are dealing with iterative procedure, the superscript k denotes the iteration number. However, when $k = 0$ we need to initialize the parameters of the model. Since the hidden states are assumed discrete we express transition matrix using a stochastic (Markov) matrix introduced in Chapter 1 and the initial distribution as a probability vector of length N. Moreover, in the introduction to Hidden Markov Models we stated that the marginal distribution of hidden states **p** is a stationary distribution of the transition matrix **A** and follows a categorical distribution with N categories, s.t. it is convenient to use Dirichlet distribution as it is a conjugate prior to categorical or multinomial distribution. Same approach is reasonable for each row of the transition matrix **A** and emission matrix **B**, since we are assuming that emission symbols are finite and countable. In a case of continuous emission symbols, e.g. in the context of Gaussian HMM, we usually use Gaussian distribution as a conjugate prior. Generally, we may decide to use other distributions as priors for our model parameters, e.g. uniform distribution as an uninformative prior, depending on whether we have a prior knowledge about the distribution of parameters. (**Rabiner1993**)

M-step

Once we have an initial estimate of θ, the EM algorithm performs E-step using initial parameters to get the posteriors and subsequently finds the maximum of the conditional expectation of the log-likelihood function using M-step.

No initial estimate of the parameters will guarantee that a global maximum of the log-likelihood function is attained, therefore one of the possible solutions to this problem would be to initialize parameters randomly multiple times and compare the maxima of the log-likelihood function. Furthermore, given a more complex models, one or both of the steps of the EM algorithm may become intractable, and we may need to use other methods such as generalized EM algorithm or Markov Chain Monte Carlo (MCMC) methods. (**Bishop2006**)

The optimization problem is constrained by the fact that the parameters must satisfy the following conditions since they are stochastic matrices or probability vectors:

$$\sum_{i}^{N} p_i = 1 \tag{3.49}$$

$$\sum_{j}^{N} a_{i,j} = 1 \quad \forall i \in I \tag{3.50}$$

$$\sum_{k}^{M} b_i(k) = 1 \quad \forall i \in I \tag{3.51}$$

where k denotes each possible emission symbol in emission matrix **B**. Given parameter constraints we maximize expected log likelihood function using Lagrange multipliers where the objective function is defined as:

$$\begin{aligned}\mathscr{L}(\theta^{(k-1)}, \mathbf{Y}) = Q(\theta^{(k)}, \theta^{(k-1)}) - \lambda(\sum_{i}^{N} p_i - 1) \\ - \sum_{i}^{N} \mu_i(\sum_{j}^{N} a_{i,j} - 1) - \sum_{i}^{N} \nu_i(\sum_{k}^{M} b_i(k) - 1)\end{aligned} \tag{3.52}$$

Taking partial derivatives with respect to each parameter and setting those partial derivatives equal to zero we arrive at parameter estimates which are also a final solution of the M-step as follows:

$$\hat{p_i}^{(k)} = \frac{\gamma_1(i)}{\lambda} = \gamma_1(i) \tag{3.53}$$

$$\hat{a}_{i,j}^{(k)} = \frac{\sum_{t=2}^{T} \xi_t(i,j)}{\mu_i} = \frac{\sum_{t=2}^{T} \xi_t(i,j)}{\sum_{t=2}^{T} \gamma_t(i)} \tag{3.54}$$

$$\hat{b}_i^{(k)}(y_t) = \frac{\sum_{t=1}^{T} \gamma_t(i) \mathbb{1}_{[Y_t = y_t]}}{\nu_i} = \frac{\sum_{t=1}^{T} \gamma_t(i) \mathbb{1}_{[Y_t = y_t]}}{\sum_{t=1}^{T} \gamma_t(i)} \tag{3.55}$$

We repeat these steps until the desired convergence criterion of log-likelihood difference (Equation 3.9) between subsequent iterations of the Baum-Welch algorithm is achieved. Afterwards, parameter estimates of k-th iteration are such that we have locally maximized the log-likelihood function given the emission sequence and initial setting of θ for $k = 0$.

3.1.3 Application to GMM

In previous section, we assumed that the emission symbols are discrete and countable, however, in case of Gaussian Mixture Models (GMM) continuity of the emissions is assumed. Therefore, considering univariate or multivariate normal distribution for emission probabilities requires slightly different approach in order to derive parameters of each component distribution.

Given the data $\mathbf{Y} = \{\mathbf{Y}_1, \dots, \mathbf{Y}_T\}$, where each element is an independent p-dimensional random vector, s.t. $Y_t \in \mathbb{R}^p$ and:

$$Y|k \sim \mathcal{N}_p(\mu_k, \Sigma_k) \quad \forall k \in \{1, \dots, N\} \tag{3.56}$$

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. (**Bishop2006**)

Note that now we will assume that each random vector $\mathbf{Y}_t$ is generated by one of the N mixture components and knowledge about the component k from which the data point was generated is hidden as in the previous section. To relate the parameters of the mixture model to the HMM, we introduce a latent variable $\mathbf{X}$ that follows categorical distribution with N categories, s.t. its parameters are directly related to the mixing proportions π_k of the mixture model:

$$X \sim Cat(\pi_1, \dots, \pi_N), \quad \text{where} \sum_{k=1}^{N} \pi_k = 1, \quad \pi_k \geq 0 \tag{3.57}$$

However, if we knew the component k from which the data point was generated, we could estimate the parameters of the mixture model using the maximum likelihood estimation (MLE) easily since from (**Davenport1988**) follows:

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{i=1}^{N} \mathbb{1}_{[X_i = k]} \mathbf{Y}_i \tag{3.58}$$

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^{N} \mathbb{1}_{[X_i = k]} (\mathbf{Y}_i - \hat{\mu}_k)(\mathbf{Y}_i - \hat{\mu}_k)^T \tag{3.59}$$

$$\hat{\pi}_k = \frac{N_k}{T} \tag{3.60}$$

where N_k is the number of data points generated by the component k and T is the total number of data points.

In order indicate which component generated the emission symbol, we introduce N-dimensional binary random variable $\mathbf{z}$ having a 1-of-N representation in which a particular element z_k is equal to 1 and all other elements are equal to 0. The values of z_k therefore satisfy $z_k \in \{0,1\}$ and $\sum_{k=1}^{N} z_k = 1$. (**Bishop2006**) This variable is also known as an indicator variable, and for each time step t it holds that:

$$z_{t,k} = \begin{cases} 1 & \text{if } \mathbf{Y}_t \text{ was generated by the } k\text{-th component} \\ 0 & \text{otherwise} \end{cases} \tag{3.61}$$

First, let us specify the likelihood function for Gaussian Mixture Model defined as:

$$L_c(\theta|\mathbf{Y}) = \prod_{t=1}^{T} \sum_{k=1}^{N} \pi_k \mathcal{N}(\mathbf{Y}_t|\mu_k, \Sigma_k) \tag{3.62}$$

and the log-likelihood function is:

$$\ell_c(\theta|\mathbf{Y}) = \sum_{t=1}^{T} \ln \sum_{k=1}^{N} \pi_k \mathcal{N}(\mathbf{Y}_t|\mu_k, \Sigma_k) \tag{3.63}$$

Equation 3.62, shows that the log-likelihood function has a form of a sum inside logarithm which again results in no direct analytical solution for the maximum likelihood estimation of model parameters.

Given, (**Pandolfi2021**), and that we have hidden binary variable $\mathbf{z}$ following multinomial distribution with parameters $\pi = \{\pi_1, \dots, \pi_N\}$, the log-likelihood function can be rewritten as follows:

$$\ell_c(\theta|\mathbf{Y}, \mathbf{z}) = \sum_{t=1}^{T} \sum_{k=1}^{N} z_{tk} \left(\ln \pi_k + \ln \mathcal{N}(\mathbf{Y}_t|\mu_k, \Sigma_k) \right) \tag{3.64}$$

The EM algorithm for GMM is very similar to the one defined for Hidden Markov Models in the previous chapter since we have a hidden variable $\mathbf{z}$. We start with the definition of the **E-step** of the EM algorithm, as stated by (**Sahu2020**), where the expected complete data log-likelihood function or Q-function is defined as:

$$Q(\theta|\theta^{(t)}) = \mathbb{E}_{\mathbf{z}}[\ell_c(\theta|\mathbf{Y}, \mathbf{z})|\mathbf{Y}, \theta^{(t)}] \tag{3.65}$$

Hence, inserting Equation 3.64 into 3.65 we obtain:

$$Q(\theta|\theta^{(t)}) = \sum_{t=1}^{T}\sum_{k=1}^{N} \mathbb{E}[z_{tk}|\mathbf{Y}_t,\theta^{(t)}]\left(\ln\pi_k + \ln\mathcal{N}(\mathbf{Y}_t|\mu_k,\Sigma_k)\right) \tag{3.66}$$

where $\theta^{(t)}$ denotes the parameter vector at the t-th iteration of the algorithm and $\mathbb{E}[z_{tk}|\mathbf{Y}_t,\theta^{(t)}]$ is the expected value of the hidden variable z_{ik} given the data $\mathbf{Y}_t$ and the parameter vector $\theta^{(t)}$. As previously mentioned, this quantity is also known as the posterior probability of the hidden variable z_k given the data $\mathbf{Y}$.

$$\mathbb{E}[z_k|\mathbf{Y},\theta^{(t)}] = P(z_k = 1|\mathbf{Y},\theta^{(t)}) = \sum_{t=1}^{T}\left(\frac{\pi_k^{(t)}\mathcal{N}(\mathbf{Y}_t|\mu_k^{(t)},\Sigma_k^{(t)})}{\sum_{j=1}^{N}\pi_j^{(t)}\mathcal{N}(\mathbf{Y}_t|\mu_j^{(t)},\Sigma_j^{(t)})}\right) \tag{3.67}$$

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.

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:

$$\theta^{(t+1)} = \underset{\theta\in\Theta}{\arg\max} Q(\theta|\theta^{(t)}) \tag{3.68}$$

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:

$$\hat{\pi}_k^{(t+1)} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[z_{tk}|\mathbf{Y}_t,\theta^{(t)}] \tag{3.69}$$

$$\hat{\mu}_k^{(t+1)} = \frac{\sum_{t=1}^{T}\mathbb{E}[z_{tk}|\mathbf{Y}_t,\theta^{(t)}]\mathbf{Y}_t}{\sum_{t=1}^{T}\mathbb{E}[z_{tk}|\mathbf{Y}_t,\theta^{(t)}]} \tag{3.70}$$

$$\hat{\Sigma}_k^{(t+1)} = \frac{\sum_{t=1}^{T}\mathbb{E}[z_{tk}|\mathbf{Y}_t,\theta^{(t)}](\mathbf{Y}_t-\mu_k^{(t+1)})(\mathbf{Y}_t-\mu_k^{(t+1)})^T}{\sum_{t=1}^{T}\mathbb{E}[z_{tk}|\mathbf{Y}_t,\theta^{(t)}]} \tag{3.71}$$

Thus, if we take into account Gaussian Mixture Model for the emission probabilities of the Hidden Markov Model, the EM algorithm assumptions and definition of the Q-function remain the same as in the previous chapter. The only difference is that we need to estimate the parameters of the mixture model in the M-step of the algorithm rather than the emission matrix **B**. That also implies that the Forward-Backward algorithm and Baum-Welsch algorithm sample probabilities from the d-dimensional Gaussian distribution rather than the Categorical distribution.

3.1.4 Application to cHMM

Application of the EM algorithm to the cHMM is similarly done in EM style as proposed first by (**Bobick1999**) and further extended by e.g. (**Radenen2014**). The only difference is that we need to also include the external variables in the M-step of the EM algorithm and also estimate additional parameters as opposed to classical HMM. This is due to the fact that the emission probabilities are now defined as a function of the external variables. In the simplest case assuming only the mean parametrization of the mean of the Gaussian distribution, we ought to estimate matrix $\mathbf{Y}^{i,k} = [V^i, \bar{\mu}_i]$ for each hidden state i and mixture component k. We will also define the re-estimation formula with the time-varying external variables, but the same approach could be used for the time-invariant external variables. (**Radenen2014**)

Parameter matrix $\mathbf{Y}^{i,k}$ is defined as follows:

$$\mathbf{Y}^{i,k} = \left[\sum_t \gamma_{i,t} x_t \theta_t^T\right] \left[\sum_t \gamma_{i,t} \theta_t \theta_t^T\right]^{-1} \tag{3.72}$$

where $\gamma_{i,t}$ is the marginal posterior probability of the hidden state i at time t and θ_t is the vector of external variables at time t. The re-estimation formula for the mean of the Gaussian distribution is then defined as follows:

$$\hat{\mu}_i = \mathbf{Y}^i \theta_t \tag{3.73}$$

where $\mathbf{Y}^i$ is the matrix of parameters for the mean of the Gaussian distribution for the hidden state i s.t. new estimate of $\hat{\mu}_i$ is either $d \times 1$ vector of means given hidden state i in case of d-dimensional Gaussian distribution or $d \times M$ matrix of means given hidden state i and mixture component k in case of d-dimensional Gaussian Mixture Model.

Decision to parametrize only the means of the Gaussian distribution as proposed by (**Bobick1999**) simplifies the re-estimation formula for the covariance matrix since it is identical to Equation 3.71. When we parametrize both the mean and the covariance matrix of the Gaussian distribution, we need to estimate the covariance matrix for each hidden state i and mixture component k which is not trivial since we lack closed form solution therefore we have to resort to gradient ascent in the M-step resulting in use of Generalized EM algorithm. The important decision will also come in terms of the gradient size determined by the learning rate parameter θ. (**Radenen2014**)

In covariance matrix re-estimation process we aim to compute the gradient of the expected log-likelihood denoted by function Q with respect to $\mathbf{Z}_i$. First step, according to (**Radenen2014**), is to re-estimate the means and covariance matrices of the Gaussian distribution using the same approach as for the cHMM with only the mean parametrization. Then, we set $\mathbf{Z}_i = 0$ and $\bar{\Sigma}_i = \Sigma_i$ and compute the gradient with respect to $\mathbf{Z}_i$ as follows:

$$\frac{\partial Q}{\partial \mathbf{Z}_i} = \sum_{k,t,i} M_{i,i}^{k,t,j} \frac{\partial D_{j,j}^{i}(\theta_t)^{-1}}{\partial \mathbf{Z}_i} \tag{3.74}$$

where $M_{i,i}^{k,t,j}$ and partial derivative of $D_{j,j}^{i}(\theta_t)$ are defined as:

$$M_{i,i}^{t,j} = \gamma_{i,t}\left[D^i(\theta_t) - \frac{(x_t - \hat{\mu}_i(\theta_t))(x_t - \hat{\mu}_i(\theta_t))^T}{\bar{\Sigma}_i D^i(\theta_t)}\right] \tag{3.75}$$

and

$$\frac{\partial D_{j,j}^{i}(\theta_t)^{-1}}{\partial \mathbf{Z}_{i,m,n}} = \begin{cases} \frac{-\theta_n}{D_{j,j}^{i}(\theta_t)} & \text{if } j = m \\ 0 & \text{otherwise} \end{cases} \tag{3.76}$$

So far we have tried to parametrize the parameters for each marginal distribution of emissions, however, we may also parametrize the transition probabilities given by matrix $\mathbf{A}$. Again, we are not equipped with the closed form solution for the re-estimation formula, therefore we use gradient ascent and start by setting $\bar{a}_{i,j} = a_{i,j}$ and initializing the weights $\mathbf{w}_{i,j} = 0$ which implies that we are starting with transition matrix that would be equal to the transition matrix of the HMM since external variables have no effect on the transition probabilities. Then, (**Radenen2014**) shows that gradient of the expected log-likelihood function with respect to $\mathbf{w}_{i,j}$ is defined as follows:

$$\frac{\partial Q}{\partial \mathbf{w}_{i,j}} = \sum_{k,t}\left[\gamma_{i,j,k,t} - \sum_{k,t} \bar{a}_{i,j}(\theta_t)\right]\theta_t \tag{3.77}$$

3.2 Application to AR(p) HMM model

3.3 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. (**Speagle2020**) As opposed to the previous subsection, the MCMC method also provides a measure of uncertainty of the estimated parameters. (**Spade2020**)

3.3.1 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:

$$\alpha(\theta^{(t-1)}, \theta^*) = \min\left(1, \frac{p(\theta^*)q(\theta^{(t-1)}|\theta^*)}{p(\theta^{(t-1)})q(\theta^*|\theta^{(t-1)})}\right) \tag{3.78}$$

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 3.78 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:

$$\alpha(\theta^{(t-1)}, \theta^*) = \min\left(1, \frac{p(\theta^*)}{p(\theta^{(t-1)})}\right) \tag{3.79}$$

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:

$$\mathcal{N}\mathcal{W}(\mu, \Sigma|\mu_0, \kappa_0, \nu_0, \Lambda_0) = \mathcal{N}(\mu|\mu_0, (\kappa_0\Sigma)^{-1})\mathcal{W}(\Sigma|\nu_0, \Lambda_0) \tag{3.80}$$

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 3.62. Together with the prior distribution, the posterior distribution is given by the following:

$$p(\pi_k, \mu_k, \Sigma_k|\mathbf{Y}) \propto \prod_{i=1}^{N}\prod_{k=1}^{K} \left[\pi_k \mathcal{N}(\mathbf{Y}_i|\mu_k, \Sigma_k)\right]^{\mathbb{E}[z_{ik}|\mathbf{Y}_i, \theta^{(t)}]} \mathcal{N}\mathcal{W}(\mu_k, \Sigma_k|\mu_0, \kappa_0, \nu_0, \Lambda_0) \tag{3.81}$$

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 3.67. 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.

3.3.2 Gibbs Sampling

3.4 Sequence decoding using Viterbi algorithm

Once we solve **Evaluation problem** using Baum-Welch algorithm, thus estimating the posterior marginal and joint distribution of hidden states as well as parameters of the model, we may proceed to the **Decoding problem**. The decoding problem aims to find the most likely sequence of hidden states given the emission sequence and model parameters, i.e. find the sequence of hidden states $\mathbf{Z}^*$ that most likely produced the emission sequence $\mathbf{Y}$. Finding the most likely sequence $\mathbf{Z}^*$ is also known as the maximum a posterior probability (MAP) estimate and is defined as:

$$\mathbf{Z}^* = \arg\max_{Z} p(\mathbf{Z}|\mathbf{Y}, \theta) \tag{3.82}$$

As in the encoding problem, the complexity of the optimisation problem explodes if we decide to account for all possible sequences of hidden states into N^T calculations. Such problem significantly simplifies when we calculate hidden states with the highest probability individually rather than as an entire sequence up to time t. The idea behind *Viterbi algorithm* 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-Backward algorithm* we define To do that, we first need variable $\gamma_t(i)$ as was defined in Equation 3.41. The problem, however, is not to estimate the most likely state at each time step but rather to find the most likely sequence of states up to time t. In such case $\gamma_t(i)$ is not sufficient to solve the problem even though it would solve the former problem, as stated by (**Rabiner1989**) and (**Oliver2013**).

The most individually likely hidden state at time t would be:

$$z_t^* = \arg\max_{i \in I}\{\gamma_t(i)\} = \arg\max_{i \in I}\mathbb{P}(X_t = i|\mathbf{Y}, \theta) \tag{3.83}$$

MAP estimate directly equals to the mode of the posterior distribution, i.e. value for which the likelihood or log-likelihood function attains its local maximum/maxima. However, the solution as such might not produce the most likely sequence of states. This is due to the fact that the individual estimates do not incorporate the transition probability between individually 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 individually most likely by Equation 3.82. Fortunately, the mentioned shortcomings of this approach are solved by Viterbi algorithm.

In order to find the most likely sequence of hidden states $\mathbf{Z}^* = \{z_1^*, z_2^*, \dots, z_T^*\}$ given the observation sequence $\mathbf{Y}$ it is necessary to maximise with respect to the whole sequence rather than individually to avoid less likely transitions among states. Therefore, we need variable $\delta_t(i)$:

$$\delta_t(i) = \max_{z_1,\dots,z_{t-1} \in I} \mathbb{P}(X_1 = z_1, \dots, X_{t-1} = z_{t-1}, X_t = i, Y_1 = y_1, \dots, Y_t = y_t | \theta) \tag{3.84}$$

That results in finding most likely state sequence of states up to time $t-1$ and then arriving at state i at time t. Moreover, for the purpose of the algorithm we also need variable $\psi_t(i)$ that stores the node of the incoming arc that leads to the most probable state path since $delta_t(i)$ only records a probability.[2] In other words, $\psi_t(i)$ is the most likely state at time $t-1$ that leads to state i at time t.

$$\psi_t(j) = \arg\max_{i \in I} \delta_{t-1}(i) a_{ij} \tag{3.85}$$

Viterbi algorithm proceeds in 4 main steps:

1. **Initialization step**: Given the initial distribution of hidden states $\mathbf{p}$ and emission probabilities $\mathbf{b}$ computes the initial values of δ and ψ for all $i \in I$ as:

$$\delta_1(i) = p_i b_i(y_1) \tag{3.86}$$

$$\psi_1(i) = 0 \tag{3.87}$$

2. **Recursive step**:

$$\delta_1(i) = \arg\max_{i \in I} [\delta_{t-1}(i) p_{ij}] b_j(y_t) \tag{3.88}$$

$$\psi_t(j) = \arg\max_{i \in I} \delta_{t-1}(i) p_{ij} \tag{3.89}$$

[2] Some literature distinguishes between δ and ψ, s.t. former is Viterbi probability and latter Viterbi path.

At each iterative step we update the time so that $t = t + 1$ as long as the $t < T$. Although, recursive step in Viterbi algorithm seems very similar to the induction step in the Forward algorithm there is a main difference between the two that lies in the fact that Viterbi algorithm uses maximization instead of summation over all states.

3. **Termination step**: if $t = T$ then the algorithm terminates and the most likely state of the sequence at time T is given by:

$$z_T^* = \arg\max_{i \in I} \delta_T(i) \tag{3.90}$$

4. **Back tracing step**: the optimal state sequence for $t = T-1, T-2, \dots, 1$ is then derived based on ψ variable as:

$$z_t^* = \psi_{t+1}(z_{t+1}^*) \tag{3.91}$$

To understand why the Viterbi algorithm is so efficient we need to look at the complexity of the problem in more detail. When naively going through exponentially many possible state sequences we would need to compute the probability of each sequence and then select the most likely one. However, Viterbi algorithm does not compute the probability of each sequence but rather stores the most likely state sequence up to time t and then uses it to compute the most likely state sequence up to time $t+1$, s.t. at each time step we only need to compute the probability of N possible state sequences. Hence, the complexity of the algorithm is $\mathcal{O}(NT^2)$ which is significantly lower than $\mathcal{O}(N^T)$. (**Bishop2006**)

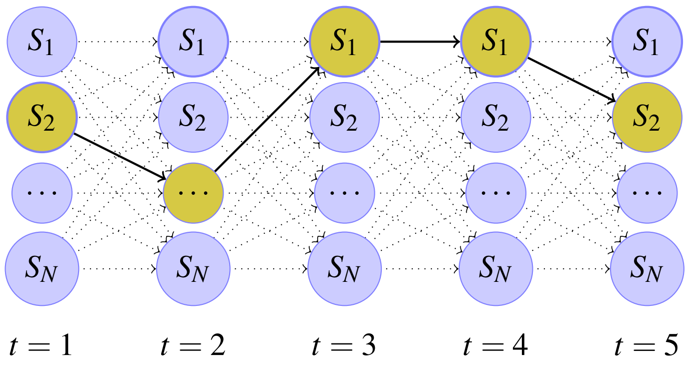


Fig. 3.5 Trellis of the observation sequence $\{y_1, \dots, y_5\}$. Bold lines illustrate the most likely state sequence Z^* found by the Viterbi algorithm.

Chapter 4

Trading strategies based on Hidden Markov Models

4.1 Selected Market Emissions

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

4.1.1 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.

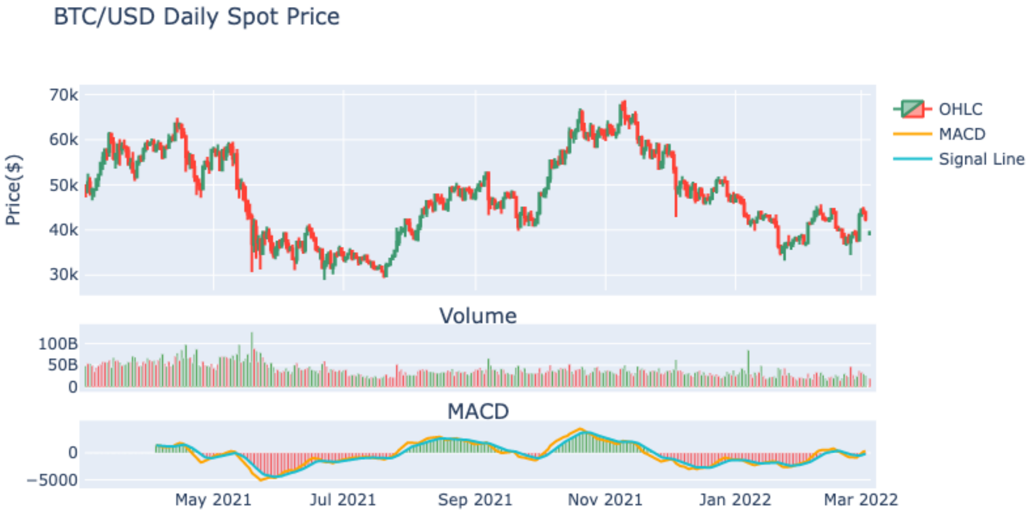


Fig. 4.1 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with subplot containing two lines for MACD and Signal line and a bar chart as their difference*

4.1.2 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:

$$SO_t = \frac{C_{t-1} - L_{14}}{H_{14} - L_{14}} \tag{4.1}$$

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.

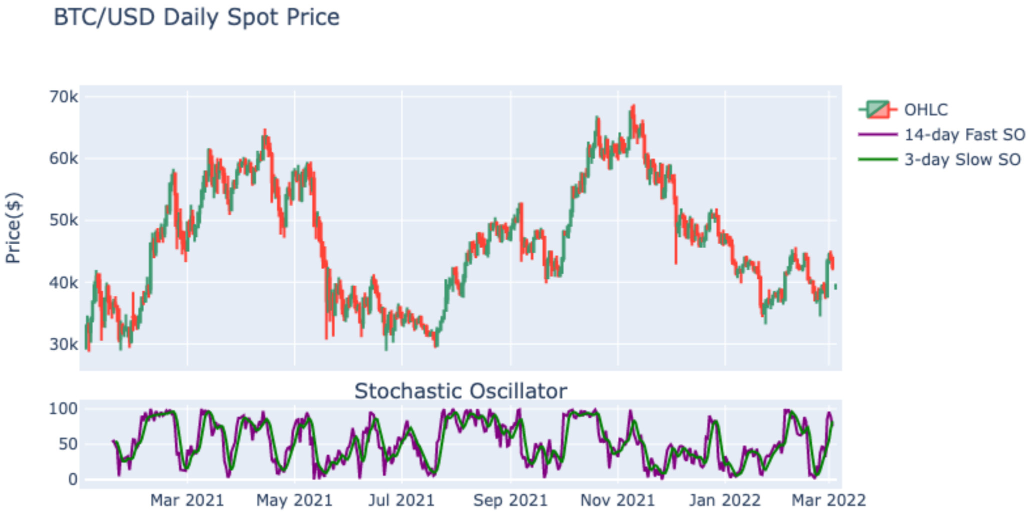


Fig. 4.2 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with subplot with line indicating Stochastic Oscillator*

4.1.3 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".

$$N_t = \frac{(Close_t - Low_t) - (High_t - Close_t)}{High_t - Low_t} \tag{4.2}$$

(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:

$$ADL_t = M_{t-1} + M_t \tag{4.3}$$

(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.

$$CO_t = \frac{\sum_{i=0}^{3}(1-\alpha)^i * Close_{t-i}}{\sum_{i=0}^{3}(1-\alpha)^i} - \frac{\sum_{i=0}^{10}(1-\beta)^i * Close_{t-i}}{\sum_{i=0}^{10}(1-\beta)^i} \tag{4.4}$$

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.

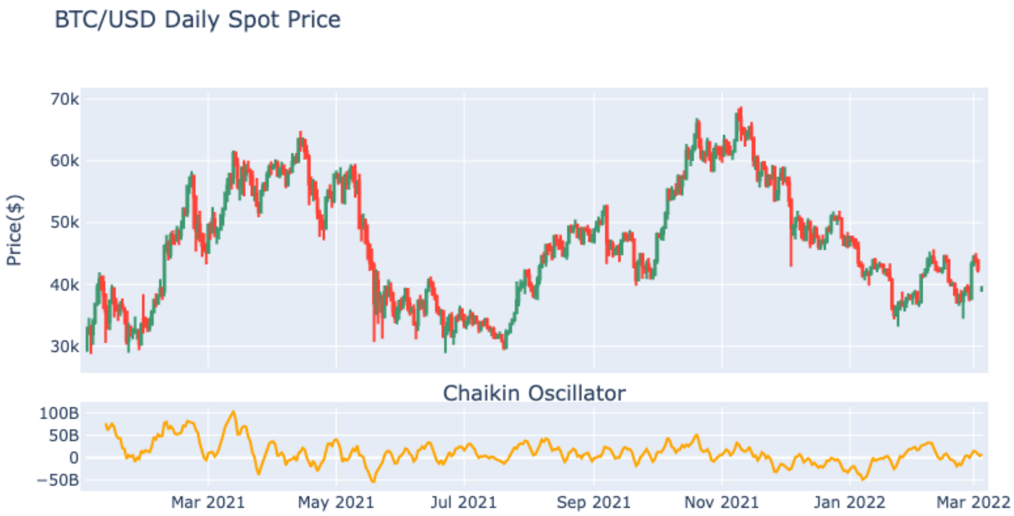


Fig. 4.3 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with subplot indicating Chaikin Oscillator*

4.1.4 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.

$$RSI_t = 100 - \frac{100}{1 + r_t} \tag{4.5}$$

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

$$r_t = \left| \frac{\sum_{i=1}^{T} (\frac{P_{t-i+1}}{P_{t-i}} - 1) \mathbb{1}_{[P_{t-i+1} - P_{t-i} > 0]}}{\sum_{i=1}^{T} (\frac{P_{t-i+1}}{P_{t-i}} - 1) \mathbb{1}_{[P_{t-i+1} - P_{t-i} < 0]}} \right| \tag{4.6}$$

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:

4.1.5 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:

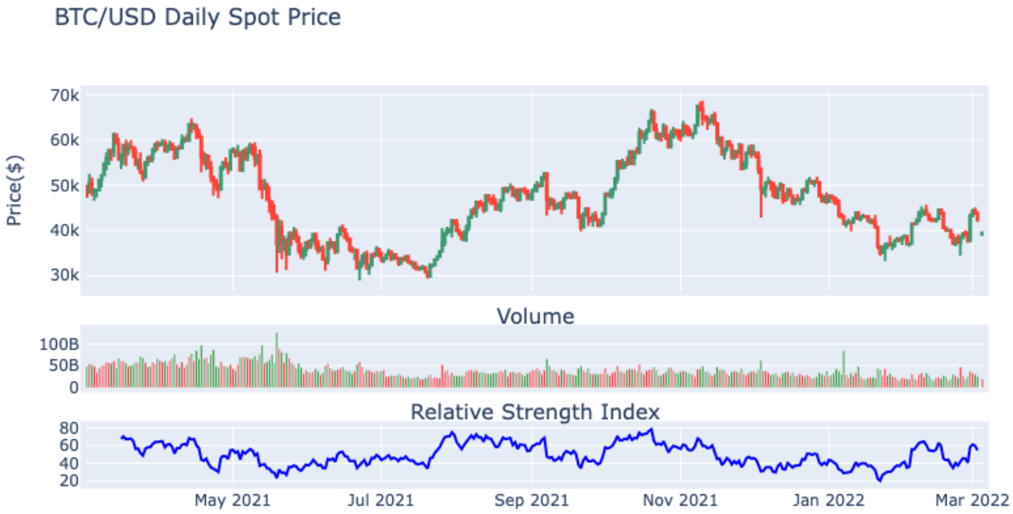


Fig. 4.4 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with a subplot containing RSI*

$$AroonUp = \frac{25 - h}{25} * 100 \tag{4.7}$$

$$AroonDown = \frac{25 - l}{25} * 100 \tag{4.8}$$

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.

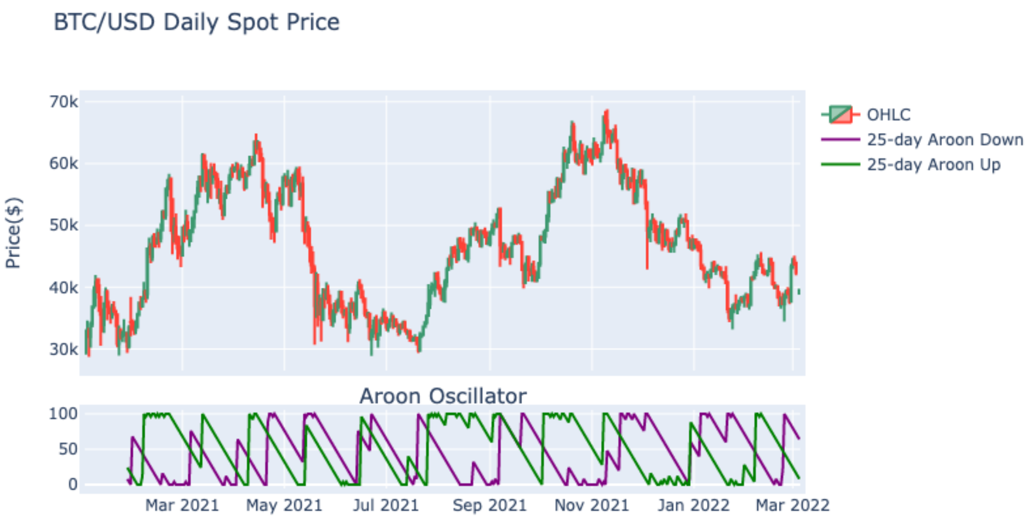


Fig. 4.5 *Candlestick graph of daily spot price of BTC/USD from March 2021 to March 2022 with subplots containing two lines for Aroon Up/Down Indicator and Aroon Oscillator*

4.2 Market regimes and trading strategies

4.2.1 Market states

4.2.2 Trading strategies

Appendix A

Maximum likelihood results

Appendix B

Python code