Contents

Introduction

The subject of prediction of stock returns has been facing criticism due to the view that market movements are random. Yet, the researchers continue to develop models aimed at increasing accuracy of predictions. Market is influenced by various, sometimes immeasurable, factors. Most models fail to account for them all, resulting in omitted variable bias. Still, it is possible to include all these variables in aggregate form in the model, and this can be accomplished with Hidden Markov Model (HMM).

Hidden Markov Model is a model which is used to account for hidden states that influence observable events (emissions). The key assumption of the model is that the process follows Markov property: only the previous state has impact on current state. Therefore, it is suitable for modelling random walks, which are itself an example of Markov process, and thus are suitable for stock market data. To specify HHM, we need to specify transition matrix (shows probability of movements between hidden states), emission matrix (shows probability of a particular emission in each state) and initial probabilities vector (shows probability that process starts in a particular hidden state), with more detailed description of the model presented in Section 4.1. There are 2 modifications of the model, which enable it to be used on both continuous and discrete data, and both of these modifications have been used for financial time series forecasting.

Market states are unknown to us and are difficult to interpret since we have no information on them. They could be periods of high or low volatility, or positive or negative investor sentiment, but it is impossible to know which interpretation is the true one. However, in our case, the lack of interpretation of states is not a significant drawback of the model. It is more important that we can include in the model all those variables which we cannot account for individually, in aggregate form as hidden market states.

The problems that HMM can solve are:

(Likelihood) Given the model and an observation sequence, output the probability of seeing such a sequence. For this, Forward algorithm (section 5.1) is used.

(Decoding) Given the model and an observation sequence, find the optimal state sequence consistent with observation sequence. Viterby algorithm (section 4.4) is applied to this task.

(Learning) Given an observation sequence, initial transition matrix, initial emission matrix and initial state probabilities vector, estimate the optimal parameters of the model, so that they are most consistent with the data. This is done via Forward-Backward and Baum-Welch algorithms (sections 5.3. and 5.4).

Although the primary fields in which HMM has been applied are bioinformatics and speech recognition, it has also been used for financial data research mainly for the following tasks: classification of market states, prediction of market trend and price prediction. The overall accuracy of HMM models prediction is rather uncertain, but in some cases researchers achieved accuracy of more than 60%. To improve performance, HMM can be used in ensemble with other models, such as Artificial Neural Network (ANN) and Genetic Algorithm (GA), or can be joined with other HMMs, trained on different types of data. This helps in overcoming certain limitations of HMM, some of which are related to the initialization of parameters.

In this paper we apply continuous HMM for prediction of direction of Amazon, Inc.(AMZN) stock returns. We use HMM to model market states as hidden variables, and close prices and returns as emissions. We train and test the model using data of different frequencies (daily, hourly and minute) to determine if HMM is more successful at forecasting data with specific characteristics. For instance, minute data is known to contain more noise than daily data. It is worth exploring how similar the performance of HMM is on these varying frequencies. We train two types of models, multivariate (containing several features) and univariate (containing one feature only), with the aim of exploring if additional information on open, high, low prices and overnight boosts performance of HMM. It could be possible that HMM can recognise patterns in the data with reasonable accuracy using only one input sequence, and other information on prices does not significantly alter its performance. If it is so, then complexity of computations could be reduced.

Another issue we look at is whether stationarity (section 2.4) is a prerequisite for application of HMM. None of the research papers we have studied check time series for stationarity, yet this characteristic is believed to be necessary for the series to be predictable. We build models both on stationary and non-stationary data for each of the three datasets for both univariate and multivariate models. For this reason, we use two target variables – stationary returns and non-stationary close prices.

Finally, we research whether predictive power of the model improves on shorter time intervals. There is a view that HMM should not be trained on data for the period of more than 5 weeks, as price patterns become less obvious. We research this question by training models on sequences of different length for each combination of stationary and non-stationary data and univariate and multivariate model types. In total we train 24 pairs of models with different features and characteristics. This enables to choose optimal specification of HMM for each data frequency and to track any interaction between parameters.

The paper is organised as follows. Section 1 presents an overview of how HMM was used for prediction and classification of financial time series. Section 2 concentrates on statistical characteristics of the data. Section 3 gives an overview of cross-validation techniques and accuracy scores that are used to measure the performance of the model. Section 4 gives a general description of HMM, whereas Section 5 describes algorithms that we use in our work. Section 6 presents the results of application of HMM to Amazon.com, Inc. stock data and discussion of these results. After that, conclusion and discussion of areas for further research on the topic follow. Annexes 1, 2 and 3 contain results of statistical tests, and annex 4 – parameters of the trained models.

1 Literature review

The researchers have used HMM to solve 2 main types of problems of financial data analysis: classification of market data and prediction of stock prices or returns. A pioneering paper which has contributed to the research related to both tasks is *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition* by [Rab89]. Although dedicated to speech recognition, this work has been widely cited in researches related to financial forecasting using HMM due to its detailed description of the theory.

HMM can be used to classify market data according to different criteria. Several researchers have used it to predict market trends and movements. [KNU13] used HMM to predict the size of market movement (low, moderate or high) by observing if prices fall or rise. There increases and decreases in stock prices are observations, and market movement sizes are hidden states. The authors use Viterby algorithm (section 4.4) to determine the most suitable sequence of states.

Another work exploring market states was done by [FW17], who use HMM to predict if the market is weak or strong in the next day, given several characteristics of the previous few days. They use in their model market indices S&P500 and CSI300, since indices are less volatile than individual stocks. They use technical indicators such as Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI) and Money Flow Index (MFI) as features.

[BGO08] use HMM with multivariate input to predict the sign of short-term market trends. They train 2 HMMs separately, one for increasing and one for decreasing trend. They use period of few weeks ending with a series of persisting increases or decreases as training set. They use discrete HMMs, and work with binary data by decoding increases and decreases. The authors point out that modelling of continuous data is a more difficult task since the distribution of observations has to be estimated. This includes accounting for heavy tails and kurtosis, along with the task of finding mean and standard deviation. The authors used observation sequences of varying length, and found that longer sequences have poorer performance, probably because trend patterns become less obvious. The predictions are made by calculating the likelihood of the sequence using each model, and assigning the type of the model that outputs highest probability. For example, if the model trained using the increasing sequence has higher probability, the forecast is positive trend. The suggested approach outperforms a single continuous model.

[DA17] use discrete HMM to predict the direction of the price change for implementation in trading algorithm. In contrast to most other authors, they use univariate model, including only close prices as a feature. They construct a trading algorithm based on 3 HMMs, trained on different patterns of price movement using different time windows. RSI index is used to switch between these models. They notice that the larger the number of observation types is, the poorer their algorithm performed, which can be explained by the deficiency of HMM in prediction of models of increased complexity. Nevertheless, their algorithm achieves a return which exceeds buy-and-hold strategy.

Another work dedicated to the prediction of stock price direction is done by [FYL+19], who employ a fusion approach with features of different nature used in different models. They use qualitative data from news articles and they also use some quantitative characteristics. They are used separately in 2 different HMMs, so that 1 state is associated with 2 observations.

Also, the authors take into account correlation between stocks, which contrasts with previously mentioned works, where stocks are assumed to be independent. They achieve the proportion of correct predictions of almost 63 %.

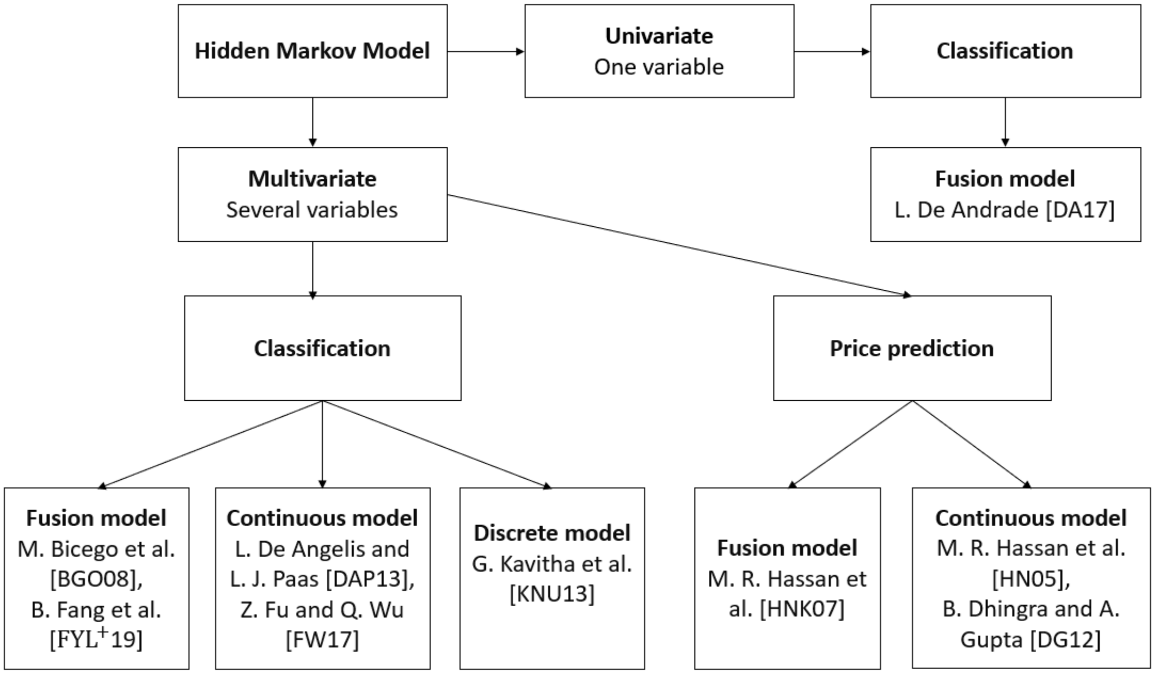
Another application of HMM is to distinguish between specific market periods. For instance, [DAP13] used HMM to distinguish between periods of crisis and stability . The authors obtain results superior to Generalized Autoregressive Conditional Heteroscedasticity (GARCH) model. Their model gives best results with 3 positive and 4 negative states of the market.

HMM can be used to predict the numeric value of price. [HN05] use in their work opening, closing, high and low prices as inputs to the model with the goal of predicting closing price for the next day. After estimation of model parameters, to make prediction they propose to find in the past data a pattern similar to the one of the current day. When similar day is located, they calculate the difference between the price in this day and the next day and add this difference to the current day to predict price of tomorrow. According to their results, this method in general is less accurate than Artificial Neural Network (ANN). In a later work, [HNK07] modify their approach by proposing a fusion model of HMM, ANN and Genetic Algorithm. ANN is used to transform input to HMM, and the latter is trained using GA. The authors propose to make predictions in 2 ways. The first one is reported in their former work [HN05], where one pattern similar to the one to be predicted is located in the historical data. In the second approach, they find several patterns in historical data, which are similar to the one to be predicted , and take weighted average of the differences of values in each chosen time period. Recent days are given more weight. ANN is used to add noise to the initial observation sequence. GA is used to generate initial parameters for HMM, which are optimal given initial sequence, and then HMM is trained using Baum-Welch algorithm (section 5.4). Baum-Welch algorithm depends on the initial parameters, so their initial values should be accurately selected. The process iterates until some stopping criterion is reached. The authors claim that the number of states should be equal to the number of observation types, and that the best model for continuous observations is left-right. The researchers note improved performance compared with simple HMM, and rather comparable results with Autoregressive Integrated Moving Average (ARIMA).

[DG12] propose to predict prices by calculating the most probable sequence of observations over all possible future prices. In other words, the authors implement Forward-Backward algorithm (section 5.3). However, it can be computationally expensive if the number of states or possible price values is large. They use fractional changes of open, close, high and low prices, not their absolute values, so that the variation in prices stays constant over the dataset. Still, the predictions are made by using discretization of the data, whilst the model is trained assuming the data is continuous, which shows there is a degree of contradiction. Similarly to [HN05], they add the predicted fractional change to the open price of day of interest. However, they use ergodic type of HMM, where all transitions between states are possible. Overall, they find that in 75% of the tested cases the predictions are superior to those of ANN and ARIMA.

One of the most noticeable problems in HMM is the determination of the number of hidden states. [Ngu18] have chosen it by computing Akaike information criterion (AIC) (section 2.4.1, equation 2.11) and Bayesian information criterion (BIC) and selecting the number of hidden states corresponding to the minimum of these measurements. Testing from 2 to 6 states, they find that optimal value for S&P500 is 4 states. [HN05] have also used 4 states, but [FW17] claim that for different markets, there is a different number of optimal states, with more mature

Figure 1: The overview of model types used for classification and price prediction problems. Source: author



markets requiring less states than emerging ones. Furthermore, they claim that the same state can change its nature and characteristics (for example, the prices in the state can start to behave differently), and as it is difficult to interpret the states, it is impossible to learn which state has changed.

Thus, the use of HMM for financial data research evolves. More complex fusion models are developed, which enable to overcome some of the limitations of HMM. Most of the papers concentrate on the prediction of price direction and the size of change, rather than on prediction of numeric price value, since the model itself is more suitable for classification. The procedure with the numeric value prediction is rather complex and requires a lot of data, yet may not yield accurate results.

In our work, we follow several approaches discussed in the papers. We follow [BGO08] in training 2 separate HMMs, each having its own direction of price change (positive or negative). However, we use continuous type of the model (section 4.2). In line with the finding of [FW17], we use choose the number of hidden states individually for each model. As suggested by [DG12], we use ergodic type of the model (Section 4.3).

2 Exploratory Data Analysis

2.1 Data Description

Daily, hourly and minute data on open, close, high and low prices for common stock of Amazon.com, Inc (*AMZN*, ISIN US0231351067) is used.

Minute data was collected from Finam.ru[1], which references Morningstar data. Morningstar uses quotes of NASDAQ, where Amazon stock trades. NASDAQ quotes at last sale price. The data is for the period 19.02.2019 – 22.05.2020, totalling 92,492 observations. For older data, the time period is from 9.32 a.m. to 16 p.m. each weekday, for more recent data – from 10 a.m. to 16 p.m. Opening price is the price of the first sale in a given minute, closing price – price of last sale in the same minute.

In order to preserve consistency of data across different frequencies, hourly and partly daily datasets were created from minute dataset. For hourly data, opening price is the price of the first sale in a given hour, closing price – price of last sale in the same hour. For daily prices, opening price is the price of first sale on a given day, closing price – price of last sale. In total, there are 2,316 observations in the hourly dataset for the period 19.02.2019 – 22.05.2020. However, in the daily dataset there are only 317 observations, constructed from minute dataset. This is a small sample to train a model, so additional daily prices for earlier dates were obtained from Yahoo Finance[2], which provides real-time data from NASDAQ. In total, there are 1858 daily observations, for the time period 03.01.2013 – 22.05.2020

In all datasets, closing prices in a particular day do not equal opening prices in the subsequent day, which means there has been some influence of overnight trading on opening price. This change in price can convey such information as investor sentiment, and provide a proxy for price movements in the next day. Therefore, we include in the model this change in price due to after-hours and pre-hours trading as explanatory variable *overnight*. At the same time, we smooth prices by setting opening price in a given day equal to closing price at previous day, so as to avoid double-counting the effect from overnight trading. Since overnight is calculated between two days in a row, and not between trading hours and minutes in a row, then it suffices to calculate *overnight* on daily dataset only. However, the *overnight* variable cannot be used as input while making predictions for minute and hour datasets, since for them it is not regularly measured, meaning that not all observations would have corresponding *overnight*.

[1] Finam data

[2] Yahoo Finance data

2.2 Descriptive statistics

Descriptive statistics, calculated for each target variable, is presented below. Descriptive statistics for other sequences used in empirical part can be found in Annex 1.

Table 1: Sample statistics on AMZN stock returns (USA dollars) for the period 13.04.2020 – 22.05.2020. Source: calculations of the author

Frequency	Per day	Per hour	Per minute
Std Deviation	61.0383	20.5820	3.0124
Mean	13.5131	1.6965	0.0436
Scaled to day (6 hours) std deviation	61.0383	50.4154	57.1563
Scaled to day (6 hours) mean	13.5131	10.1790	15.6960
Number of observations used	29	231	8995

Table 2: Sample statistics on AMZN stock close prices (USA dollars) for the period 13.04.2020 – 22.05.2020. Source: calculations of the author

Frequency	Per day	Per hour	Per minute
Std deviation	66.6192	69.5947	71.8372
Mean	2371.8238	2369.2207	2367.4283

Scaled statistics for minute and hourly returns are calculated for differences of consecutive close prices for each hour and minute over the period of 6 hours.

Statistics are comparable across frequencies for close price data. For returns data, standard deviations are quite incomparable across frequencies if scaling is done. For scaling, we use the following formula:

$$\sigma(nD) = \sqrt{n}\sigma(D) \tag{2.1}$$

where n is some constant, D is a random variable and σ is standard deviation. Using it, we can compare standard deviations between hourly and minute returns. In an hour, there are 60 minutes, thus the formula for scaled variance of the minute returns is:

$$\sigma_{hourly} = \sqrt{60}\sigma_{minute} \tag{2.2}$$

Thus we calculate 3.0124 $\sqrt{60} = 23.3339$, which is rather similar to standard deviation of hourly returns, equal to 20.5820. In a similar manner, we can compare standard deviations of daily and hourly returns. We normalize statistics for hourly data: $20.5820\sqrt{6} = 50.4154$. However, it differs from 61.0383. Finally, compare minute and daily data: $3.0124\sqrt{360} = 57.1563$ – different from standard deviation of daily data. There are several explanations why this difference between scaled and actual variances is observed. The formula for the sample standard deviation is

$$\sigma = \sum_{i=1}^{n} \frac{(x_i - \hat{x})^2}{n-1} \tag{2.3}$$

where n is sample size, x_i is some observation and $\hat{x}$ is the average of all observations. We can see that the larger is n, the smaller is the standard deviation, so naturally standard deviation of minute returns is the smallest. The numerator of 2.3 is the square of deviation from the mean. Most of the time, this deviation is smaller for data of higher frequencies (minute data), since prices do not change much between subsequent moments. Also, if there are any shocks to prices, they do not have enough time to show their influence, in contrast to daily data, for example. Thus, as subsequent prices are similar, the deviation for minute returns is lower. It is the characteristic of the dataset that cannot be transmitted through scaling, because daily data does not include this effect of smoothing of prices for higher frequencies. This partly explains the difference between scaled minute and actual daily standard deviations. This smoothing effect which arises with higher frequencies can be observed in the following graphs:

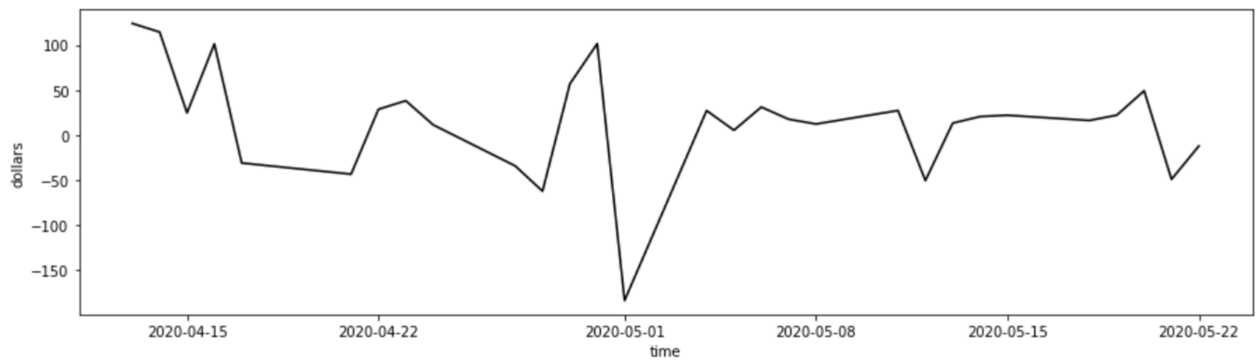


Figure 2: Daily returns for the period 13.04.2020 – 22.05.2020. Source: calculations of the author

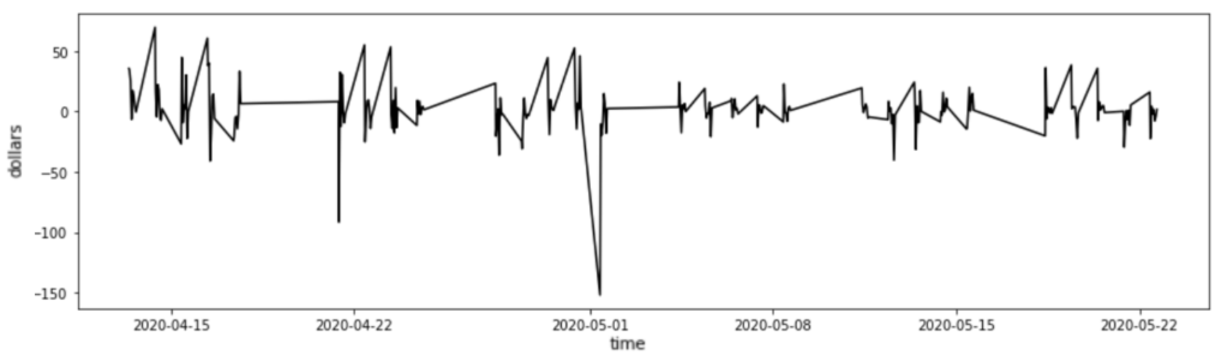


Figure 3: Hourly returns for the period 13.04.2020 – 22.05.2020. Source: calculations of the author

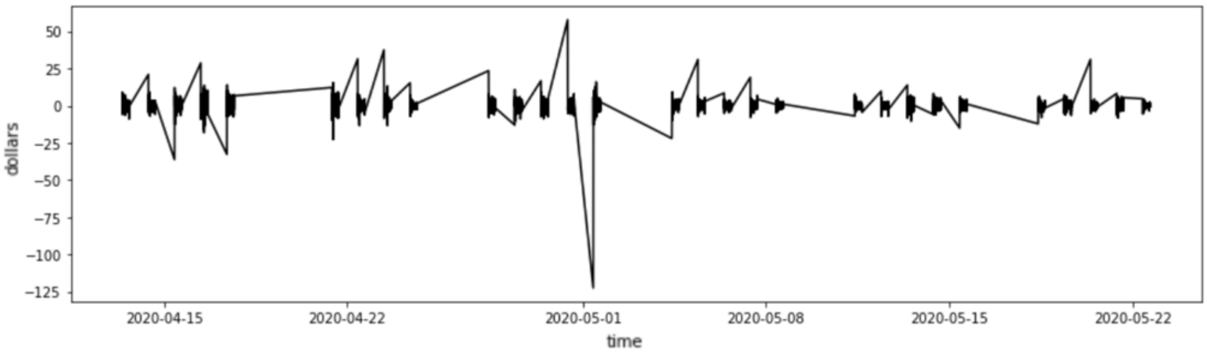


Figure 4: Minute returns for the period 13.04.2020 – 22.05.2020. Source: calculations of the author

Moreover, there are some discrepancies due to approximation, so this provides additional explanation for the differences.

Means for return data are incomparable across frequencies, since returns are both negative and positive. This means that for more frequent changes in prices there is more of the effect that negative returns cancel out positive ones. Thus, this implies that the mean is the smallest for minute data. This effect is again not transmitted across datasets. For instance, scaled minute return mean to the daily return mean is $0.0436 \cdot 360 = 15.696$, which is more than actual daily mean return. This can be explained by the fact that changes in prices within day can be larger than the resulting difference between daily close and open prices.

In what follows, the presented notation will be used:

- n — sample size
- y_{t-k} — k_{th} lag of observation y. For y_1 at the start of the day, there is only zero lag, which is y_1 itself.
- x — some independent variable
- $\hat{x}$ — mean of x
- ϵ_t — disturbance term
- α, β — regression coefficients
- Δy_t — difference between y_t and y_{t-1}
- γ — constant

2.3 Data Predictability Analysis

2.3.1 Hurst Exponent: Definition

Before proceeding to the prediction of stock returns, it is necessary to determine if they can be predicted at all. Given the assumption of efficient market – prices reflect all available information – returns follow random walk and are therefore unpredictable. This argument was introduced in the work of [Fam65], who defined predictability as the possibility to predict future stock price, given historical information. However, there have been several studies that argue that returns are in fact predictable. For instance, [Bis11] applies spectral analysis to stock returns and finds evidence of periodicity and complex patterns. One approach to determine predictability is through the use of Hurst exponent, following the work of [JTJ16].

Hurst exponent can be considered as one of the measures of long-term memory. Long-term memory is used to provide a context in which the model treats inputs. In other words, the model can account for the relation of one observation to another in a sequence. This should improve predictability since it means some information on the observation to be predicted can be inferred from past observations. Hurst exponent is calculated for a range of observations of length n. First, the range $R(n)$ of total deviations from the mean is computed:

$$Z_t = \sum_{i=1}^{t} x_i - \hat{x} \qquad t = 1, 2, ..., n \tag{2.4}$$

$$R_t = max(Z_1, ..., Z_t) - min(Z_1, ..., Z_t) \qquad t = 1, 2, ..., n \tag{2.5}$$

$S(n)$ is standard deviation:

$$S_t = \sqrt{\frac{\sum_{i=1}^{t}(x_i - \hat{x})^2}{t}} \qquad t = 1, 2, ..., n \tag{2.6}$$

Finally, rescaled range is calculated as:

$$\text{Rescaled range} = \frac{R_t}{S_t} \qquad t = 1, 2, ..., n \tag{2.7}$$

To calculate Hurst exponent, power law is used. The ansatz for Hurst exponent takes the following form:

$$\frac{R_t}{S_t} = cn^H \tag{2.8}$$

where c is some constant, and H is the Hurst exponent. The equation 2.8 is fit by taking $log\frac{R_t}{S_t}$ and then fitting it to the straight line. Still, this approach yields biased results, so it is recommended for large samples ($n > 100$). Next, H is compared to a range of values. If $H \in (0, 0.5)$, then the time series is called anti-persistent, since its values tend to alternate in sign. Thus, over long-term it reverts to its mean. If $H \approx 0.5$, then time series is called Brownian, and its future values do not depend on past values. Finally, if $H \in (0.5, 1)$, then the time series is persistent — its values are positively correlated.

2.3.2 Results of Hurst Exponent Application

Tables 3 and 4 give the statistics on Hurst exponent for returns and close prices. Hurst exponent calculated for other sequences used in empirical research can be found in Annex 2. There are 5 types of Hurst exponent, we take Corrected R over S Hurst exponent since this approach is most widely used as it minimizes bias. Notably, a time series becomes Brownian for long time periods, but for short time series certain patterns are exhibited. We also take into account that for the same number of observations used (we use 800 observations), the results may differ across time. This means that some patterns of time series may change. Therefore, we present results for first and last 800 observations. As can be seen, certain patterns in time series are present.

Table 3: Hurst exponent for returns, calculated with the R function *hurstexp* from *pracma* package. Source: calculations of the author

Frequency	Daily	Hourly	Minute
Exponent value (first 800 obs.)	0.5608	0.6131	0.5736
Result (first 800 obs.)	persistent	persisitent	persistent
Exponent value (last 800 obs.)	0.5528	0.5683	0.5300
Result (800 obs.)	persistent	persisitent	persistent

Table 4: Hurst exponent for close prices, calculated with the R function *hurstexp* from *pracma* package. Source: calculations of the author

Frequency	Daily	Hourly	Minute
Exponent value (first 800 obs.)	1	1	1
Result (first 800 obs.)	persistent	persistent	persistent
Exponent value (last 800 obs.)	1	0.9845	0.9915
Result (800 obs.)	persistent	persistent	persistent

Close prices and returns are persistent for all frequencies, yet the pattern is more pronounced for close price data. It could be the case that by taking difference of close prices some information about patterns is eliminated, so Hurst exponent has lower value for return data. Therefore, close price data shows strong patterns and can be forecasted. For return data, some slight patterns may be recognised and can be predicted. Time series graphs for returns support this conclusion – in general, positive returns are followed by positive ones, and negative - by negative.

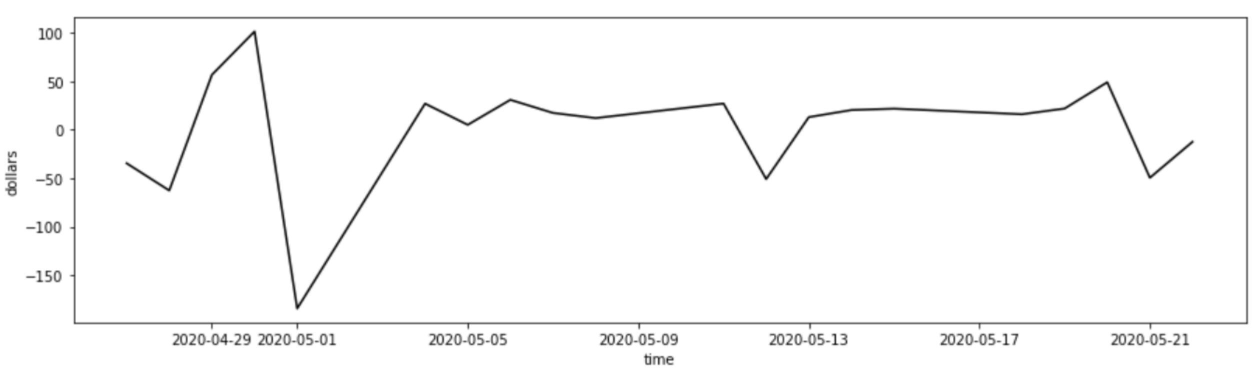


Figure 5: Daily returns for the period 27.04.2020 - 22.05.2020. Source: calculations of the author

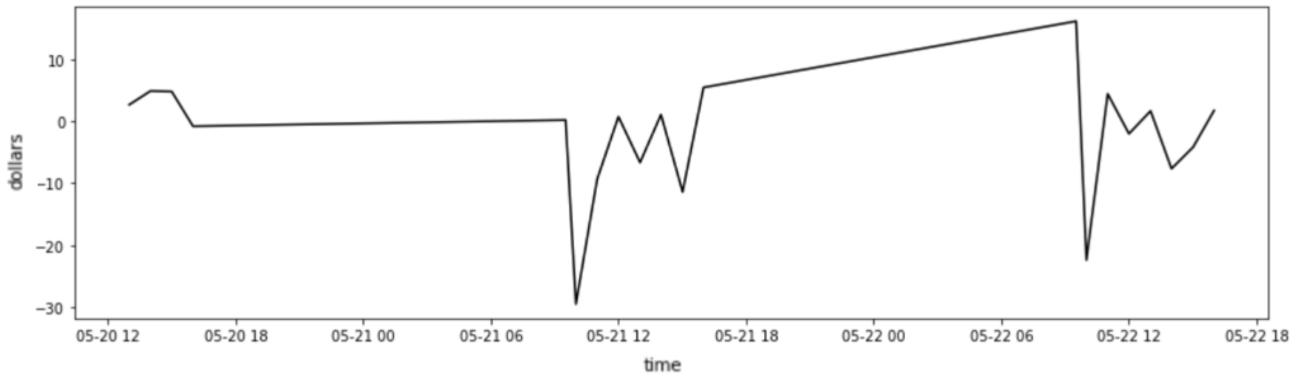


Figure 6: Hour returns for the period 20.05.2020 12:00:00 - 22.05.2020 16:00:00. Source: calculations of the author

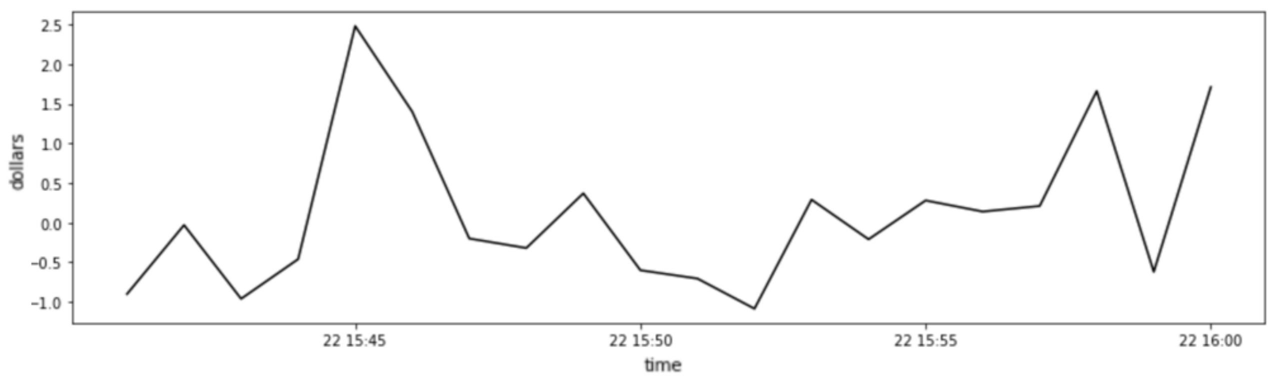


Figure 7: Minute returns for the period 22.05.2020 15:40:00 - 22.05.2020 16:00:00. Source: calculations of the author

2.4 Stationarity

Stationarity means that statistical properties of a process do not change over time. In many cases, stationarity is a prerequisite for predictability of time series. In this sense, it is similar to Hurst exponent as it guarantees the statistical patterns of the series stay constant in time. If time series is not stationary, in some cases is is possible to difference transform it to make it stationary. This means taking differences between two subsequent values: $y_t - y_{t-1}$. Currently, there are mostly parametric tests used to check stationarity, while non-parametric are not widely used. Parametric tests check for certain types of stationarity only, however, they are not always able to recognise complex patterns. In our work we use 2 tests – Augmented Dickey-Fuller (ADF) test and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test.

To select right specification for tests, it should be identified if there is any trend in the time series. However, we cannot test for trend as the time series does not satisfy assumptions of standard tests for trend. For instance, Mann-Kendall test and Spearman's rank correlation test check only for linear trends, which is unlikely to hold for returns. However, it is unlikely there are any trends present in the financial time series data.

2.4.1 Dickey–Fuller Test

ADF test is a unit root test for stationarity. A process can be decomposed into several monomials, each corresponding to a root. If a root equals 1, then it is a unit root. Alternatively, the presence of unit root means that in the following equation:

$$y_t = \alpha y_{t-1} + \beta x + \epsilon_t \tag{2.9}$$

$\alpha = 1$. The presence of a unit root indicates time series is non-stationary. The null hypothesis of ADF is H_0: non-stationary. ADF test is based on 3 regression models, which differ based on characteristics of the data. To select appropriate one, we note there is neither upward nor downward trend, so we use specification without trend. We calculate mean for close prices for all frequencies and for daily and hourly returns, and as they are non-zero, we include constant (drift):

$$\Delta y_t = \gamma + \alpha y_{t-1} + \sum_{n=1}^{m} \beta_n \Delta y_{t-n} + \epsilon_t \tag{2.10}$$

where Δy_{t-n} is lagged difference. As mean for minute returns is near zero, constant γ is excluded.

To run the above test in 2.10, we should choose the number of lags after which autocorrelation is eliminated. We use minimization of AIC for this purpose, where AIC is:

$$2k - 2ln(L) \tag{2.11}$$

k is the number of parameters and L is the maximum value of likelihood function. The drawback of ADF test is that it often fails to distinguish between the cases when $\alpha \simeq 0$and $\alpha = 0$ – a near observation equivalence problem.

For minute data, the test was computed not on the whole dataset but on 5,000 only observations because of high computational cost.

2.4.2 Kwiatkowski–Phillips–Schmidt–Shin Test

KPSS tests for a different type of stationarity – trend-stationarity. It differs from unit-root stationarity in that, given a shock, trend-stationary time series has its mean unaffected whereas mean of unit-root series is affected. In other words, the slope of trend-deterministic time series should not change permanently for the entire time period. KPSS test has the null hypothesis H_0: stationary. KPSS splits time series into random walk w_t, deterministic trend βt and stationary error ϵ_t:

$$y_t = w_t + \beta t + \epsilon_t \tag{2.12}$$

Specification of test with drift and without trend is used, so the regression is

$$y_t = \gamma + w_t + \epsilon_t \tag{2.13}$$

2.4.3 Results of the Tests on Stationarity

The decision rule is as follows: we take 5% confidence level, and if the p-value is less than 5%, then we reject H_0. All tests conclude that all time series for returns are stationary, for close prices – non-stationary. Thus, despite the fact that Hurst exponent shows rather weak patterns for return data, the presence of stationarity means that statistical properties of the series are unchanged, and the predictions can be made. Results of tests on stationarity for other sequences used in empirical research can be found in Annex 3.

Table 5: Results of stationarity tests for returns. Source: calculations of the author

Frequency	Daily	Hourly	Minute
ADF p-value	0	0	0
KPSS p-value	0.1	0.1	0.1
Conclusion	stationary trend stationary	stationary trend stationary	stationary trend stationary

Table 6: Results of stationarity tests for close prices. Source: calculations of the author

Frequency	Daily	Hourly	Minute
ADF p-value	0.9948	0.8331	0.5547
KPSS p-value	0.01	0.1	0.01
Conclusion	non-stationary	non-stationary trend stationary	non-stationary

It should be noted that KPSS test has a relatively high probability of making Type I error, which means that H_0 is rejected when it is true. ADF suffers from the same problem. However, when results of these tests agree, then the time series most likely has the characteristic on which they agree.

2.5 Markov property

The process modelled in HMM is assumed to follow Markov property, which means that current state at time t depends only on state at time $t-1$. All observations that happened at time $t-2$ and later are irrelevant. Let S_i be a particular state, Q be a sequence of states. Then Markov property is defined as:

$$P(q_t = S_i | q_1, q_2, ..., q_{t-1}) = P(q_t = S_i | q_{t-1}) \tag{2.14}$$

In other words, the future is independent of the past, given the present [KNU13]. However, there are higher-order Markov models of order k, which take into account k past observations:

$$P(q_t = S_i | q_1, q_2 ... q_{t-1}) = P(q_t = S_i | q_{t-k}, ..., q_{t-2}, q_{t-1}) \tag{2.15}$$

If higher-order Markov chains were used in HMM, the complexity of the model would greatly increase. First-order Markov property allows HMM to be computed more efficiently.

To proceed, we need to justify the assumption that Markov property holds on our dataset. It is required that states should follow this property. However, states are unobservable, and we cannot reliably check that the property holds. We make an assumption that Markov property is satisfied for our data. Intuitively, due to volatile market environment, market state should depend on preceding state only. If it is indeed so, then the accuracy of predictions of our models would be generally high.

3 Validation of the Model

3.1 Types of Cross-Validation

Cross validation (CV) is a technique used to estimate forecast error to determine how the model performs on the real data. One of the methods is k-fold cross validation. We use its modification, suitable for validation of time series models.

For k-fold CV, the first step consists in randomly splitting data into k subsets. Then $k-1$ subsets are used to train the model, and the remaining subset is used for testing. The forecast error is recorded for this prediction. Then, the procedure is repeated for the next subsets, such that each subset has been used as CV sample. The errors across all iterations are averaged to arrive at final error, which serves as assessment of the performance. K-fold CV assumes that CV set can happen to be in the middle of train set. Its main advantages are that it is relatively computationally inexpensive and has rather low variance of error estimate.

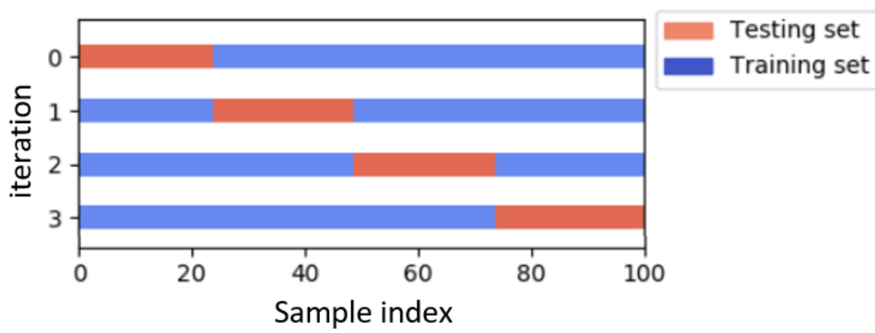


Figure 8: K-fold cross validation. Source: scikit-learn documentation

For CV of time series, it is required that training and test sets come in chronological order – a method called forward chaining. We use this method to split data into training and CV samples. The reason why the samples should come in chronological order is because time series usually has correlation between subsequent observations, and hence between subsequent subsets. Thus, if the CV set was chosen randomly, it would have many observations correlated with training observations, so essentially the model would know some information about CV sample beforehand, resulting in artificially low CV error. This should be avoided, as in reality we would have no such information when forecasting some variables using already trained model. This problem is solved by purging, meaning that any training observations that come after CV observations are dropped from training set.

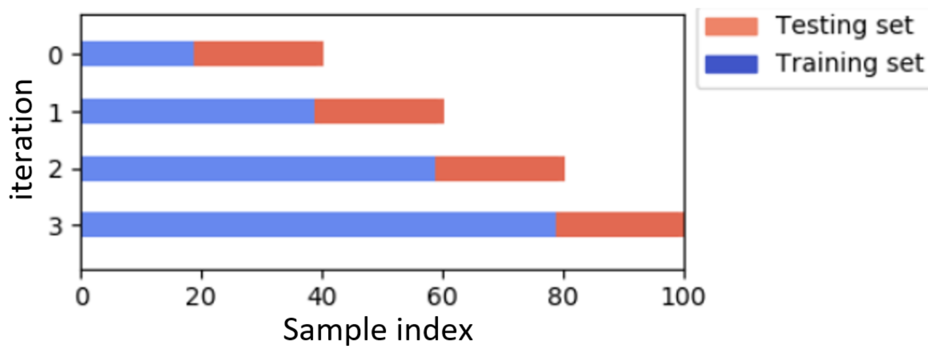


Figure 9: Time-series cross validation. Source: scikit-learn documentation

However, standard forward chaining has some limitations. One of them is that given training and CV samples come in chronological order, due to high correlation the information from the last observations in training set can be already contained in the first few observations of CV set. It is required that these should be independent, so a blocked forward chaining should be used. This means that some observations between training and CV sets should be omitted so as to make these samples independent – a technique called embargoing.

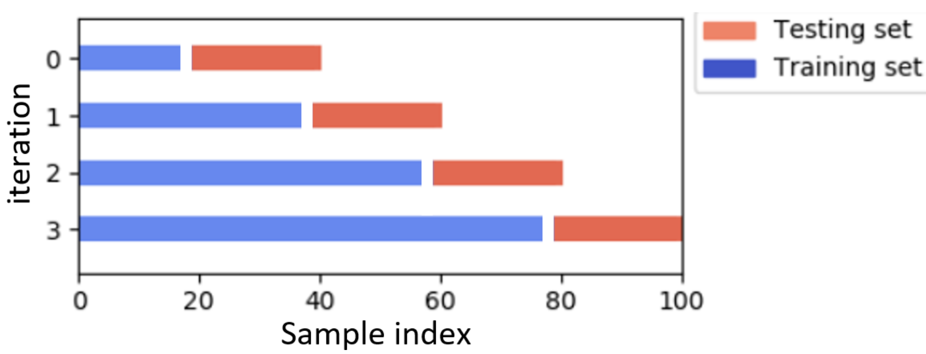


Figure 10: Embargoing for time-series cross validation. Source: scikit-learn documentation, author

3.2 Forecast Accuracy Measures

The best model is chosen on the basis of accuracy metrics. It is vital to choose such metrics which would provide accurate measure of model performance. It is likely to be the case that the dataset is unbalanced, meaning that there are more observations of one class than of other classes. In this case, HMM would suffer from frequency bias – it would be trained with the emphasis on observations belonging to a class that occurs most often. One of the ways to mitigate this problem is to choose appropriate accuracy measures that take data imbalance into account. We use precision and recall metrics, as well as F1 score, which is based on previous two. These metrics refer to positives and negatives, which are names for two classes. Usually, a class is named positive if some condition is present (such as illness, for instance). True and false indicate whether the prediction of class is correct.

3.2.1 Precision

Precision is the ratio of observations, correctly determined to belong to a certain class (true positives), to the total number of observations predicted to belong to this class. The latter also includes observations, incorrectly determined to belong to this class (false positives):

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \tag{3.1}$$

Higher value of precision indicates better model.

3.2.2 Recall

Recall is the ratio of correct predictions for the class (true positives) to the total number of observations belonging to this class. The latter also includes observations which were incorrectly predicted to belong to the other class (false negatives):

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \tag{3.2}$$

Higher value of recall indicates better model.

3.2.3 F score

F score combines precision and recall into a single metric, which allows to choose appropriate model for specific needs:

$$\text{F score} = (1+\beta^2)\frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}} \tag{3.3}$$

The higher F score the better the model is. β measures the trade-off between accuracy and precision. If $\beta < 1$, then more importance is given to precision, if $\beta > 1$ – to recall. For instance, if we treat loss from investment as negative class, we would like to minimize it. Thus, we calculate F score for negative class, taking $\beta > 1$, since we are more interested in the share of correct predictions out of all loss cases. For validation of our models, we take β=1, since we are more interested in the total predictive power of the models.

4 General Description of the Model

4.1 Parameters of HMM

As has been mentioned, HMM is based on the inclusion of unobservable states. Thus, to specify HHM, we need to specify 3 components: transition matrix (shows probability of movements between hidden states), emission matrix (shows probability of a particular emission in each state) and initial probabilities vector (shows probability that process starts in a particular hidden state). These are denoted as:

- A — transition matrix
- B — emission matrix
- π — initial probabilities vector

Additionally, we introduce the following notation:

- λ(A, B, π) — Hidden Markov Model
- N — number of states
- M — number of emissions
- a_{ij} — transition probability from state i to state j
- $b_i(O_t)$ — emission probability of observation O_t in state i
- π_i — i_{th} entry of initial state probabilities vector
- T — number of observations
- O — observation sequence, either discrete or continuous
- Q — state sequence, discrete only
- S_i — an example of a particular state

Transition matrix is based on the assumption that some, or all, states can progress into other states. The probability that state i progresses into state j is called transition probability and is denoted by a_{ij}. There can be states from which it is impossible to visit other states, in such cases they have zero transition probability. Transition matrix depicts all possible transitions between states by showing corresponding probabilities. Let A be N x N transition matrix. In rows it has states from which the process transits, in columns - states to which it transits. The sum of probabilities within a row should equal 1. An example of such matrix with 3 states is presented below. The states are denoted by w, x, and z. a_{xz} denotes probability of transition from state x to state z.

Table 7: Example of transition matrix with 3 states. Source: calculations of the author

	w	x	z
w	a_{ww}	a_{wx}	a_{wz}
x	a_{xw}	a_{xx}	a_{xz}
z	a_{zw}	a_{zx}	a_{zz}

There are no transitions when the process starts. Therefore, initial state probabilities vector is defined. It is the N x 1 vector showing probabilities that the process starts at each state, and it is denoted by π . The sum of probabilities of all initial states should equal 1.

There are several types (emissions) of observable phenomena, and each type has a certain probability to occur within a particular state. This probability depends only on the state that produced it, and not on any other state. Such distribution of each type within states is known as emission matrix. The number of states does not have to be equal to the number of emissions. Let B be N x M emission matrix for discrete observations. States are defined by rows, and emissions - by columns. The sum within each row should equal 1. Below is the example with emissions f and g and states w, x, z. $b_x(O_g)$ denotes the probability for observation of type g to occur in state x.

Table 8: Example of emission matrix with 3 states and 2 observation types for discrete data. Source: calculations of the author

	f	g
w	$b_w(O_f)$	$b_w(O_g)$
x	$b_x(O_f)$	$b_x(O_g)$
z	$b_z(O_f)$	$b_z(O_g)$

Thus, the model parameters consist of transition matrix A, emission matrix B and initial state probabilities vector π. These parameters are iteratively updated to optimal values until some stopping criterion is reached. Hidden Markov Model can solve 3 problems:

(Likelihood) Given the model λ(A, B, π) and an observation sequence $O_1, O_2, ..., O_n$, output the probability of seeing such a sequence P(O | λ). For this, Forward algorithm (section 5.1) is used.

(Decoding) Given the model λ(A, B, π) and an observation sequence $O_1, O_2, ..., O_n$, find the optimal state sequence consistent with $O_1, O_2, ..., O_n$. Viterby algorithm (section 4.4) is applied to this task.

(Learning) Given an observation sequence $O_1, O_2, ..., O_T$ of length T, initial transition matrix A, initial emission matrix B and initial state probabilities vector π, estimate the optimal parameters of the model A, B, π so that they are most consistent with data. This is done via Forward-Backward and Baum-Welch algorithms (sections 5.3 and 5.4).

4.2 Continuous Hidden Markov Model

In some cases, emissions are represented not as discrete symbols but as continuous data. It is possible to discretize it in order to make application of discrete HMM possible, but in most cases this severely decreases performance of the model. However, there exists a modification of HMM which makes it suitable for continuous data.

In the setup of the model, the changes occur in the emission matrix only. Now the emission probability at state S_i is represented not by several discrete probabilities but by one parametric probability density function (PDF) of some continuous distribution. Quite often a finite weighted mixture of PDFs of normal distributions (called Gaussian Mixture Model) is taken, as in [HNK07]. The parameters of PDFs can differ across states. When the model is trained, these parameters are estimated. In our model, we use single PDF of normal distribution. Hence, for each state emission probability is:

$$b_i(O_t) = N(O_t, \mu_i, \Sigma_i) \tag{4.1}$$

where μ_i is mean, Σ_i is covariance matrix. Thus, emission matrix takes the form:

Table 9: Example of emission matrix with 3 states for continuous data. Source: calculations of the author

	O_t
w	$N(O_t, \mu_w, \Sigma_w)$
x	$N(O_t, \mu_x, \Sigma_x)$
z	$N(O_t, \mu_z, \Sigma_z)$

When there are many states or complicated mixture model, then there are many parameters to estimate. In this case, larger datasets are required for the model to be trained. If there is not enough data, then the results of parameter estimation, especially covariance matrix, would be unreliable.

The algorithms for continuous HMM are generally the same as for discrete HMM. Only in Baum-Welch algorithm there are some slight changes in parameter estimation. There is also a difference in the definition of $b_i(O_t)$ – now it is probability of observation O_t in continuous distribution, corresponding to state i, not discrete probability.

4.3 Types of Hidden Markov Model

In publications on financial data analysis described in section 1, two main types of HMM have been used: right-left (used by [HN05]) and ergodic (used by [DG12]).

In ergodic model, every state can be reached from any other state in finite number of steps. Alternatively, this means there are no null transition probabilities.

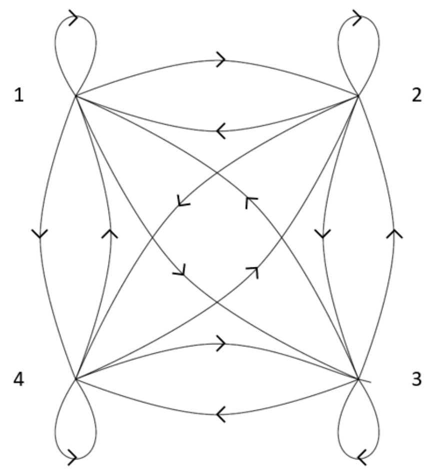


Figure 11: Example of ergodic HMM. Source: [Rab89]

Another type is left-right, or Bakis, model. This model does not allow transitions to the previous states, so the process evolves from left to right. This model can be used for processes that develop over time and attain some new characteristics. Obviously, there would be several zero transition probabilities. For example, if the process is currently at state i, it cannot return to previous state j, so transition probability a_{ij} is zero. Besides, initial state probabilities vector should have one non-zero entry only, since the order of states is predetermined and the process should begin in a particular state [Rab89].

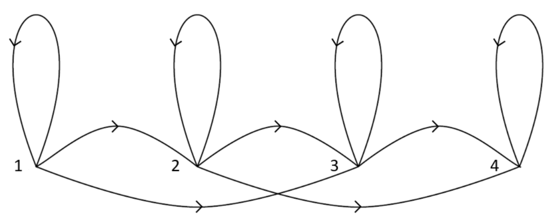


Figure 12: Example of left-right HMM. Source: [Rab89]

Sometimes bounds can be placed on the length of transitions to prevent significant and abrupt changes between states. For instance, in the figure above it is possible to make transitions at most over 2 states in one step (for instance, from state 2 to state 4).

In what follows we will use ergodic model due to the assumption that market conditions can change in any way due to the influence of different factors, and the state the market is currently in does not prevent other states from realization in future. For example, if today market is in favourable state for investors, tomorrow the state can continue to be favourable or can worsen as well.

4.4 Viterby Algorithm

Although not used in our model, Viterby algorithm is one of the main algorithms of HMM. It is used to find a state sequence, which is most consistent with observation sequence. As [Rab89] suggests, there are several possible criteria by which consistency can be defined. It can be that states should be optimal individually at every time moment, irrespectively of future and preceding states. However, this approach may produce erroneous results when HMM is not ergodic, i.e. there are some null transition probabilities. Since the optimal states are chosen individually, it can be that the two chosen subsequent states actually have null probability of transition between them, and thus the chosen sequence of states would be invalid. To account for this, it is required that optimality applies to the whole sequence, which means maximizing P(Q|O, λ) - probability of state sequence given the model and observations. The aim is to find $\delta_t(i)$ such that

$$\delta_t(i) = \max_{q_1,q_2,\ldots,q_{t-1}} P(q_1, q_2, ..., q_t = S_i, O_1, O_2, ..., O_t|\lambda) \tag{4.2}$$

Thus, $\delta_t(i)$ is the probability of seeing an observation sequence of length t ending in state S_i, maximized over all possible preceding states.

$$\delta_{t+1} = [\max_i \delta_t(i)a_{ij}]b_j(O_{t+1}) \tag{4.3}$$

4.3 takes the most probable state path to state j, at the same time accounting for the probability of observation in state j. The sequence of optimal states is retrieved via maximization of 4.3. Previous values of states that maximized 4.3 should be stored in an array, denoted by $\psi_t(j)$, for each state j . For the first observation:

$$\delta_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N \tag{4.4}$$

$$\psi_1(i) = 0 \tag{4.5}$$

According to 4.5, the first element of ψ is zero since there are no previous values across which maximization can be performed. For subsequent observations:

$$\delta_t(j) = [\max_{1\leq i\leq N} \delta_{t-1}(i)a_{ij}]b_j(O_t), \qquad 2 \leq t \leq T, 1 \leq j \leq N \tag{4.6}$$

$$\psi_t(j) = \arg\max_{1\leq i\leq N} \delta_{t-1}(i)a_{ij}, \qquad 2 \leq t \leq T, 1 \leq j \leq N \tag{4.7}$$

Using 4.6, we maximize the probability of being in state j and having observation O_t. By calculating $\delta_t(j)$, we find the most probable state path, leading to $\delta_t(j)$. For each state j at time t, the state at time $t-1$ which maximizes the probability of being in current state j, is chosen and written into array, according to 4.7. Thus, each of states j has its own corresponding path of optimal states, leading to it. At the last moment T there are N possible optimal state paths. To choose the most probable among them, the probability of seeing last observation should be maximized over all states. When the optimal δ_T is found, the sequence that corresponds to this δ_T, including the last state, is the optimal state path:

$$P = \max_{1\leq i\leq N} \delta_T(i) \tag{4.8}$$

$$q_T = \arg\max_{1\leq i\leq N} \delta_T(i) \tag{4.9}$$

Because the first element in ψ is zero, to retrieve optimal sequence of states at time t, the index in ψ should be moved one step further, i.e. to $t+1$:

$$q_t = \psi_{t+1}(q_{t+1}), \qquad t = T-1, T-2, ..., 1 \tag{4.10}$$

For sequence of N states and T observations, there are N^T possible state paths. Due to the fact that intermediate state paths are stored in array, time complexity of Viterby algorithm is reduced to $\theta(TN^2)$ ([CR11]) and space complexity – to $\theta(TN)$. Time complexity refers to the amount of time for the algorithm to run, and space complexity – to the amount of memory consumed. Forward (section 5.1) and Backward (section 5.3) algorithms have the same complexity.

Viterbi algorithm pseudocode

function Viterbi(observation sequence O of length T, state sequence Q of length N, $N \times N$ transition matrix A, $N \times M$ emission matrix B, $N \times 1$ initial state probabilities vector π)
returns optimal state path
initialize path probability matrix viterbi$[N, T]$
initialize optimal path matrix paths$[N, T]$
for each state i from 1 to N **do**
    viterbi[i,1] $= \pi[i] \cdot B[i, O_1]$
    paths[i,1]$= 0$
end
for each time t from 2 to T **do**
    for each state j from 1 to N **do**
        viterbi[j,t] $= \max\limits_{1 \leq i \leq N}$ viterbi$[i,\, t-1] \cdot A[i,j] \cdot B[j, O_t]$
        paths[j,t] $= \arg\max\limits_{1 \leq i \leq N}$ viterbi$[i,\, t-1] \cdot A[i,j]$
    end
end
bestprob $= \max\limits_{1 \leq i \leq N}$ viterby[i,T]
bestpath $= \arg\max\limits_{1 \leq i \leq N}$ viterby[i,T]
return the state path ending in bestpath

5 Our Model

5.1 Forward Algorithm

Forward algorithm, also known as filtering process, is used to compute probability of observing a particular sequence of observations given a model – to compute $P(O \mid \lambda)$. Let us define probability of a sequence of observations until time t, where state at time t is S_i:

$$\alpha_t(i) = P(O, q_t = S_i|\lambda) \tag{5.1}$$

The variable $\alpha_t(i)$ has different formulas, depending of whether the process is in $t = 1$ or $t > 1$. At t =1, the process has just started, so we use 5.2 to denote probability of seeing observation at start:

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N \tag{5.2}$$

For later observations, this probability is calculated as:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \qquad 1 \leq t \leq T-1, 1 \leq j \leq N \tag{5.3}$$

where the sum element represents the probability of being at state j at time $t+1$. This step is performed across all states. Finally, using 5.3, α_T is calculated for the final observation, and summed across all states in 5.4 to arrive at total probability of seeing such a sequence [Rab89].

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{5.4}$$

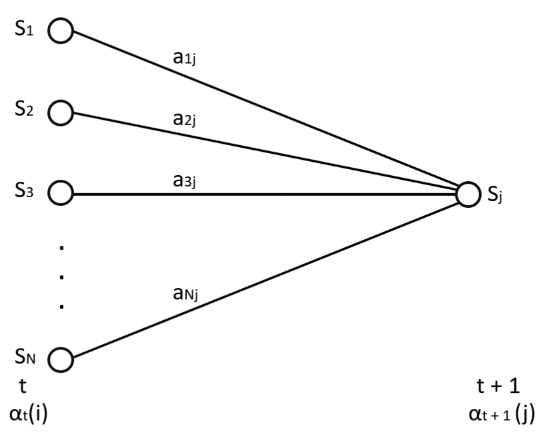


Figure 13: The sum in 5.3 represents transition from all N states at time t to state S_j at time $t+1$. Source: [Rab89]

As [JM09] note, Forward algorithm resembles Viterby algorithm, except that Forward algorithm takes the sum of previous probabilities, and Viterby takes maximum over them.

Forward algorithm pseudocode

function Forward(observation sequence O of length T, state sequence Q of length N, $N \times N$ transition matrix A, $N \times M$ emission matrix B, $N \times 1$ initial state probabilities vector π)
returns the probability of seeing a sequence of states forwseq, $N{\times}T$ matrix of forward probabilities forward
initialize probability matrix forward$[N, T]$
for each state i from 1 to N **do**
  forward[i,1] = $\pi[i]{\cdot}B[i, O_1]$
end
for each time t from 1 to T **do**
  for each state j from 1 to N **do**
   forward[j,$t+1$] = $\sum_{i=1}^{N}$ forward[i, t]$\cdot A[i, j]\cdot B[j, O_t]$
   end
end
forwseq = $\sum_{i=1}^{N}$ forward[i, T]
return forwseq, forward

5.2 Training the Model

The aim of training is to choose such parameters A, B and π of the model as to make it most consistent with observations – in other words, to maximize P(O|λ). Some of methods used by researchers to train HMM are Expectation-Maximization (EM) algorithm and gradient descent methods.

Gradient descent methods are based on the fact that negative gradient represents the direction of the steepest descent. Thus, if we formulate some loss function and move along the negative of its gradient, then the loss function will decrease at the fastest rate. To implement the most simple form of gradient descent, first some initial parameters are chosen, and then the gradient of loss function is computed with respect to these parameters. Learning rate, which is in fact some weight of the gradient, is also chosen. Then parameters are decreased by the gradient multiplied by learning rate. This process continues until some local minimum is reached.

EM is a generalization of Maximum Likelihood estimation. EM algorithms are used in probabilistic models, and can deal with incomplete data, which means there are missing observations or hidden variables. Probabilistic models are the ones that output distribution of the variable of interest, rather than make some certain numeric prediction.

EM and gradient descent algorithms would most likely find local maximum, not global one, and the result depends heavily on the values of initial parameters [CRR06]. [BD08] propose 2 solutions that can make this problem less severe. First, estimation can be performed with different initial values of parameters. Secondly, they note that it is useful to start with such parameters that break model symmetry.

In our work we use one of the algorithms of EM – Baum-Welch algorithm. To implement it, first we need to apply Forward-Backward algorithm, which calculates probabilities that are further used as inputs in Baum-Welch algorithm.

5.3 Forward-Backward Algorithm

Forward-Backward algorithm represents the first step in the learning task. Part of it (calculation of forward probabilities) has been already covered in the section on Forward algorithm, which evolves from start to end of the process. This section describes Backward algorithm, which is performed from end to the beginning of the process. Further, the results of the two algorithms are combined in Baum-Welch algorithm to arrive at smoother probabilities. First, we proceed to derivation of backward probability $\beta_t(i)$:

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, ..., O_T \mid q_t = S_i, \lambda) \tag{5.5}$$

Thus, $\beta_t(i)$ is the probability of having a sequence of observations, starting at $t+1$, given current state S_i and the model. For the last time moment, the probability is 1 across all states:

$$\beta_T(i) = 1, \qquad 1 \leq i \leq N \tag{5.6}$$

Then, $\beta_t(i)$ is calculated from the end of the process to its beginning.

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad t = T-1, T-2, ..., 1, \quad 1 \leq i \leq N \tag{5.7}$$

To arrive at time t to state S_i from the moment $t+1$, all states at $t+1$ should be considered. That is why the process involves summation. For each state S_j at $t+1$, it should be accounted for the probability that the next observation O_{t+1} happens at S_j ($b_j(O_{t+1})$), probability of transition from S_i to S_j (a_{ij}) and probability of remaining observation sequence, starting from O_{t+2} and given S_j ($\beta_{t+1}(j)$).

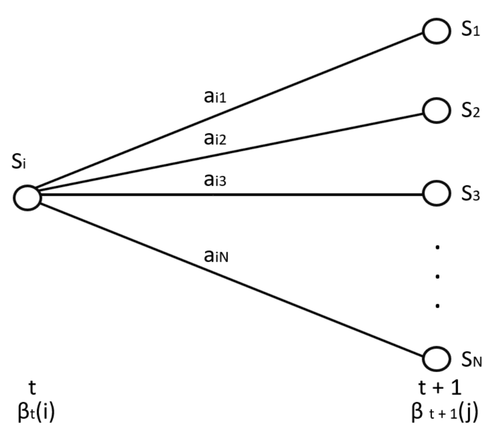


Figure 14: Illustration of Backward algorithm. All states S_j should be considered to arrive at S_i. Source: [Rab89]

$\beta_t(i)$ and $\alpha_t(i)$ can become very small and this may introduce computational problems. To overcome them, [GKMS10] recommend to normalize $\beta_t(i)$ and $\alpha_t(i)$, using the same normalization constant, so that for each time t the sum of $\beta_t(i)$ and $\alpha_t(i)$ across all states equals one:

$$\sum_{i=1}^{N} \beta_t(i) = 1 \tag{5.8}$$

$$\sum_{i=1}^{N} \alpha_t(i) = 1 \tag{5.9}$$

Since we cannot observe and control intermediate calculations, we follow this recommendation to avoid possible computational problems in case some probabilities become too low.

Next, define $\gamma_t(i)$ - probability to be in time t at state S_i, given observation sequence and the model:

$$\gamma_t(i) = P(q_t = S_i \mid O, \lambda) \tag{5.10}$$

In fact, $\gamma_t(i)$ is a smoothed probability, as it combines both forward and backward probabilities:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{5.11}$$

$\alpha_t(i)$ is the probability to be at state S_i, given observation sequence up to the moment t. $\beta_t(i)$ is the probability to be in state S_i, given observations after time t. Thus, taken together, $\alpha_t(i)$ and $\beta_t(i)$ account for the entire sequence of observations. Numerator in 5.11 denotes probability of being in state S_i at time t, having entire sequence of observations and the model. Denominator in 5.11 sums this probability across states. Dividing by it allows to normalize $\gamma_t(i)$:

$$\sum_{i=1}^{N} \gamma_t(i) = 1 \tag{5.12}$$

Forward-Backward algorithm pseudocode

function Forward-Backward(observation sequence O of length T, state sequence Q of length N, $N \times N$ transition matrix A, $N \times M$ emission matrix B, forward probabilities forward)
returns $N \times T$ matrix of smoothed probabilities smoothprob, $N \times T$ matrix of backward probabilities backward
initialize probability matrix backward$[N, T]$
initialize probability matrix smoothprob$[N, T]$
for each state i from 1 to N **do**
    backward[i,T] = 1
end
for each time t from $T - 1$ to 1 **do**
    for each state i from 1 to N **do**
        backward[i,t] $= \sum_{j=1}^{N} A[i, j] \cdot B[j, O_{t+1}] \cdot$backward$[j,\ t+1]$
    end
end
for each time t from 1 to T **do**
    for each state i from 1 to N **do**
        smoothprob[i,t] $= \dfrac{\text{forward}[i,t] \cdot \text{backward}[i,t]}{\sum_{i=1}^{N} \text{forward}[i,t] \cdot \text{backward}[i,t]}$
    end
end
return smoothprob, backward

5.4 Baum-Welch Algorithm

This is final step in the training task. Previously, we have calculated forward and backward probabilities, $\alpha_t(i)$ and $\beta_t(i)$, and smoothed probabilities $\gamma_t(i)$. These probabilities are used in Baum-Welch algorithm to estimate parameters of the model – transition matrix A, emission matrix B and initial state probabilities vector π.

First, we define $\xi_t(i, j)$ as the probability to be in state S_i at time t, and at state S_j at time $t + 1$, given the model and observations.

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda) \tag{5.13}$$

$\xi_t(i, j)$ can be calculated as:

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O \mid \lambda)} = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \tag{5.14}$$

$\alpha_t(i)$ denotes the probability of being in state S_i at time t, given past t observations. It is calculated with Forward Algorithm. $\beta_{t+1}(j)$ denotes the probability of being in state S_j at $t + 1$, given the remaining future part of the sequence not used by $\alpha_t(i)$. It is calculated with Backward algorithm. Between S_i and S_j, there is transition probability a_{ij}. Also, we should account for observation sequence, so $b_j(O_{t+1})$ is introduced. Thus, numerator of 5.14 is

$P(q_t = S_i, q_{t+1} = S_j, O \mid \lambda)$. This expression is divided by the sum P(O | λ), to arrive at 5.13. This is in accordance with the following law:

$$P(A \mid B, C) = \frac{P(A, B \mid C)}{P(B \mid C)} \tag{5.15}$$

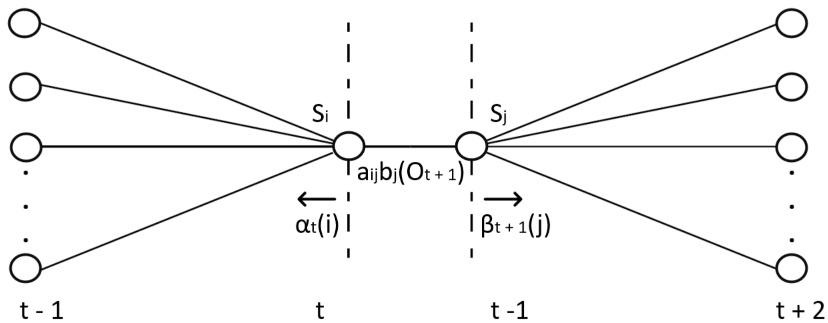


Figure 15: Illustration of relation between S_i and S_j. Source: [Rab89]

Having calculated $\gamma_t(i)$ and $\xi_t(i,j)$, we completed expectation step of EM algorithm. Now we can relate $\gamma_t(i)$ and $\xi_t(i,j)$. As $\xi_t(i,j)$ is the probability to be in state S_i at time t, and state S_j at time $t+1$, given model and observations, then by summing over j we arrive at $\gamma_t(i)$. This is because we account for all possible transitions from S_i:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{5.16}$$

Alternatively, summation of $\gamma_t(i)$ over t represents expected number of times S_i is visited. Then, summation of $\gamma_t(i)$ from $t = 1$ to $t = T - 1$ is the expected number of transitions from S_i. Similarly, for $\xi_t(i,j)$ summation from $t = 1$ to $t = T - 1$ is the expected number of transitions from S_i to S_j:

$$\sum_{t=1}^{T-1} \gamma_t(i) - \text{expected number of transitions from } S_i \tag{5.17}$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) - \text{expected number of transitions from } S_i \text{ to } S_j \tag{5.18}$$

Finally, using 5.17 and 5.18, we can re-estimate parameters of the model π, A and B (maximization step of EM algorithm). π_i is a probability to be in state S_i at time $t = 1$, given the model and observations:

$$\pi_i = \gamma_1(i) \tag{5.19}$$

a_{ij} is calculated as the ratio of expected number of transitions from S_i to S_j, divided by expected number of transitions from S_i:

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{5.20}$$

$b_j(k)$ is the probability to have observation of type k in state S_j. It is calculated as the ratio of expected number of times in S_j, simultaneously having observation of type k, to total expected number in state S_j:

$$b_j(k) = \frac{\sum_{t=1, s.t. O_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{5.21}$$

It is satisfied that:

$$\sum_{i=1}^{N} \pi_i = 1 \tag{5.22}$$

$$\sum_{j=1}^{N} a_{ij} = 1, \quad 1 \leq i \leq N \tag{5.23}$$

$$\sum_{k=1}^{M} b_i(k) = 1, \quad 1 \leq j \leq N \tag{5.24}$$

This procedure is repeated until some stopping criterion is reached. On each iteration, new parameters A, B and π are used as initial input parameters of the algorithm. With each iteration, the model becomes more consistent with the data: P(O | λ^{new}) > P(O | λ^{old}).

Baum-Welch algorithm pseudocode

function Baum-Welch(observation sequence O of length T, state sequence Q of length N, $N \times N$ transition matrix A, $N \times M$ emission matrix B, forward probabilities forward, backward probabilities backward, smoothed probabilities smoothprob)
returns parameters of HMM: transition matrix A, emission matrix B, initial probabilities vector π
iterate until convergence
  for each state i from 1 to N **do**
    initialize probability matrix state$_i[N, T]$
      for each time t from 1 to T **do**
        for each state j from 1 to N **do**

$$\text{state}_i[j,t] = \frac{\text{forward}[i,t] \cdot A[i,j] \cdot B[j, O_{t+1}] \cdot \text{backward}[j, t+1]}{\sum_{i=1}^{N} \sum_{j=1}^{N} \text{forward}[i,t] \cdot A[i,j] \cdot B[j, O_{t+1}] \cdot \text{backward}[j, t+1]}$$

        end
      end
    $\pi_i = \text{smoothprob}[i,1]$
    for each state j from 1 to N **do**

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \text{state}_i[j,t]}{\sum_{t=1}^{T-1} \text{smoothprob}[i,t]}$$

$$b_j(k) = \frac{\sum_{t=1, s.t. O_t = v_k}^{T} \text{smoothprob}[j,t]}{\sum_{t=1}^{T-1} \text{smoothprob}[j,t]}$$

    end
  end
return A, B, π

Baum-Welch algorithm has time complexity $\theta(N(1 + T(M + N)))$ and space complexity $\theta(N(N + M + TN))$, which is rather low due to the fact that intermediate calculations are stored.

5.5 Baum-Welch: Continuous Model

5.5.1 Single Observation Sequence

For continuous model, only emission matrix is estimated differently, since there is a need to estimate means μ and covariance matrices Σ of PDFs of normal distributions. Estimation of initial state probabilities vector and transition matrix is similar to discrete case. For emission matrix which consists of PDFs of single normal distributions, estimation formulas for state j are:

$$\mu_j = \frac{\sum_{t=1}^{T} \gamma_t(j) \cdot O_t}{\sum_{t=1}^{T} \gamma_t(j)} \tag{5.25}$$

$$\Sigma_j = \frac{\sum_{t=1}^{T} \gamma_t(j)(O_t - \mu_j)(O_t - \mu_j)^T}{\sum_{t=1}^{T} \gamma_t(j)} \tag{5.26}$$

We see that 5.25 is the weighted average of smoothed probabilities $\gamma_t(j)$, weighted by observations at time t. 5.26 is also weighted average of smoothed probabilities $\gamma_t(j)$, but it is weighted by variance $(O_t - \mu_j)(O_t - \mu_j)^T$. Thus, estimation of both μ_j and Σ_j takes into account the magnitude of observations.

5.5.2 Multiple Observation Sequences

Denote O a sequence of R observation sequences:

$$O = \{O^{(1)}, O^{(2)}, ..., O^{(R)}\} \tag{5.27}$$

Observation sequences are independent and can have different lengths. The rth sequence of length T_r is:

$$O^{(r)} = \{o_1^{(r)}, o_2^{(r)}, ..., o_{T_r}^{(r)}\} \tag{5.28}$$

For parameter estimation, for each sequence $O^{(r)}$ $\alpha_t(i)$, $\beta_t(i)$, $\gamma_t(i)$ and $\xi_t(i,j)$ are calculated. Then, the results are combined across individual sequence parameters to arrive at new estimates:

$$\pi_i = \frac{1}{R}\sum_{r=1}^{R} \gamma_1^{(r)}(i) \tag{5.29}$$

$$a_{ij} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r-1} \xi_t^{(r)}(i,j)}{\sum_{r=1}^{R}\sum_{t=1}^{T_r-1} \gamma_t^{(r)}(i)} \tag{5.30}$$

$$\mu_j = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r} \gamma_t^{(r)}(j) \cdot O_t^{(r)}}{\sum_{r=1}^{R}\sum_{t=1}^{T_r} \gamma_t^{(r)}(j)} \tag{5.31}$$

$$\Sigma_j = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r} \gamma_t^{(r)}(j)(O_t^{(r)} - \mu_j)(O_t^{(r)} - \mu_j)^T}{\sum_{r=1}^{R}\sum_{t=1}^{T_r} \gamma_t^{(r)}(j)} \tag{5.32}$$

6 Empirical Results

We develop our model with Python 3.7.6 using *pomegranate* v0.13.3 library, which implements HMM. The code can be found on Github [3].

We train 24 pairs of models overall, which is 8 pairs for each of the daily, hourly and minute datasets. For each frequency, we test two modifications of HMM: univariate (consisting of one input sequence) and multivariate (consisting of several input sequences). We also test models using stationary and non-stationary data for all datasets. Finally, for each dataset, we train HHMs on sets of two lengths: 1858 (the length of daily dataset) and 800 (chosen arbitrarily) observations. In total, there are 8 combinations of data characteristics for each frequency.

In training models we follow [BGO08]. We train 2 separate HMMs, one for increasing trend and one for decreasing. We define increasing (decreasing) trend as a sequence of observations ending with τ consecutive close price increases (decreases). Thus, each HMM for positive (negative) trend is trained on sequences ending with τ close price increases (decreases). We choose the number of states N and τ empirically for each model. The number of states N varies from 3 to 8, and the length of trend τ – from 3 to 5. There is insufficient amount of data for training and testing if $\tau>5$ is taken. We consider $\tau<3$ not enough to be considered a trend, and $N<3$ not enough to be able to explain complex market conditions. We randomly generate initial state probabilities vector π, emission matrix B and transition matrix A. In B we use normal distribution to represent emission probabilities in each state. The parameters are generated so as to be consistent in magnitude with observations. μ is generated as random share from 0.5 to 1 of the actual mean of the time series, Σ as random share from 0.5 to 1 of the actual standard deviation. In case of multivariate model, mean and standard deviation of close prices or returns are taken, depending on the target variable. For each combination of N and τ, parameters are generated 30 times, resulting in 540 models to choose from for each frequency for each of 8 combinations of data characteristics.

The datasets are unbalanced, meaning that there is generally a different number of sequences with positive and negative trend. This means that models are trained and tested with different amount of data. We tackle this problem through the use of F score for model selection. We use 70% of input data for training, 1% for embargoing and 29% for testing, where input data is the set of sequences with positive or negative trend retrieved from 1858 or 800 observations.

Some authors ([DA17]) train HMM on one sequence only, usually close prices, while others use several features for training ([FW17], [DAP13]). We aim to find if decrease in the number of features has any negative effect on model performance. If it is not so, then the complexity of computations could be reduced by eliminating some variables.

We note that none of the works mentioned in section 1 check data for stationarity. Yet, at least as some researchers believe, stationarity is a prerequisite for data predictability. We aim to explore whether stationarity is an essential data characteristic for successful implementation of Hidden Markov Model. Thus, we train models using 2 datasets, one stationary and one non-stationary. For univariate model, stationary dataset consists of stock returns data and non-stationary – of close price data. For multivariate model stationary dataset consists of first

[3] https://github.com/anastasia-yaschenko/Hidden-Markov-Model.git

difference (meaning the difference between observations at time t and time $t-1$) of open, high, low and close data, non-stationary - of open, high, low and close data without differencing. For daily prices, there is also a variable *overnight* for daily dataset, which is used with stationary data only due to its small magnitude. The division of datasets is also supported by the fact that for multivariate sequences, estimation of means μ and covariance matrices Σ of normal distribution PDFs depends on observations. Thus, it is desirable that all sequences are of approximately the same magnitude, in order to provide the same amount of information to the model. For instance, if we used open, high, low prices and returns, this would mean that estimated μ and Σ would contain little information from returns, since they are significantly smaller in magnitude than prices. We aim to avoid this situation and to give each sequence approximately the same weight.

[BGO08] mention that there is hardly any sense to train model on data covering more than 5 weeks, since any market patterns are destroyed during longer time periods. They note that performance of the model generally decreases with the increase in the training sequence length. However, as [Nil05] reports, HMM require large amount of data to be trained. We do not follow recommendation of [BGO08] to use sequences of no more than 5 weeks at maximum for training, since this is insufficient for training continuous HMM with many parameters. Thus, we also explore this issue of pattern destruction by comparing performance of models using 1858 observations and 800 observations as input data. We take the same length of data across all frequencies to to allow for greater comparability of the models, as we aim to learn whether HMMs perform better on particular frequency.

Positive and negative trend HMMs are trained separately, using Baum-Welch algorithm. Then, the best pair of positive and negative trend HMMs is chosen on the basis of F score. The procedure is as follows. Each pair of HMMs, where both models have the same N and τ, is tested on positive and negative CV datasets. Positive dataset consists of sequences with positive trend, with τ last observations (representing consecutive price increases) cut off. Thus, there is no indication that there is a positive trend, apart from some patterns in the data which HMM should detect. Each of the two HMMs predicts the probability of each sequence from positive dataset with Forward algorithm. Sequence is labelled as having positive trend if HMM trained on positive data outputs probability higher than HMM trained on negative data. Then, F score is calculated for positive CV dataset. The procedure is repeated for the same pair of models with negative CV dataset, and F score is calculated. Then, two F scores, one for positive and one for negative datasets, are summed. The procedure is repeated for each pair of HMMs. Finally, the pair with highest total F score is chosen. The drawback of this method is that pairs are formed of models with the same N and τ. It could be the case that optimal N and τ are different for positive and negative trend. However, testing each combination of negative and positive trend HMMs would be computationally infeasible, as there are $540 \times 540 = 291{,}600$ such combinations. When choosing the best model, we focus on the overall performance of the pair. This decision rule can be adjusted according to the needs. For instance, if it is required to minimize losses, then the pair with highest F score on negative dataset should be chosen.

In almost all cases the model outputs insignificant probability of observing a sequence (less than 1%), however, it does not have detrimental effect on the performance of classification. The models manage to assign correct classes to sequences with rather high total F score. The explanation can be that the patterns in the prices are so complex that the model cannot assign them to any class with high certainty, based on absolute values of probabilities. However, HMM is able to recognize that the sequence of prices has higher probability to belong to one class rather than the other. Thus, for the task of return prediction with HMMs, the models are more useful when used with relative probabilities, which can be compared, than with absolute values of probabilities.

We have also trained several discrete univariate models. We have applied discretization to return and close price sequences by dividing the data into several bins, according to magnitude. However, these models showed low total F score (less than 0.5) on training dataset. We also noticed that several discrete HMMs that corresponded to positive (negative) trend have overfitted and for all test sequences predicted probability higher than HMM of negative (positive) trend.

Below we present F scores, number of states N and length of trend τ for best pairs of continuous univariate HMMs for 3 datasets – daily, hourly and minute. The estimated parameters for the corresponding pairs can be found in Annex 4.

Table 10: F scores for pairs of univariate HMMs, daily dataset. F score positive and negative refer to positive and negative CV datasets respectively. Source: calculations of the author

Sequence	Dataset size	Total F score	F score negative	F score positive	N	τ
Close prices	1858	1.5714	1	0.5714	8	5
Returns	1858	1.5	0.8333	0.6667	7	3
Close prices	800	2	1	1	5	5
Returns	800	2	1	1	4	5

Table 11: F scores for pairs of univariate HMMs, hourly dataset. F score positive and negative refer to positive and negative CV datasets respectively. Source: calculations of the author

Sequence	Dataset size	Total F score	F score negative	F score positive	N	τ
Close prices	1858	1.6	1	0.6	6	5
Returns	1858	1.5	0.8	0.7	6	5
Close prices	800	1.4667	0.6667	0.8	3	5
Returns	800	1.8	0.8	1	6	4

Table 12: F scores for pairs of univariate HMMs, minute dataset. F score positive and negative refer to positive and negative CV datasets respectively. Source: calculations of the author

Sequence	Dataset size	Total F score	F score negative	F score positive	N	τ
Close prices	1858	1.4	0.8	0.6	5	5
Returns	1858	1.8	0.8	1	6	5
Close prices	800	1.4	0.4	1	4	5
Returns	800	2	1	1	4	5

For multivariate models we also train HMMs with Baum-Welch algorithm. For non-stationary data, the sequences are open, close, high and low prices, and the prediction is made with Forward algorithm for close price only. For stationary data, the training sequences are *overnight* and the differences at time t and $t-1$ for each of open, close, high and low price data. Differenced close prices are returns for which predictions are made with Forward algorithm. The reason that only one sequence is used in Forward algorithm is that there is no multivariate extension for this algorithm. However, the data of all training sequences has been already incorporated into the model, as the predictions are made with calibrated parameters that account for information from these sequences.

Table 13: F scores for pairs of multivariate HMMs, daily dataset. F score positive and negative refer to positive and negative CV datasets respectively. Source: calculations of the author

Sequence	Dataset size	Total F score	F score negative	F score positive	N	τ
Close prices	1858	1.5714	1	0.5714	3	5
Returns	1858	1.5714	1	0.5714	6	5
Close prices	800	2	1	1	4	5
Returns	800	1.2917	0.6667	0.625	8	4

Table 14: F scores for pairs of multivariate HMMs, hourly dataset. F score positive and negative refer to positive and negative CV datasets respectively. Source: calculations of the author

Sequence	Dataset size	Total F score	F score negative	F score positive	N	τ
Close prices	1858	1.6	0.6	1	3	5
Returns	1858	1.8571	1	0.8571	8	5
Close prices	800	1.475	0.6	0.875	4	4
Returns	800	2	1	1	8	5

Table 15: F scores for pairs of multivariate HMMs, minute dataset. F score positive and negative refer to positive and negative CV datasets respectively. Source: calculations of the author

Sequence	Dataset size	Total F score	F score negative	F score positive	N	τ
Close prices	1858	1.3	0.9	0.4	4	5
Returns	1858	1.3939	0.6667	0.72721	7	5
Close prices	800	1.4556	0.5556	0.9	8	3
Returns	800	1.625	1	0.625	8	3

Most of our models choose τ=5 as optimal trend length, in line with the results of [BGO08]. With this value of τ, the trend is more well-defined, which means that pattern in prices becomes easier to recognise by HMM. The number of states N varies across the models, however, in most cases $N > 4$ is chosen, which is a larger number than used in works of [BGO08] and [HN05]. This indicates that the optimal number of states should be chosen for each combination of data characteristics individually, which confirms the finding of [FW17]. Such large number of states may indicate that AMZN stock is highly liquid, and so orders involving it are placed in a variety of market conditions.

In many cases, the model classifies with F score = 1 for either positive or negative subsets, or both. As we have chosen optimal models based on overall F score, in some instances this has resulted in bias in favour of a particular trend. This issue could be mitigated if more balanced datasets would be available, which consist of roughly the same number of positive and negative sequences both for training and testing. Also, other pairs of models could be chosen on the basis not of highest total F score, but on the basis of more balanced accuracy of classification.

We note that HMMs generally classify with F scores higher than 0.5 for both positive and negative trend data on all frequencies, with the exception of 2 predictions on minute dataset, for univariate HHM trained on 800 observations of close prices and multivariate HMM trained on 1858 observations of close prices.

We see that generally, for minute dataset for both multivariate and univariate HMMs, total F scores are higher for models trained on differenced data. This indicates that stationary data is preferred for higher frequencies. The results for daily and hourly datasets are mixed, so probably the issue of stationarity is not an important data characteristic for these frequencies. This can be explained by the fact that due to additional noise in minute data, it requires more defined statistical characteristics of data in order to classify with high F score.

Increase in the number of training sequences has generally positive effect on F score for daily and hourly data – the accuracy measure either increases or at least remains unchanged. Yet, the performance of HMM on minute dataset decreases, even for stationary data. This could be attributed to the fact that additional sequences, even though they provide more information, introduce more noise to the minute data, and the effect from it outweighs the gain from more information. This is supported by the finding that as frequency of data increases, introduction of more features increases the number of states. Comparing univariate and multivariate models for the same length of observations and stationarity type, we see that as more sequences are introduced, for daily dataset the number of states is reduced in 3 cases, for each of hourly and minute datasets it is increased in 3 cases. Thus, for daily data new information seems to assist in explaining existing patterns, and so the number of states falls. For hourly and minute datasets, it introduces more patterns and noise, so larger number of states is needed to describe them.

The effect of time period length, over which the observations are taken, varies across HMM types. For univariate HMM, the reduction in observation length has positive effect for return prediction over all datasets. For close price forecast the evidence is mixed, since the increase in F score occurs only on daily dataset. For multivariate model, shorter training sequences result in increased F score for predicted returns for hourly and minute datasets, for predicted close price – for daily and minute datasets. Thus, we conclude there is evidence that in some cases decrease in time period over which HMMs are trained has positive effect on accuracy of trend detection. This is consistent with Hurst exponent result, showing that price patterns are less obvious over longer periods.

Finally, we conclude that HMM can be applied to data of all frequencies, as it classifies trend with high enough overall F score in all cases. The univariate model performance is slightly better on the daily dataset than on others. Multivariate HMM performs worst on the minute dataset, probably due to the effect from additional data, described above. However, with appropriate data specifications, the performance of models can be improved.

We note that although we have not placed any strict restriction on the type of HMM (we have chosen the most general ergodic model), estimated initial state probabilities vectors

π and transition matrices A may indicate that a more suitable HMM type is left-right, as suggested by [HNK07]. In all cases, π vector clearly indicates the starting state by assigning unit probability to it. Starting probabilities of all other states are zero. In transition matrices, quite a large number of transition probabilities are zero. Therefore, additional bounds on initial and transition probabilities can be placed.

We also note that in several transition matrices, such as 96 and 72, columns corresponding to several states are zero, meaning that transition to these states are blocked. However, transitions from these states are possible. Starting probabilities in these states are also zero. This means that optimal model includes inaccessible states, and this is the case for univariate HMMs trained on hour and minute close price datasets with 1858 observations. It could be that over longer time intervals of higher frequency data, these additional states help to explain variation in prices. This remains an issue for additional research.

Conclusion

In this work, we applied HMM for datasets of AMZN stock prices for three frequencies: daily, hourly and minute. We also investigated whether such characteristics of data as stationarity of time series, length of observation sequence and number of features can improve model performance. We find that HMM generally predicts with high F score on all datasets, yet there are some specifications of data that can boost model performance on a particular frequency. Minute frequency requires stationary data, and it shows better performance when trained with one feature only, measured over shorter time interval. For daily and hourly datasets, stationarity of data does not alter performance significantly. However, larger number of features enhances classification on these frequencies. What concerns the length of training data, the findings are mixed.

Further work

Additional research in the following areas may improve performance of HMM on data of different frequencies. First, another distribution for emissions could be used, such as Gaussian Mixture Model, which consists of convex combination of several Gaussian distributions. Returns and close prices may not always follow single normal distribution, for instance, they could be bimodal (with two local maxima), which can be modelled with Gaussian Mixture. It is recommended that along with it, a better method for the generation of initial parameters is tested. For instance, in case of Gaussian Mixture, it is preferable to use clustering algorithms, such as k-means or Self-Organizing Feature Map (SOFM). Since we have compared the pairs of models with the same number of states and τ, a more efficient technique for comparison of pairs with different number of states and trend length could be investigated. This could produce better results, since negative and positive trends may have different statistical characteristics. Finally, left-right HMM could be used instead of ergodic one, considering the estimated initial state probabilities vectors and transition matrices of our models.

.

Bibliography

[AK15] M. Awad and R. Khanna. *Efficient Learning Machines*. Apress, Berkeley, 2015.

[BD08] S. Batzoglou and C. B. Do. What is the Expectation Maximization Algorithm? *Nature Biotechnology*, 26(8), 897–899, 2008.

[BGO08] M. Bicego, E. Grosso, and E. Otranto. A Hidden Markov Model Approach to Classify and Predict the Sign of Financial Local Trends . In *Structural, Syntactic, and Statistical Pattern Recognition*, volume 5342, Berlin, Heidelberg, 852-861, 2008.

[Bis11] P. Biswal. Spectral and Wavelet Analysis of Stock Price Behaviour: an Overview of the Studies. 2011.

[CR11] S. Chatterjee and S. Russell. A Temporally Abstracted Viterbi Algorithm. In *In Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, Arlington, Virginia, USA, 96–104, 2011.

[CRR06] H. D. Chiang, C. K. Reddy, and B. Rajaratnam. Stability Region Based Expectation Maximization for Model-Based Clustering. In *Proceedings - IEEE International Conference on Data Mining*, Hong Kong, 522–531, 2006.

[DA17] L. De Andrade. Stock Market Index Trading Algorithm Using Discrete Hidden Markov Models and Technical Analysis. 2017.

[DAP13] L. De Angelis and L. J. Paas. A Dynamic Analysis of Stock Markets Using a Hidden Markov Model. *Journal of Applied Statistics*, 40(8), 1682-1700, 2013.

[DG12] B. Dhingra and A. Gupta. Stock Market Prediction Using Hidden Markov Models. In *2012 Students Conference on Engineering and Systems*, Allahabad, Uttar Pradesh, 1-4, 2012.

[Fam65] E. F. Fama. The Behavior of Stock Market Prices. *The Journal of Business*, 38(1), 34–105, 1965.

[FW17] Z. Fu and Q. Wu. Quantitative Trading Strategy of Market States Prediction Based on HMM. In *Proceedings of the 2017 2nd International Conference on Materials Science, Machinery and Energy Engineering*, Dalian, China, 1309-1319, 2017.

[FYL+19] B. Fang, P. S. Yu, Y. Li, S. Wang, and X. Zhang. Enhancing Stock Market Prediction with Extended Coupled Hidden Markov Model over Multi-sourced Data. *Knowledge and Information Systems*, 61(2), 1071–1090, 2019.

[GKMS10] E. Granger, W. Khreich, A. Miri, and R. Sabourin. On the Memory Complexity of the Forward-Backward Algorithm. *Pattern Recognition Letters*, 31(2), 91–99, 2010.

[GMM99] R. Gimeno, B. Manchado, and R. Mínguez. Stationarity Tests for Financial Time Series. *Physica A: Statistical Mechanics and its Applications*, 269(1), 72–78, 1999.

[HN05] M. R. Hassan and B. Nath. Stock Market Forecasting Using Hidden Markov Model: A New Approach. In *Proceedings - 5th International Conference on Intelligent Systems Design and Applications*, Warsaw, 192–196, 2005.

[HNK07] M. R. Hassan, B. Nath, and M. Kirley. A Fusion Model of HMM , ANN and GA for Stock Market Forecasting. *Expert Systems With Applications*, 33(1), 171–180, 2007.

[Hyn18] A. Hyndman. *Forecasting: Principles and Practice, 2nd edition.* OTexts, Melbourne, Australia, 2018.

[Jef98] B.A. Jeff. A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. *International Computer Science Institute*, 510(4), 1998.

[JM09] D. Jurafsky and J. H. Martin. *Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Pearson Prentice Hall, New Jersey, 2009.

[JTJ16] A. Joseph, C. Turner, and R. Jeremiah. Comparative Analyses of Stock Returns Properties and Predictability. *Procedia Computer Science*, 95, 272–280, 2016.

[KNU13] G. Kavitha, D. Nagarajan, and A. Udhayakumar. Stock Market Trend Analysis Using Hidden Markov Models. 2013.

[Ngu18] N. Nguyen. Hidden Markov Model for Stock Trading. *International Journal of Financial Studies*, 36(6), 2018.

[Nil05] M. Nilsson. First Order Hidden Markov Model – Theory and Implementation Issues. 2005.

[Rab89] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2), 257-286, 1989.

[SS10] M. Stamp and P. Stavroulakis. *Handbook of Information and Communication Security, 1st edition.* Springer Publishing Company Incorporated, New York, 2010.

[Tuy18] L. Tuyen. A Higher Order Markov Model for Time Series Forecasting. *International Journal of Applied Mathematics and Statistics*, 57, 1-18, 2018.

Annexes

Annex 1. Descriptive Statistics

Annex

Table 16: Sample statistics on AMZN stock prices (USA dollars) for the period 13.04.2020 – 22.05.2020, daily dataset.

Statistics	Open price	Low price	High price	Close price
Std Deviation	89.1148	84.0580	67.7543	66.6192
Mean	2358.3107	2334.0143	2399.5322	2371.8238

Table 17: Sample statistics on AMZN stock prices (USA dollars) for the period 13.04.2020 – 22.05.2020, hourly dataset.

Statistics	Open price	Low price	High price	Close price
Std Deviation	72.7308	72.4161	69.5943	69.5947
Mean	2367.5242	2358.8447	2378.2781	2369.2207

Table 18: Sample statistics on AMZN stock prices (USA dollars) for the period 13.04.2020 – 22.05.2020, minute dataset.

Statistics	Open price	Low price	High price	Close price
Std Deviation	71.8717	71.7699	71.9426	71.8372
Mean	2367.4243	2368.0713	2366.8060	2367.4283

Table 19: Sample statistics on AMZN stock returns (USA dollars) for the period 13.04.2020 – 22.05.2020, daily dataset.

Statistics	Open price	Low price	High price	Close price	Overnight
Std Deviation	60.8933	52.5185	53.7603	61.0383	35.5158
Mean	13.9407	14.2390	14.3241	13.5131	4.0701

Table 20: Sample statistics on AMZN stock returns (USA dollars) for the period 13.04.2020 – 22.05.2020, hourly dataset.

Statistics	Open price	Low price	High price	Close price
Std Deviation	19.6679	20.4574	19.4838	20.5820
Mean	1.6928	1.7045	1.6992	1.6965

Table 21: Sample statistics on AMZN stock returns (USA dollars) for the period 13.04.2020 – 22.05.2020, minute dataset.

Statistics	Open price	Low price	High price	Close price
Std Deviation	3.0649	3.0495	2.8502	3.0124
Mean	0.0435	0.0438	0.0436	0.0436

Annex 2. Hurst Exponent

Table 22: Hurst exponent, non-differenced time series, daily dataset.

	Open price	Low price	High price	Close price
Exponent value (first 800 obs.)	0.9969	0.9982	0.9969	1
Result (first 800 obs.)	persistent	persistent	persistent	persistent
Exponent value (last 800 obs.)	1	1	1	1
Result (800 obs.)	persistent	persistent	persistent	persistent

Table 23: Hurst exponent, non-differenced time series, hourly dataset.

	Open price	Low price	High price	Close price
Exponent value (first 800 obs.)	1	1	1	1
Result (first 800 obs.)	persistent	persistent	persistent	persistent
Exponent value (last 800 obs.)	0.9885	0.9840	0.9875	0.9845
Result (800 obs.)	persistent	persistent	persistent	persistent

Table 24: Hurst exponent, non-differenced time series, minute dataset.

	Open price	Low price	High price	Close price
Exponent value (first 800 obs.)	1	1	1	
Result (first 800 obs.)	persistent	persistent	persistent	persistent
Exponent value (last 800 obs.)	0.9912	0.9909	0.9908	0.9915
Result (800 obs.)	persistent	persistent	persistent	persistent

Table 25: Hurst exponent, differenced time series, daily dataset.

	Open price	Low price	High price	Close price	Overnight
Exponent value (first 800 obs.)	0.5490	0.5512	0.5608	0.5608	0.5754
Result (first 800 obs.)	persistent	persistent	persistent	persistent	persistent
Exponent value (last 800 obs.)	0.5502	0.5576	0.5531	0.5528	0.5500
Result (800 obs.)	persistent	persistent	persistent	persistent	persistent

Table 26: Hurst exponent, differenced time series, hourly dataset.

	Open price	Low price	High price	Close price
Exponent value (first 800 obs.)	0.6116	0.6137	0.6124	0.6131
Result (first 800 obs.)	persistent	persistent	persistent	persistent
Exponent value (last 800 obs.)	0.5690	0.5686	0.5786	0.5683
Result (800 obs.)	persistent	persistent	persistent	persistent

Table 27: Hurst exponent, differenced time series, minute dataset.

	Open price	Low price	High price	Close price
Exponent value (first 800 obs.)	0.6032	0.6252	0.6117	0.5736
Result (first 800 obs.)	persistent	persistent	persistent	persistent
Exponent value (last 800 obs.)	0.5304	0.5288	0.5317	0.5300
Result (800 obs.)	persistent	persistent	persistent	persistent

Annex 3. Results of Stationarity Tests

Table 28: Results of stationarity tests, non-differenced time series, daily dataset.

	Open price	Low price	High price	Close price
ADF p-value	0.9953	0.9936	0.9923	0.9948
KPSS p-value	0.01	0.01	0.01	0.01
Conclusion	non-stationary	non-stationary	non-stationary	non-stationary

Table 29: Results of stationarity tests, non-differenced time series, hourly dataset.

	Open price	Low price	High price	Close price
ADF p-value	0.8268	0.8115	0.8099	0.8331
KPSS p-value	0.1	0.1	0.1	0.1
Conclusion	non-stationary trend stationary	non-stationary trend stationary	non-stationary trend stationary	non-stationary trend stationary

Table 30: Results of stationarity tests, non-differenced time series, minute dataset.

	Open price	Low price	High price	Close price
ADF p-value	0.5819	0.6153	0.5838	0.5547
KPSS p-value	0.01	0.01	0.01	0.01
Conclusion	non-stationary	non-stationary	non-stationary	non-stationary

Table 31: Results of stationarity tests, differenced time series, daily dataset.

	Open price	Low price	High price	Close price	Overnight
ADF p-value	0.0	0.0	0.0	0.0	0.0
KPSS p-value	0.1	0.1	0.1	0.1	0.05973
Conclusion	stationary trend-stationary	stationary trend-stationary	stationary trend-stationary	stationary trend-stationary	stationary trend-stationary

Table 32: Results of stationarity tests, differenced time series, hourly dataset.

	Open price	Low price	High price	Close price
ADF p-value	0.0	0.0	0.0	0.0
KPSS p-value	0.1	0.1	0.1	0.1
Conclusion	stationary trend-stationary	stationary trend-stationary	stationary trend-stationary	stationary trend-stationary

Table 33: Results of stationarity tests, differenced time series, minute dataset.

	Open price	Low price	High price	Close price
ADF p-value	0.0	0.0	0.0	0.0
KPSS p-value	0.1	0.1	0.1	0.1
Conclusion	stationary trend-stationary	stationary trend-stationary	stationary trend-stationary	stationary trend-stationary

Annex 4. Model Parameters

Annex

Univariate HHM for Daily Dataset

Univariate HHM for daily close price data 800 obs

Table 34: Initial state probabilities vector of positive HMM

s0	0.0
s1	0.0
s2	1.0
s3	0.0
s4	0.0

Table 35: Emission matrix of positive HMM

s0	N(321.531823, 9.947829)
s1	N(294.294682, 7.730715)
s2	N(262.435861, 5.589138)
s3	N(372.635889, 18.467408)
s4	N(275.312396, 3.139033)

Table 36: Transition matrix of positive HMM

	s0	s1	s2	s3	s4
s0	0.961806	0.014553	0.000000	0.023641	0.000000
s1	0.048278	0.951722	0.000000	0.000000	0.000000
s2	0.000000	0.000000	0.965396	0.000000	0.034604
s3	0.017998	0.000000	0.000000	0.982002	0.000000
s4	0.000000	0.042378	0.045494	0.000000	0.912128

Table 37: Initial state probabilities vector of negative HMM

s0	0.0
s1	1.0
s2	0.0
s3	0.0
s4	0.0

Table 38: Emission matrix of negative HMM

s0	N(427.935949, 4.108464)
s1	N(292.543877, 20.978462)
s2	N(439.87686, 3.551162)
s3	N(372.65094, 4.142112)
s4	N(383.782851, 3.984235)

Table 39: Transition matrix of negative HMM

	s0	s1	s2	s3	s4
s0	0.970839	0.000000	0.029161	0.000000	0.000000
s1	0.000000	0.990289	0.000000	0.009711	0.000000
s2	0.232448	0.000000	0.767552	0.000000	0.000000
s3	0.000000	0.000000	0.000000	0.936245	0.063755
s4	0.000000	0.000000	0.041375	0.045285	0.913339

Univariate HHM for daily return data 800 obs

Table 40: Initial state probabilities vector of positive HMM

s0	0.0
s1	1.0
s2	0.0
s3	0.0

Table 41: Emission matrix of positive HMM

s0	N(-0.434444, 1.196246)
s1	N(5.681427, 1.664249)
s2	N(-2.726704, 18.554374)
s3	N(-0.569682, 4.592474)

Table 42: Transition matrix of positive HMM

	s0	s1	s2	s3
s0	0.109361	0.007761	0.079559	0.803319
s1	0.832727	0.077127	0.090147	0.000000
s2	0.000000	0.069668	0.325424	0.604909
s3	0.000000	0.165369	0.037281	0.797350

Table 43: Initial state probabilities vector of negative HMM

s0	0.0
s1	0.0
s2	1.0
s3	0.0

Table 44: Emission matrix of negative HMM

s0	N(48.930008, 6.180009)
s1	N(5.373551, 3.828571)
s2	N(-1.304473, 5.003264)
s3	N(-2.078312, 3.112277)

Table 45: Transition matrix of negative HMM

	s0	s1	s2	s3
s0	0.000000	0.50675	0.00000	0.493250
s1	0.041058	0.00000	0.00000	0.958942
s2	0.000000	1.00000	0.00000	0.000000
s3	0.000000	0.00000	0.44748	0.552520

Univariate HHM for daily close price data 1858 obs

Table 46: Initial state probabilities vector of positive HMM

s0	0.0
s1	1.0
s2	0.0
s3	0.0
s4	0.0
s5	0.0
s6	0.0
s7	0.0

Table 47: Emission matrix of positive HMM

s0	N(522.646989, 20.12367)
s1	N(266.49427, 7.167302)
s2	N(587.081847, 23.867609)
s3	N(310.998289, 15.556745)
s4	N(696.351583, 32.23982)
s5	N(859.411949, 89.55986)
s6	N(429.043962, 5.397253)
s7	N(373.796621, 17.188847)

Table 48: Transition matrix of positive HMM

	s0	s1	s2	s3	s4	s5	s6	s7
s0	0.959938	0.000000	0.040062	0.000000	0.000000	0.000000	0.000000	0.000000
s1	0.000000	0.991872	0.000000	0.008128	0.000000	0.000000	0.000000	0.000000
s2	0.014198	0.000000	0.957764	0.000000	0.028037	0.000000	0.000000	0.000000
s3	0.000000	0.000000	0.000000	0.986660	0.000000	0.000000	0.000000	0.013340
s4	0.000000	0.000000	0.009537	0.000000	0.980941	0.009522	0.000000	0.000000
s5	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000
s6	0.024393	0.000000	0.000000	0.000000	0.000000	0.000000	0.975607	0.000000
s7	0.000000	0.000000	0.000000	0.014558	0.000000	0.000000	0.007211	0.978231

Table 49: Initial state probabilities vector of negative HMM

s0	0.0
s1	1.0
s2	0.0
s3	0.0
s4	0.0
s5	0.0
s6	0.0
s7	0.0

Table 50: Emission matrix of negative HMM

s0	N(1919.912488, 29.114933)
s1	N(340.153487, 29.308962)
s2	N(1665.09297, 60.889295)
s3	N(1996.772355, 13.225601)
s4	N(509.962993, 61.809025)
s5	N(2002.682252, 0.309901)
s6	N(1984.295232, 26.708466)
s7	N(1498.521059, 60.511379)

Table 51: Transition matrix of negative HMM

	s0	s1	s2	s3	s4	s5	s6	s7
s0	0.773497	0.000000	0.075879	0.150625	0.000000	0.000000	0.000000	0.000000
s1	0.000000	0.989001	0.000000	0.000000	0.010999	0.000000	0.000000	0.000000
s2	0.000000	0.000000	0.969714	0.000000	0.000000	0.000000	0.000000	0.030286
s3	0.000000	0.000000	0.000000	0.000000	0.000000	0.137735	0.862265	0.000000
s4	0.000000	0.000000	0.000000	0.000000	0.980799	0.019201	0.000000	0.000000
s5	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000
s6	0.501278	0.000000	0.000000	0.498722	0.000000	0.000000	0.000000	0.000000
s7	0.000000	0.000000	0.124041	0.000000	0.000000	0.000000	0.000000	0.875959

Univariate HHM for daily return data 1858 obs

Table 52: Initial state probabilities vector of positive HMM

s0	0.0
s1	0.0
s2	1.0
s3	0.0
s4	0.0
s5	0.0
s6	0.0
s7	0.0

Table 53: Emission matrix of positive HMM

s0	N(-7.710271, 34.394733)
s1	N(-0.607454, 9.06479)
s2	N(4.597057, 2.53043)
s3	N(5.233083, 8.952792)
s4	N(15.592106, 7.916606)
s5	N(0.466149, 14.016022)
s6	N(-0.216793, 2.84429)
s7	N(-5.316526, 0.489314)

Table 54: Transition matrix of positive HMM

	s0	s1	s2	s3	s4	s5	s6	s7
s0	0.346239	0.171881	0.000000	0.161241	0.246889	0.000000	0.073749	0.000000
s1	0.000000	0.163200	0.836800	0.000000	0.000000	0.000000	0.000000	0.000000
s2	0.000000	0.040693	0.076927	0.037132	0.000000	0.098456	0.726943	0.019849
s3	0.009810	0.001127	0.000000	0.967337	0.021726	0.000000	0.000000	0.000000
s4	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
s5	0.000000	0.000000	0.000000	0.000000	0.142357	0.000000	0.000000	0.857643
s6	0.000000	0.198215	0.283472	0.000000	0.000000	0.000000	0.500031	0.018282
s7	0.000000	0.000000	0.856914	0.000000	0.143086	0.000000	0.000000	0.000000

Table 55: Initial state probabilities vector of negative HMM

s0	0.0
s1	0.0
s2	0.0
s3	0.0
s4	0.0
s5	1.0
s6	0.0
s7	0.0

Table 56: Emission matrix of negative HMM

s0	N(-2.613208, 2.216413)
s1	N(-1.703417, 1.548146)
s2	N(-6.638792, 2.383039)
s3	N(1.04695, 5.659635)
s4	N(10.320656, 2.891253)
s5	N(3.904317, 2.449364)
s6	N(-6.625937, 6.543973)
s7	N(0.878381, 23.947764)

Table 57: Transition matrix of negative HMM

	s0	s1	s2	s3	s4	s5	s6	s7
s0	0.000000	0.000000	0.332060	0.000000	0.149844	0.000000	0.518096	0.000000
s1	0.066743	0.333519	0.151053	0.000000	0.000000	0.448685	0.000000	0.000000
s2	0.000000	0.424128	0.000000	0.112134	0.079458	0.321273	0.000000	0.063007
s3	0.000000	0.833337	0.166663	0.000000	0.000000	0.000000	0.000000	0.000000
s4	0.497786	0.000000	0.152938	0.000000	0.134425	0.000000	0.000000	0.214851
s5	0.000000	0.314177	0.101687	0.584136	0.000000	0.000000	0.000000	0.000000
s6	0.067162	0.000000	0.000000	0.000000	0.000000	0.000000	0.463285	0.469554
s7	0.344344	0.000000	0.000000	0.000000	0.083507	0.000000	0.439587	0.132562

Univariate HHM for Hourly Dataset

Univariate HHM for hourly close price data 800 obs

Table 58: Initial state probabilities vector of positive HMM

s0	1.0
s1	0.0
s2	0.0

Table 59: Emission matrix of positive HMM

s0	N(1937.097098, 81.338004)
s1	N(2361.368898, 53.259493)
s2	N(1868.718212, 20.041134)

Table 60: Transition matrix of positive HMM

	s0	s1	s2
s0	0.978640	0.006884	0.014476
s1	0.000000	1.000000	0.000000
s2	0.028136	0.000000	0.971864

Table 61: Initial state probabilities vector of negative HMM

s0	0.0
s1	1.0
s2	0.0

Table 62: Emission matrix of negative HMM

s0	N(1040.528284, 62.967453)
s1	N(1875.427391, 27.914099)
s2	N(2060.331544, 98.585243)

Table 63: Transition matrix of negative HMM

	s0	s1	s2
s0	0.0	0.000000	1.000000
s1	0.0	0.975264	0.024736
s2	0.0	0.020487	0.979513

Univariate HHM for hourly return data 800 obs

Table 64: Initial state probabilities vector of positive HMM

s0	0.0
s1	0.0
s2	0.0
s3	1.0
s4	0.0
s5	0.0

Table 65: Emission matrix of positive HMM

s0	N(2.049434, 15.365045)
s1	N(11.31645, 28.353881)
s2	N(-9.371325, 0.642473)
s3	N(0.789532, 1.527008)
s4	N(-4.856323, 1.350253)
s5	N(6.22666, 3.531429)

Table 66: Transition matrix of positive HMM

	s0	s1	s2	s3	s4	s5
s0	0.000000	0.000000	0.063057	0.373794	0.158670	0.404479
s1	0.170066	0.722358	0.107577	0.000000	0.000000	0.000000
s2	0.000000	0.000000	0.190732	0.239874	0.241586	0.327808
s3	0.456449	0.079301	0.005104	0.459146	0.000000	0.000000
s4	0.875563	0.000000	0.000000	0.124437	0.000000	0.000000
s5	0.103133	0.000000	0.000000	0.365027	0.115092	0.416749

Table 67: Initial state probabilities vector of negative HMM

s0	0.0
s1	1.0
s2	0.0
s3	0.0
s4	0.0
s5	0.0

Table 68: Emission matrix of negative HMM

s0	N(-1.994825, 5.246659)
s1	N(-0.606353, 0.568397)
s2	N(-2.910542, 2.163434)
s3	N(-2.3887, 17.328639)
s4	N(2.0553, 13.91503)
s5	N(-0.529126, 51.760438)

Table 69: Transition matrix of negative HMM

	s0	s1	s2	s3	s4	s5
s0	0.229536	0.466905	0.000000	0.000000	0.303559	0.000000
s1	0.594999	0.043646	0.000000	0.000000	0.326676	0.034679
s2	0.000000	0.000000	0.321511	0.070316	0.000000	0.608172
s3	0.000000	0.000000	0.178867	0.558855	0.000000	0.262278
s4	0.683477	0.316523	0.000000	0.000000	0.000000	0.000000
s5	0.000000	0.000000	0.107170	0.892830	0.000000	0.000000

Univariate HHM for hourly close price data 1858 obs

Table 70: Initial state probabilities vector of positive HMM

s0	0.0
s1	0.0
s2	0.0
s3	0.0
s4	0.0
s5	1.0

Table 71: Emission matrix of positive HMM

s0	N(1095.101752, 68.589639)
s1	N(1040.346665, 71.817386)
s2	N(930.83649, 75.045134)
s3	N(1204.611928, 57.292522)
s4	N(1095.101752, 49.223152)
s5	N(1825.169587, 80.629625)

Table 72: Transition matrix of positive HMM

	s0	s1	s2	s3	s4	s5
s0	0.0	0.0	0.0	0.0	0.0	1.0
s1	0.0	0.0	0.0	0.0	0.0	1.0
s2	0.0	0.0	0.0	0.0	0.0	1.0
s3	0.0	0.0	0.0	0.0	0.0	1.0
s4	0.0	0.0	0.0	0.0	0.0	1.0
s5	0.0	0.0	0.0	0.0	0.0	1.0

Table 73: Initial state probabilities vector of negative HMM

	6
s0	0.0
s1	1.0
s2	0.0
s3	0.0
s4	0.0
s5	0.0

Table 74: Emission matrix of negative HMM

s0	N(1749.136953, 24.404791)
s1	N(1878.533468, 17.606755)
s2	N(1273.153902, 53.784004)
s3	N(1143.993361, 81.998563)
s4	N(2046.728005, 50.360312)
s5	N(1819.104768, 14.80882)

Table 75: Transition matrix of negative HMM

	s0	s1	s2	s3	s4	s5
s0	0.962576	0.006016	0.0	0.0	0.000000	0.031408
s1	0.000000	0.972649	0.0	0.0	0.009094	0.018257
s2	1.000000	0.000000	0.0	0.0	0.000000	0.000000
s3	1.000000	0.000000	0.0	0.0	0.000000	0.000000
s4	0.000000	0.014706	0.0	0.0	0.985294	0.000000
s5	0.030729	0.014876	0.0	0.0	0.000000	0.954395

Univariate HHM for hourly return data 1858 obs

Table 76: Initial state probabilities vector of positive HMM

s0	0.0
s1	0.0
s2	1.0
s3	0.0
s4	0.0
s5	0.0

Table 77: Emission matrix of positive HMM

s0	N(2.962299, 4.709231)
s1	N(1.265018, 2.59677)
s2	N(0.619566, 5.770988)
s3	N(-0.557929, 4.473684)
s4	N(1.118485, 16.856689)
s5	N(0.216595, 0.791372)

Table 78: Transition matrix of positive HMM

	s0	s1	s2	s3	s4	s5
s0	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000
s1	0.327064	0.000000	0.000000	0.000000	0.000000	0.672936
s2	0.000000	0.000000	0.000000	0.987884	0.012116	0.000000
s3	0.000000	0.525293	0.414919	0.000000	0.059788	0.000000
s4	0.000000	0.000000	0.879945	0.008074	0.063823	0.048158
s5	0.050186	0.238762	0.000000	0.021904	0.689147	0.000000

Table 79: Initial state probabilities vector of negative HMM

s0	0.0
s1	0.0
s2	0.0
s3	1.0
s4	0.0
s5	0.0

Table 80: Emission matrix of negative HMM

s0	N(0.007182, 13.635381)
s1	N(-1.794817, 3.032998)
s2	N(-1.507028, 8.167702)
s3	N(-0.455021, 0.961028)
s4	N(3.781993, 2.63922)
s5	N(4.040608, 49.045624)

Table 81: Transition matrix of negative HMM

	s0	s1	s2	s3	s4	s5
s0	0.279145	0.378159	0.000000	0.256061	0.086635	0.000000
s1	0.170127	0.509470	0.000000	0.320402	0.000000	0.000000
s2	0.000000	0.000000	0.766621	0.000000	0.233379	0.000000
s3	0.445557	0.213742	0.000000	0.064091	0.098399	0.178212
s4	0.111965	0.264215	0.000000	0.328822	0.261954	0.033044
s5	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000

Univariate HHM for Minute Dataset

Univariate HHM for minute close price data 800 obs

Table 82: Initial state probabilities vector of positive HMM

s0	1.0
s1	0.0
s2	0.0
s3	0.0

Table 83: Emission matrix of positive HMM

s0	N(2489.630707, 2.56554)
s1	N(2456.744618, 6.179991)
s2	N(1654.159852, 14.21268)
s3	N(1604.781946, 12.373392)

Table 84: Transition matrix of positive HMM

	s0	s1	s2	s3
s0	0.989130	0.010870	0.000000	0.000000
s1	0.000000	1.000000	0.000000	0.000000
s2	0.452552	0.163871	0.303133	0.080444
s3	0.311840	0.488718	0.098169	0.101273

Table 85: Initial state probabilities vector of negative HMM

S0	0.0
S1	1.0
S2	0.0
S3	0.0

Table 86: Emission matrix of negative HMM

s0	N(1263.003902, 14.39992)
s1	N(2476.478238, 19.692793)
s2	N(2005.947373, 17.753326)
s3	N(1981.182591, 11.835551)

Table 87: Transition matrix of negative HMM

	s0	s1	s2	s3
s0	0.19233	0.03355	0.710565	0.063555
s1	0.00000	1.00000	0.000000	0.000000
s2	0.00000	1.00000	0.000000	0.000000
s3	0.00000	1.00000	0.000000	0.000000

Univariate HHM for minute return data 800 obs

Table 88: Initial state probabilities vector of positive HMM

s0	0.0
s1	0.0
s2	0.0
s3	1.0

Table 89: Emission matrix of positive HMM

s0	N(-2.887066, 0.605295)
s1	N(-0.281029, 3.396971)
s2	N(0.195978, 0.8519)
s3	N(0.724942, 1.536395)

Table 90: Transition matrix of positive HMM

	s0	s1	s2	s3
s0	0.000000	0.234969	0.096461	0.668570
s1	0.236875	0.763125	0.000000	0.000000
s2	0.089484	0.017397	0.893120	0.000000
s3	0.000000	0.000000	0.183041	0.816959

Table 91: Initial state probabilities vector of negative HMM

s0	1.0
s1	0.0
s2	0.0
s3	0.0

Table 92: Emission matrix of negative HMM

s0	N(0.096521, 0.561535)
s1	N(0.705236, 1.164605)
s2	N(-0.660392, 1.463034)
s3	N(-0.320625, 2.879737)

Table 93: Transition matrix of negative HMM

	s0	s1	s2	s3
s0	0.880639	0.000000	0.000000	0.119361
s1	0.000000	0.000035	0.999965	0.000000
s2	0.038633	0.587781	0.373586	0.000000
s3	0.000000	0.029443	0.000000	0.970557

Univariate HHM for minute close price data 1858 obs

Table 94: Initial state probabilities vector of positive HMM

s0	0.0
s1	1.0
s2	0.0
s3	0.0
s4	0.0

Table 95: Emission matrix of positive HMM

s0	N(1337.397628, 31.886785)
s1	N(2431.632052, 42.479047)
s2	N(1604.877154, 27.210057)
s3	N(1215.816026, 35.7132)
s4	N(1313.081308, 27.210057)

Table 96: Transition matrix of positive HMM

	s0	s1	s2	s3	s4
s0	0.0	1.0	0.0	0.0	0.0
s1	0.0	1.0	0.0	0.0	0.0
s2	0.0	1.0	0.0	0.0	0.0
s3	0.0	1.0	0.0	0.0	0.0
s4	0.0	1.0	0.0	0.0	0.0

Table 97: Initial state probabilities vector of negative HMM

s0	1.0
s1	0.0
s2	0.0
s3	0.0
s4	0.0

Table 98: Emission matrix of negative HMM

s0	N(2449.641968, 44.299768)
s1	N(1249.317403, 28.371867)
s2	N(1665.756538, 37.681386)
s3	N(1273.813823, 24.825384)
s4	N(1788.238636, 31.031729)

Table 99: Transition matrix of negative HMM

	s0	s1	s2	s3	s4
s0	1.00000	0.000000	0.000000	0.000000	0.000000
s1	1.00000	0.000000	0.000000	0.000000	0.000000
s2	1.00000	0.000000	0.000000	0.000000	0.000000
s3	0.06381	0.208246	0.302755	0.095595	0.329595
s4	1.00000	0.000000	0.000000	0.000000	0.000000

Univariate HHM for minute return data 1858 obs

Table 100: Initial state probabilities vector of positive HMM

s0	0.0
s1	1.0
s2	0.0
s3	0.0
s4	0.0
s5	0.0

Table 101: Emission matrix of positive HMM

s0	N(1.20054, 2.711246)
s1	N(0.445102, 0.753723)
s2	N(4.117337, 16.146786)
s3	N(-3.761182, 0.621197)
s4	N(0.247771, 1.522118)
s5	N(0.114576, 1.260616)

Table 102: Transition matrix of positive HMM

	s0	s1	s2	s3	s4	s5
s0	0.937982	0.000000	0.00000	0.062018	0.000000	0.000000
s1	0.000000	0.115222	0.00000	0.028093	0.000000	0.856685
s2	0.678392	0.000000	0.00000	0.000000	0.000000	0.321608
s3	0.000000	0.000000	0.00000	0.000000	1.000000	0.000000
s4	0.001848	0.000000	0.00000	0.018564	0.943669	0.035919
s5	0.008613	0.962847	0.02854	0.000000	0.000000	0.000000

Table 103: Initial state probabilities vector of negative HMM

s0	0.0
s1	0.0
s2	0.0
s3	0.0
s4	0.0
s5	1.0

Table 104: Emission matrix of negative HMM

s0	N(0.213141, 0.331566)
s1	N(-0.545955, 2.326013)
s2	N(-1.217029, 0.74823)
s3	N(-0.510056, 0.701559)
s4	N(1.059182, 1.010476)
s5	N(-0.337532, 4.183732)

Table 105: Transition matrix of negative HMM

	s0	s1	s2	s3	s4	s5
s0	0.041225	0.150811	0.000000	0.807963	0.000000	0.000000
s1	0.000000	0.835060	0.000000	0.000000	0.164940	0.000000
s2	0.000000	0.174622	0.575592	0.158935	0.000000	0.090851
s3	0.000000	0.000000	0.135228	0.334089	0.530683	0.000000
s4	0.275391	0.000000	0.000000	0.468227	0.256382	0.000000
s5	0.000000	0.000000	0.112070	0.000000	0.000000	0.887930

Multivariate HHM for Daily Dataset

Multivariate HHM for daily close price data 800 obs

Table 106: Initial state probabilities vector of positive HMM

s0	1.0
s1	0.0
s2	0.0
s3	0.0

Table 107: Emission matrix of positive HMM

s0	N(266.428136, 7.825681)
s1	N(359.864696, 8.642749)
s2	N(393.448979, 7.23366)
s3	N(312.131826, 15.876584)

Table 108: Transition matrix of positive HMM

	s0	s1	s2	s3
s0	0.988719	0.000000	0.000000	0.011281
s1	0.000000	0.956922	0.014351	0.028727
s2	0.000000	0.022623	0.977377	0.000000
s3	0.000000	0.014920	0.000000	0.985080

Table 109: Initial state probabilities vector of negative HMM

s0	0.0
s1	1.0
s2	0.0
s3	0.0

Table 110: Emission matrix of negative HMM

s0	N(305.345097, 10.828037)
s1	N(264.417253, 3.925005)
s2	N(400.807383, 27.347138)
s3	N(271.469255, 4.985109)

Table 111: Transition matrix of negative HMM

	s0	s1	s2	s3
s0	0.985714	0.000000	0.014286	0.000000
s1	0.044669	0.901367	0.000000	0.053964
s2	0.000000	0.000000	1.000000	0.000000
s3	0.000000	0.095782	0.000000	0.904218

Multivariate HHM for daily return data 800 obs

Table 112: Initial state probabilities vector of positive HMM

s0	0.0
s1	0.0
s2	1.0
s3	0.0
s4	0.0
s5	0.0
s6	0.0
s7	0.0

Table 113: Emission matrix of positive HMM

s0	N(-3.072723, 1.158957)
s1	N(0.147222, 0.394519)
s2	N(3.328597, 2.165013)
s3	N(0.19864, 1.492403)
s4	N(1.413741, 0.599488)
s5	N(-0.642593, 0.957489)
s6	N(-5.291973, 22.614647)
s7	N(-1.718251, 0.372483)

Table 114: Transition matrix of positive HMM

	s0	s1	s2	s3	s4	s5	s6	s7
s0	0.000000	0.250345	0.284002	0.000000	0.162937	0.000000	0.097208	0.205508
s1	0.108597	0.137619	0.114083	0.205579	0.331933	0.000000	0.000000	0.102190
s2	0.000000	0.536208	0.000000	0.000000	0.262720	0.000000	0.000000	0.201072
s3	0.000000	0.116733	0.000000	0.000000	0.745565	0.000000	0.137701	0.000000
s4	0.126983	0.278660	0.100879	0.000000	0.360020	0.000000	0.000000	0.133458
s5	0.000000	0.050115	0.622709	0.000000	0.000000	0.327176	0.000000	0.000000
s6	0.121517	0.131350	0.000000	0.000000	0.000000	0.515023	0.232110	0.000000
s7	0.000000	0.000000	0.000000	0.525427	0.236040	0.238533	0.000000	0.000000

Table 115: Initial state probabilities vector of negative HMM

s0	0.0
s1	0.0
s2	0.0
s3	0.0
s4	0.0
s5	0.0
s6	0.0
s7	1.0

Table 116: Emission matrix of negative HMM

s0	N(-0.907975, 1.240765)
s1	N(5.15064, 22.006968)
s2	N(2.601527, 1.892601)
s3	N(-2.724483, 1.358808)
s4	N(-0.609725, 2.417479)
s5	N(2.114246, 0.547413)
s6	N(3.729587, 1.694622)
s7	N(-0.073725, 0.942595)

Table 117: Transition matrix of negative HMM

	s0	s1	s2	s3	s4	s5	s6	s7
s0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
s1	0.314673	0.000000	0.000000	0.685327	0.000000	0.000000	0.000000	0.000000
s2	0.000000	0.745603	0.254397	0.000000	0.000000	0.000000	0.000000	0.000000
s3	0.215979	0.000000	0.000000	0.438620	0.000000	0.000000	0.345401	0.000000
s4	0.000000	0.000000	0.000000	0.000000	0.000000	0.150164	0.000000	0.849836
s5	0.535551	0.000000	0.000000	0.464449	0.000000	0.000000	0.000000	0.000000
s6	0.000000	0.000000	0.613664	0.000000	0.000000	0.386336	0.000000	0.000000
s7	0.470344	0.000000	0.000000	0.000000	0.395751	0.133906	0.000000	0.000000

Multivariate HHM for daily close price data 1858 obs

Table 118: Initial state probabilities vector of positive HMM

s0	0.0
s1	0.0
s2	1.0

Table 119: Emission matrix of positive HMM

s0	N(834.702248, 96.73279)
s1	N(589.594879, 59.129892)
s2	N(325.970502, 50.256375)

Table 120: Transition matrix of positive HMM

	s0	s1	s2
s0	1.000000	0.000000	0.000000
s1	0.005834	0.994166	0.000000
s2	0.000000	0.001891	0.998109

Table 121: Initial state probabilities vector of negative HMM

s0	1.0
s1	0.0
s2	0.0

Table 122: Emission matrix of negative HMM

s0	N(339.800917, 29.223715)
s1	N(1707.473019, 160.248588)
s2	N(509.789839, 62.970029)

Table 123: Transition matrix of negative HMM

	s0	s1	s2
s0	0.989001	0.000000	0.010999
s1	0.000000	1.000000	0.000000
s2	0.000000	0.019199	0.980801

Multivariate HHM for daily return data 1858 obs

Table 124: Initial state probabilities vector of positive HMM

s0	1.0
s1	0.0
s2	0.0
s3	0.0
s4	0.0
s5	0.0

Table 125: Emission matrix of positive HMM

s0	N(2.833896, 2.549636)
s1	N(0.852475, 0.913467)
s2	N(1.204854, 5.96861)
s3	N(1.072747, 2.675156)
s4	N(-0.341128, 28.458764)
s5	N(-0.854543, 1.563173)

Table 126: Transition matrix of positive HMM

	s0	s1	s2	s3	s4	s5
s0	0.068802	0.364972	0.000000	0.000000	0.058454	0.507772
s1	0.069503	0.464976	0.000000	0.000000	0.011161	0.454360
s2	0.000000	0.000000	0.834809	0.090845	0.074346	0.000000
s3	0.000000	0.000000	0.054200	0.929404	0.016396	0.000000
s4	0.000000	0.000000	0.405495	0.153242	0.258314	0.182948
s5	0.147365	0.449514	0.000000	0.000000	0.012865	0.390256

Table 127: Initial state probabilities vector of negative HMM

s0	0.0
s1	0.0
s2	0.0
s3	1.0
s4	0.0
s5	0.0

Table 128: Emission matrix of negative HMM

s0	N(11.06095, 4.318375)
s1	N(0.492753, 3.823511)
s2	N(-8.630525, 7.483881)
s3	N(-0.532159, 1.687951)
s4	N(-15.168848, 77.274175)
s5	N(3.450927, 27.225945)

Table 129: Transition matrix of negative HMM

	s0	s1	s2	s3	s4	s5
s0	0.308207	0.000000	0.691793	0.000000	0.000000	0.000000
s1	0.020194	0.864196	0.000000	0.056613	0.058998	0.000000
s2	0.720762	0.000000	0.231508	0.000000	0.000000	0.047730
s3	0.000000	0.047823	0.000000	0.952177	0.000000	0.000000
s4	0.000000	0.529389	0.143828	0.000000	0.000000	0.326783
s5	0.000000	0.000000	0.000000	0.000000	0.041776	0.958224

Multivariate HHM for Hourly Dataset

Multivariate HHM for hourly close price data 800 obs

Table 130: Initial state probabilities vector of positive HMM

s0	0.0
s1	1.0
s2	0.0
s3	0.0

Table 131: Emission matrix of positive HMM

s0	N(2033.444791, 14.547369)
s1	N(1877.438905, 49.915468)
s2	N(2169.923109, 70.926335)
s3	N(987.79082, 77.447015)

Table 132: Transition matrix of positive HMM

	s0	s1	s2	s3
s0	0.963491	0.000000	0.036509	0.0
s1	0.010971	0.989029	0.000000	0.0
s2	0.000000	0.013715	0.986285	0.0
s3	0.000000	1.000000	0.000000	0.0

Table 133: Initial state probabilities vector of negative HMM

s0	0.0
s1	0.0
s2	1.0
s3	0.0

Table 134: Emission matrix of negative HMM

s0	N(1018.269268, 62.78185)
s1	N(1764.789362, 54.143313)
s2	N(1984.300265, 63.368285)
s3	N(1882.088195, 17.391001)

Table 135: Transition matrix of negative HMM

	s0	s1	s2	s3
s0	0.0	0.264123	0.000000	0.735877
s1	0.0	0.961914	0.000000	0.038086
s2	0.0	0.000000	0.970762	0.029238
s3	0.0	0.018182	0.009134	0.972684

Multivariate HHM for hourly return data 800 obs

Table 136: Initial state probabilities vector of positive HMM

s0	0.0
s1	0.0
s2	0.0
s3	0.0
s4	0.0
s5	0.0
s6	1.0
s7	0.0

Table 137: Emission matrix of positive HMM

s0	N(-20.075633, 5.474202)
s1	N(-4.140214, 6.449671)
s2	N(59.60029, 77.907365)
s3	N(21.502738, 8.071347)
s4	N(2.289299, 2.880107)
s5	N(14.110205, 13.840083)
s6	N(-0.466484, 4.697067)
s7	N(6.461699, 5.088513)

Table 138: Transition matrix of positive HMM

	s0	s1	s2	s3	s4	s5	s6	s7
s0	0.000000	0.703981	0.154110	0.141909	0.000000	0.000000	0.000000	0.000000
s1	0.000000	0.568479	0.000000	0.021521	0.000000	0.038735	0.000000	0.371265
s2	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000
s3	0.000000	0.000000	0.000000	0.000000	0.640821	0.000000	0.226946	0.132233
s4	0.000000	0.000000	0.024564	0.152839	0.000000	0.000000	0.822597	0.000000
s5	0.182088	0.000000	0.000000	0.000000	0.000000	0.817912	0.000000	0.000000
s6	0.109495	0.000000	0.000000	0.036774	0.853731	0.000000	0.000000	0.000000
s7	0.096087	0.490006	0.000000	0.000000	0.000000	0.065741	0.090310	0.257856

Table 139: Initial state probabilities vector of negative HMM

s0	0.0
s1	0.0
s2	0.0
s3	0.0
s4	1.0
s5	0.0
s6	0.0
s7	0.0

Table 140: Emission matrix of negative HMM

s0	N(-5.278832, 6.265705)
s1	N(34.772423, 20.453443)
s2	N(-42.505772, 18.064031)
s3	N(-0.512273, 3.704983)
s4	N(0.484566, 24.453747)
s5	N(-115.849788, 16.855203)
s6	N(5.425868, 7.660269)
s7	N(-2.521976, 17.832519)

Table 141: Transition matrix of negative HMM

	s0	s1	s2	s3	s4	s5	s6	s7
s0	0.288809	0.000000	0.000000	0.254628	0.000000	0.017833	0.438729	0.000000
s1	0.000000	0.000000	0.000000	0.000000	0.000000	0.072212	0.000000	0.927788
s2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.441724	0.558276
s3	0.000000	0.000000	0.000000	0.772497	0.219089	0.008414	0.000000	0.000000
s4	0.349539	0.000000	0.000000	0.000000	0.000000	0.074604	0.575857	0.000000
s5	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
s6	0.722432	0.000000	0.000000	0.000000	0.000000	0.000000	0.277568	0.000000
s7	0.000000	0.147174	0.055846	0.000000	0.000000	0.000000	0.000000	0.796980

Multivariate HHM for hourly close price data 1858 obs

Table 142: Initial state probabilities vector of positive HMM

s0	0.0
s1	0.0
s2	1.0

Table 143: Emission matrix of positive HMM

s0	N(1768.736558, 25.072984)
s1	N(1168.108536, 43.574594)
s2	N(1915.701865, 52.2941)

Table 144: Transition matrix of positive HMM

	s0	s1	s2
s0	0.994876	0.0	0.005124
s1	1.000000	0.0	0.000000
s2	0.008396	0.0	0.991604

Table 145: Initial state probabilities vector of negative HMM

s0	0.0
s1	0.0
s2	1.0

Table 146: Emission matrix of negative HMM

s0	N(1756.124976, 28.226435)
s1	N(2047.520733, 48.58692)
s2	N(1853.522325, 32.119029)

Table 147: Transition matrix of negative HMM

	s0	s1	s2
s0	0.963085	0.000000	0.036915
s1	0.000000	0.985303	0.014697
s2	0.017661	0.005030	0.977310

Multivariate HHM for hourly return data 1858 obs

Table 148: Initial state probabilities vector of positive HMM

s0	0.0
s1	1.0
s2	0.0
s3	0.0
s4	0.0
s5	0.0
s6	0.0
s7	0.0

Table 149: Emission matrix of positive HMM

s0	N(-23.080433, 3.295244)
s1	N(53.301278, 105.10802)
s2	N(13.969846, 8.956859)
s3	N(9.186824, 4.462533)
s4	N(0.733865, 1.319265)
s5	N(-4.163188, 5.655828)
s6	N(0.86825, 3.157686)
s7	N(5.064734, 2.577267)

Table 150: Transition matrix of positive HMM

	s0	s1	s2	s3	s4	s5	s6	s7
s0	0.000000	0.00000	0.123045	0.000000	0.000000	0.876955	0.000000	0.000000
s1	0.000000	0.00000	0.000000	0.415704	0.000000	0.584296	0.000000	0.000000
s2	0.041790	0.00000	0.000000	0.141585	0.000000	0.329119	0.350851	0.136655
s3	0.000000	0.00000	0.287216	0.000000	0.574501	0.086475	0.000000	0.051807
s4	0.041800	0.00000	0.000000	0.198064	0.219114	0.000000	0.541023	0.000000
s5	0.006350	0.00000	0.175179	0.000000	0.000000	0.186499	0.569620	0.062353
s6	0.000000	0.00000	0.071908	0.000000	0.186365	0.692815	0.000000	0.048912
s7	0.067155	0.01986	0.000000	0.122814	0.200626	0.000000	0.000000	0.589545

Table 151: Initial state probabilities vector of negative HMM

s0	0.0
s1	1.0
s2	0.0
s3	0.0
s4	0.0
s5	0.0
s6	0.0
s7	0.0

Table 152: Emission matrix of negative HMM

s0	N(5.764038, 9.763766)
s1	N(-7.970582, 3.799571)
s2	N(1.63538, 2.683107)
s3	N(1.993417, 4.112958)
s4	N(-2.961853, 6.079174)
s5	N(-2.268251, 2.502579)
s6	N(-3.965262, 21.676379)
s7	N(1.43996, 3.072994)

Table 153: Transition matrix of negative HMM

	s0	s1	s2	s3	s4	s5	s6	s7
s0	0.102910	0.512498	0.384592	0.000000	0.000000	0.000000	0.000000	0.000000
s1	0.000000	0.094002	0.163568	0.000000	0.042593	0.434602	0.000000	0.265235
s2	0.000000	0.000000	0.155019	0.000000	0.000000	0.000000	0.000000	0.844981
s3	0.000000	0.000000	0.090798	0.000000	0.000000	0.000000	0.909202	0.000000
s4	0.329284	0.000000	0.000000	0.541419	0.077975	0.000000	0.051322	0.000000
s5	0.000000	0.000000	0.000000	0.000000	0.687265	0.000000	0.000000	0.312734
s6	0.813171	0.063333	0.000000	0.000000	0.000000	0.000000	0.123496	0.000000
s7	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000

Multivariate HHM for Minute Dataset

Multivariate HHM for minute close price data 800 obs

Table 154: Initial state probabilities vector of positive HMM

s0	0.0
s1	0.0
s2	0.0
s3	1.0
s4	0.0
s5	0.0
s6	0.0
s7	0.0

Table 155: Emission matrix of positive HMM

s0	N(1535.1616, 9.235489)
s1	N(2154.178374, 13.853233)
s2	N(2458.304619, 5.834312)
s3	N(2491.020044, 9.792946)
s4	N(2352.263741, 12.190845)
s5	N(2352.263741, 9.789618)
s6	N(1337.076232, 13.114394)
s7	N(1262.794219, 16.43917)

Table 156: Transition matrix of positive HMM

	s0	s1	s2	s3	s4	s5	s6	s7
s0	0.153372	0.110956	0.049175	0.185234	0.161976	0.014813	0.186550	0.137925
s1	0.000000	0.000000	0.001172	0.998828	0.000000	0.000000	0.000000	0.000000
s2	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
s3	0.000000	0.000000	0.006412	0.993588	0.000000	0.000000	0.000000	0.000000
s4	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000
s5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
s6	0.088908	0.209540	0.175623	0.037221	0.144831	0.211727	0.038035	0.094116
s7	0.075192	0.037737	0.027081	0.212587	0.206111	0.120506	0.166183	0.154602

Table 157: Initial state probabilities vector of negative HMM

s0	0.0
s1	0.0
s2	0.0
s3	1.0
s4	0.0
s5	0.0
s6	0.0
s7	0.0

Table 158: Emission matrix of negative HMM

s0	N(1530.703269, 17.625749)
s1	N(1826.968418, 11.317586)
s2	N(1728.213368, 10.018847)
s3	N(2469.483249, 18.61111)
s4	N(1901.034705, 15.02827)
s5	N(1308.504408, 16.698078)
s6	N(2448.839518, 0.099999)
s7	N(1530.703269, 10.018847)

Table 159: Transition matrix of negative HMM

	s0	s1	s2	s3	s4	s5	s6	s7
s0	0.139194	0.280043	0.047340	0.015645	0.102815	0.159353	0.180568	0.075040
s1	0.162788	0.206177	0.220620	0.127143	0.174608	0.102876	0.005163	0.000625
s2	0.179281	0.036407	0.140901	0.164363	0.181763	0.103916	0.005244	0.188126
s3	0.000000	0.000000	0.000000	0.996212	0.000000	0.000000	0.003788	0.000000
s4	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000
s5	0.191329	0.057355	0.156834	0.180747	0.139653	0.093166	0.076796	0.104120
s6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
s7	0.106744	0.124955	0.128859	0.113057	0.183494	0.172505	0.150927	0.019460

Multivariate HHM for minute return data 800 obs

Table 160: Initial state probabilities vector of positive HMM

s0	0.0
s1	1.0
s2	0.0
s3	0.0
s4	0.0
s5	0.0
s6	0.0
s7	0.0

Table 161: Emission matrix of positive HMM

s0	N(-1.558676, 0.756473)
s1	N(0.423205, 0.150446)
s2	N(1.693765, 2.446124)
s3	N(1.652346, 0.402738)
s4	N(-1.964377, 0.125919)
s5	N(-0.679966, 2.544655)
s6	N(0.50042, 0.584363)
s7	N(1.084604, 0.612203)

Table 162: Transition matrix of positive HMM

	s0	s1	s2	s3	s4	s5	s6	s7
s0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
s1	0.000000	0.170608	0.214527	0.614866	0.000000	0.000000	0.000000	0.000000
s2	0.000000	0.000000	0.720456	0.000000	0.000000	0.279544	0.000000	0.000000
s3	0.000000	0.000000	0.000000	0.425491	0.092484	0.000000	0.170912	0.311113
s4	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
s5	0.000000	0.000000	0.230951	0.000000	0.000000	0.044707	0.000000	0.724341
s6	0.202577	0.076646	0.000000	0.000000	0.000000	0.000000	0.720777	0.000000
s7	0.000000	0.298946	0.000000	0.000000	0.000000	0.429520	0.000000	0.271534

Table 163: Initial state probabilities vector of negative HMM

s0	0.0
s1	1.0
s2	0.0
s3	0.0
s4	0.0
s5	0.0
s6	0.0
s7	0.0

Table 164: Emission matrix of negative HMM

s0	N(-0.85567, 0.145301)
s1	N(-0.17181, 0.215036)
s2	N(-1.386152, 0.367709)
s3	N(0.865824, 0.976414)
s4	N(-2.827417, 1.201794)
s5	N(-0.920152, 1.524207)
s6	N(-0.707353, 1.499156)
s7	N(3.289406, 2.467702)

Table 165: Transition matrix of negative HMM

	s0	s1	s2	s3	s4	s5	s6	s7
s0	0.119565	0.179549	0.034017	0.666869	0.000000	0.000000	0.000000	0.000000
s1	0.008531	0.497408	0.000000	0.000000	0.000000	0.000000	0.418528	0.075532
s2	0.298197	0.000000	0.000000	0.230105	0.036448	0.000000	0.435250	0.000000
s3	0.000000	0.214293	0.494302	0.291405	0.000000	0.000000	0.000000	0.000000
s4	0.000000	0.000000	0.000000	0.000000	0.000000	0.800335	0.000000	0.199665
s5	0.000000	0.000000	0.007950	0.000000	0.456315	0.437927	0.000000	0.097808
s6	0.161161	0.142308	0.117828	0.408571	0.000000	0.000000	0.000000	0.170132
s7	0.098244	0.149537	0.177254	0.000000	0.574965	0.000000	0.000000	0.000000

Multivariate HHM for minute close price data 1858 obs

Table 166: Initial state probabilities vector of positive HMM

s0	0.0
s1	0.0
s2	1.0
s3	0.0

Table 167: Emission matrix of positive HMM

s0	N(1726.458757, 37.838985)
s1	N(1920.989321, 35.7132)
s2	N(2384.333375, 12.404796)
s3	N(2456.918485, 29.073996)

Table 168: Transition matrix of positive HMM

	s0	s1	s2	s3
s0	0.0	0.0	0.998932	0.001068
s1	0.0	0.0	0.997822	0.002178
s2	0.0	0.0	0.995082	0.004918
s3	0.0	0.0	0.000000	1.000000

Table 169: Initial state probabilities vector of negative HMM

s0	0.0
s1	1.0
s2	0.0
s3	0.0

Table 170: Emission matrix of negative HMM

s0	N(1518.834426, 26.577659)
s1	N(2397.534049, 24.438307)
s2	N(1371.850449, 35.87984)
s3	N(2480.8491, 13.14385)

Table 171: Transition matrix of negative HMM

	s0	s1	s2	s3
s0	0.0	0.475148	0.0	0.524852
s1	0.0	0.996198	0.0	0.003802
s2	0.0	0.999563	0.0	0.000437
s3	0.0	0.000000	0.0	1.000000

Multivariate HHM for minute return data 1858 obs

Table 172: Initial state probabilities vector of positive HMM

	7
s0	0.0
s1	1.0
s2	0.0
s3	0.0
s4	0.0
s5	0.0
s6	0.0

Table 173: Emission matrix of positive HMM

s0	N(-0.357118, 1.156212)
s1	N(0.922951, 0.113866)
s2	N(-1.08847, 1.634198)
s3	N(0.274542, 0.303944)
s4	N(1.03678, 2.272437)
s5	N(1.508373, 0.701397)
s6	N(-0.587116, 0.193835)

Table 174: Transition matrix of positive HMM

	s0	s1	s2	s3	s4	s5	s6
s0	0.041951	0.000000	0.221296	0.736753	0.000000	0.000000	0.000000
s1	0.000000	0.238378	0.000000	0.351761	0.176727	0.233133	0.000000
s2	0.141079	0.000000	0.421884	0.000000	0.000000	0.347980	0.089057
s3	0.111723	0.025207	0.120340	0.343641	0.000000	0.252681	0.146409
s4	0.000000	0.170334	0.000000	0.000000	0.829666	0.000000	0.000000
s5	0.065294	0.026254	0.173964	0.189239	0.032237	0.329155	0.183855
s6	0.578445	0.230170	0.000000	0.000000	0.000000	0.000000	0.191385

Table 175: Initial state probabilities vector of negative HMM

s0	0.0
s1	0.0
s2	0.0
s3	0.0
s4	1.0
s5	0.0
s6	0.0

Table 176: Emission matrix of negative HMM

s0	N(-0.215905, 0.503073)
s1	N(-0.340396, 2.215899)
s2	N(0.812044, 1.453226)
s3	N(5.761663, 0.496274)
s4	N(2.92366, 16.142816)
s5	N(-2.323837, 2.134517)
s6	N(-0.936038, 1.271269)

Table 177: Transition matrix of negative HMM

	s0	s1	s2	s3	s4	s5	s6
s0	0.528938	0.000000	0.000000	0.002569	0.000000	0.021350	0.447144
s1	0.000000	0.941325	0.000000	0.000000	0.006205	0.016719	0.035752
s2	0.348283	0.000000	0.411973	0.006764	0.000000	0.010099	0.222881
s3	0.000000	0.000000	0.000000	0.148408	0.000000	0.851592	0.000000
s4	0.000000	0.000000	0.000000	0.613461	0.386539	0.000000	0.000000
s5	0.000000	0.250520	0.000000	0.152821	0.000000	0.596659	0.000000
s6	0.000000	0.000000	0.583837	0.000000	0.000000	0.000000	0.416163