Introduction of Hidden Markov Model

Qing Shi, Jinyi Luo, Yueer Li

May 1, 2019

THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Abstract

The objective of this research is to understand what Hidden Markov Model is, and all the theorems, properties behind, and then apply it in data S&P 500 from 2004 to 2019. A Markov Chain model is determined based on probability transition matrix and initial states vector. The Hidden Markov Models consists two sequences: an internal process follows Markov property and a sequence of observations which only depends on the current hidden state. We use EM algorithm to figure out the probability distribution of the hidden process of S&P 500 and conclude that the four-state HMM fits our data well.

I. Data and Motivation

Real estate collapse resulted in the financial crisis in 2007-2008 is the most serious financial crisis since the Great Depression that millions of investors and companies were suffered from this crash. In order to explore the underlying relationship between closing price and stock market performance, Standard & Poor's 500 Index (S&P 500) has been chosen to be the representative dataset. S&P 500 is a market-capitalization-weighted index of the 500 largest American publicly traded companies. Hidden Markov Model (HMM) can be applied to S&P 500 dataset to detect the hidden regimes of the financial market.

S&P 500 dataset contains open, high, low and close prices, as well as the volume and adjusted close price from January 2nd, 2004 to April 30th, 2019. By using the closing price of S&P 500, HMM can be defined as 2-state HMM with bullish and bearish hidden states, or 3-state HMM with bullish, intermediate and bearish hidden states.

HMM has strong statistical foundation, by applying this model to our dataset, we are able to detect the underlying message of the stock market, such as whether the market is in a weaken

condition, in a boom condition or volatile condition. However, sometimes it is hard to interpret the meaning of hidden states in HMM. For example, if we use 5-state HMM to fit the data, interpretation of those 5 hidden states is hard to define in the real life.

II. Methods

Hidden Markov Model (HMM) is a statistical model that consists a sequence of observed output and a sequence of internal states which follow Markov Property. In this section, we introduce the basic ideas, important quantities, and properties of Markov Chain (MC) first, based on the knowledge of MC, we further demonstrate the concepts of HMM, including joint probability distribution, marginal distribution of observed output, and posterior probability distribution. Expectation-Maximization Algorithm (EM) is used commonly in estimating the maximum parameter distribution of the posterior.

2.1 Markov Chain

The Markov Chain and related processes have been widely used in mathematical and statistical analyses, and some significant theorems and properties have many real-life applications, such as stock return prediction, customer behaviour prediction, music composition (McAlpine, Miranda and Hoggar, 1999). MC as the fundamental Markov model, has been regarded as the foundation theory of machine learning algorithm, Hidden Markov Model, Markov random field, and Markov decision process.

Markov Chain process can be either discrete or continuous. A Markov Chain is a sequence of random variables $\{C_t : t \in \mathbb{N}\}$ with m states that satisfies the property, the probability of the

current state information only depends on the previous state, independent to the other prior previous states. Mathematically expressed the Markov property as:

$$\mathrm{P}(C_{t+1} \mid C_t, \dots, C_1) = \mathrm{P}(C_{t+1} \mid C_t) \tag{2.1}$$

This "memorylessness" property eliminates the effect that the history $(C_1, C_2, \dots, C_{t-1})$, and denotes that the past state and future state are dependent only through the present (Zucchini, 2016).

In Markov Model, each state in $\{Ct\}$ has a weighted transition probability γ_{ij}, the probability of moving from the current state i to the other state j, $i, j = 1, 2, \dots m$, states prior to the previous state has no effect on the transition probability.

$$\gamma_{ij}(1) = P(C_{t+1} = j | C_t = i)$$

This is a one-step transition probability from state i to state j, expressed in a matrix form $\mathbf{\Gamma}$:

$$\mathbf{\Gamma} = \begin{pmatrix} \gamma_{11} & \cdots & \gamma_{1m} \\ \vdots & \ddots & \vdots \\ \gamma_{m1} & \cdots & \gamma_{mm} \end{pmatrix}$$

The t-step transition probability for longer time can be denoted as

$$\gamma_{ij}(t) = P(C_{s+t} = j \,|C_s = i)$$

The matrix $\mathbf{\Gamma}(\mathrm{t})$ is defined as the matrix with (i, j) element $\gamma_{ij}(t)$. If these transition probabilities are independent to s, then we call this MC is homogenous. It satisfies the Chapman-Kolmogorov equations:

$$\mathbf{\Gamma}(t + u) = \mathbf{\Gamma}(t)\mathbf{\Gamma}(u) \tag{2.2}$$

We also interested unconditional probability distribution $P(C_t = i)$ that being in a given state at a given time, and it can be expressed in a row vector:

$$\boldsymbol{u}(t) = (P(C_t = 1), P(C_t = 2), \dots P(C_t = m))$$

$$= \big(u_1(t), u_2(t), \dots u_m(t)\big), t \in N$$

Assume $\boldsymbol{u}(1)$ is the initial distribution of the MC, then the distribution for time 2 is $\boldsymbol{u}(2) = \boldsymbol{u}(1)\boldsymbol{\Gamma}$. It can be generated into a general case:

$$\boldsymbol{u}(t+1) = \boldsymbol{u}(t)\boldsymbol{\Gamma} = \boldsymbol{u}(1)\boldsymbol{\Gamma}^t \tag{2.3}$$

To consider the probability of a sequence of Markov states, the joint probability distribution helps such that for any state

$$\begin{aligned} & P(C_1, C_2, \dots, C_t) \\ &= P(C_1)P(C_2 \mid C_1)P(C_3 \mid C_2, C_1) \dots P(C_t \mid C_{t-1}, C_{t-2}, \dots) \\ &= P(C_1)P(C_2 \mid C_1)P(C_3 \mid C_2) \dots P(C_t \mid C_{t-1}) \\ &= P(C_1)\prod_{k=2}^{t} P(C_k \mid C_{k-1}) \end{aligned}$$

Stationarity is considered to be a special element for Markov Chain to discuss. We define a Markov Chain with transition probability matrix $\boldsymbol{\Gamma}$ has stationary distribution with a row vector with non-negative elements $\boldsymbol{\delta}$ if

$$\boldsymbol{\delta\Gamma} = \boldsymbol{\delta}\ and\ \boldsymbol{\delta 1} = 1$$

The continuity implies that a Markov Chain will continue to have stationary distributions at all subsequent time points if it starts from a stationary distribution, and we shall refer to such a process as a stationary Markov Chain.

2.2 Hidden Markov Model

Hidden Markov Model (HMM) is a statistical model introduced by L. E. Baum and Ted Petrie in 1960s (Baum and Petrie, 1966), and according to Wikipedia, HMM have been applied to reinforcement learning and temporal pattern recognition such as financial daily return analysis, speech recognition and bioinformatics (2019).

HMMs are probabilistic frameworks where a sequence of observed output dependent on one of several hidden states. In a standard type of HMMs, we split it into 2 parts: firstly, an unobserved "parameter process" $\{Ct: t = 1, 2, ...\}$ is discrete, following Markov property that the current state information only depends on the previous state, which we have the equation:

$$P\left(C_t \mid C^{(t-1)}\right) = P\left(C_t \mid C_{t-1}\right), \quad t = 2, 3, ... \tag{2.4}$$

Secondly, an observed "state-dependent process" $\{X_t: t = 1, 2, ...\}$ can be either discrete or continuous, giving information about the sequence of states, depends on the current state C_t only, and not on previous states or observations. We have the equation:

$$P\left(X_t \mid X^{(t-1)}, C^{(t)}\right) = P\left(X_t \mid C_t\right), \quad t \in N \tag{2.5}$$

Figure 1 demonstrates a basic HMM structure clearly.

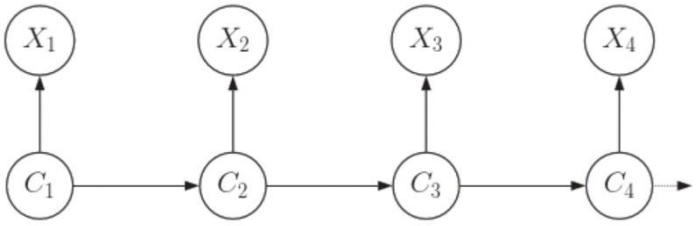


Figure 1: basic HMM structure

We call $\{X_t\}$ is a m-state HMM if underlying process $\{C_t\}$ has m state. For example, a room contains 3 boxes, X_1, X_2 and X_3, each box has 4 balls labelled Y_1, Y_2, Y_3 and Y_4. A sequence of balls is randomly drawn. In this case, balls drawn out are observed output and there are 4 possible states; we do not know the which box that each ball picked from, so there are 3 hidden states. We say it is a 3-state HMM.

In m-state HMM, $\{C_t\}$ follows Markov property, hence transition probability given as equation (2.1)

$$\gamma_{ij} = P(C_{t+1} = j | C_t = i)$$

Emission probability $p_i(x)$ is the possibility of the observed output at a particular time given the state of the hidden variable at that time. In the discrete observation case, we have the equation:

$$P\ (X_t = x \mid C_t = i) = p_i(x), i = 1,2, \dots m$$

We call $p_i(x)$ as the probability mass function of X_t associated with state i; for the continuous case, we say $p_i(x)$ is the probability density function of X_t when the Markov chain is at state x in time t.

The general idea of HMM is to find out the distribution of the hidden parameter based on the observed output, $P(C_t|X_t)$, and the maximum estimator of the parameter of HMM $\hat{v}(\mathrm{t}) = \underset{\theta}{\operatorname{argmax}} P\ (C_t|X_t)$ which will be discussed in the next section, by using the *joint probabilistic model* $P(X_t, C_t)$.

$$P(X_1, X_2, \dots, X_t, C_1, C_2, \dots, C_t)$$

$$= P(C_1)P(X_1|C_1, X_2, \dots, X_t, C_2, \dots,\ C_t)P(X_2, \dots, X_t, C_2, \dots,\ C_t)$$

$$= P(C_1)P(X_1|C_1)P(X_2, \dots, X_t, C_2, \dots,\ C_t)\ by\ equation\ (2.5)$$

$$= P(C_1)P(X_1|C_1)P(C_2|C_1)P(X_2|C_2) \dots P(C_t|C_{t-1})P(X_t|C_t)\ by\ equation\ (2.4)$$

$$= P(C_1)\prod\nolimits_{k=1}^{t} P(X_k|C_k)\prod\nolimits_{k=2}^{t} P(\ C_k|C_{k-1})$$

It is also useful to determine the marginal distribution of the observed output X_t, and higher order marginal distribution, such as (X_t, X_{t+k}). The derivation of the marginal distribution for the discrete state-dependent process would be the primary focus, and for continuous case, the derivation would be a similar equation by substituting the summation with integration.

First of all, univariate case is intuitive to compute as below:

$$P(X_t = x) = \sum\nolimits_{i=1}^{m} P(X_t = x, C_t = i)$$

$$= \sum_{i=1}^{m} P(C_t = i)P(X_t = x|C_t = i)$$

$$= \sum_{i=1}^{m} u_i(t)p_i(x)$$

The matrix form would be like:

$$P(X_t = x) = \begin{pmatrix} u_1(t) & \dots & u_m(t) \end{pmatrix} \begin{pmatrix} p_1(x) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & p_m(x) \end{pmatrix} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

$$= \boldsymbol{u}(t)\boldsymbol{P}(x)\mathbf{1}'$$

where $\boldsymbol{P}(x)$ is an $m \times m$ diagonal matrix with i-th diagonal element $p_i(x)$. By substituting $\boldsymbol{u}(t) = \boldsymbol{u}(1)\boldsymbol{\Gamma}^{t-1}$ (2.3), given:

$$P(X_t = x) = \boldsymbol{u}(1)\boldsymbol{\Gamma}^{t-1}\boldsymbol{P}(x)\mathbf{1}' \tag{2.6}$$

This equation works when the Markov Chain $\{C_t\}$ is homogenous but not necessarily to be stationary. If the Markov Chain is stationary, then there exits stationary distribution $\boldsymbol{\delta}$ such that equation (2.6) can be expressed as:

$$P(X_t = x) = \boldsymbol{\delta}\boldsymbol{P}(x)\mathbf{1}'$$

Compared to univariate case, bivariate distribution involves 2 states computing:

$$P(X_t = v, X_{t+k} = w)$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} P(X_t = v,\ X_{t+k} = w, C_t = i, C_{t+k} = j)$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} P(C_t = i)P(X_t = v|C_t = i)\ P(C_{t+k} = j|C_t = i)P(X_{t+k} = w|C_{t+k} = j)$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} u_i(t)p_i(v)\gamma_{ij}(k)p_j(w)$$

(Here $\gamma_{ij}(k)$ denotes the transition probability from i to j in k steps.) The matrix form would be:

$$P(X_t = v, X_{t+k} = w) = \boldsymbol{u}(t)\boldsymbol{P}(v)\boldsymbol{\Gamma}^k\boldsymbol{P}(w)\mathbf{1}' \tag{2.7}$$

If $\{C_t\}$is a stationary Markov Chain, equation (2.7) is given as:

$$P(X_t = v, X_{t+k} = w) = \delta \boldsymbol{P}(v)\boldsymbol{\Gamma}^k \boldsymbol{P}(w)\mathbf{1}'$$

Respectively, higher-order marginal distribution such as tri-variate distribution can be directly stated as:

$$P(X_t = v, X_{t+k} = w, X_{t+k+h} = z) = \boldsymbol{u}(t)\boldsymbol{P}(v)\boldsymbol{\Gamma}^k \boldsymbol{P}(w)\boldsymbol{\Gamma}^h \boldsymbol{P}(z)\mathbf{1}'$$

Based on the all the theorems and properties discussed above, the overall likelihood possibility of producing a sequence of output is inferred by multiplying all of the probabilities of state-paths through HMM.

$$L_T = \delta P(x_1)\Gamma P(x_2) \ldots \Gamma P(x_T)1'$$

2.3 Expectation-Maximization Algorithm

The Expectation-Maximization algorithm (EM) was introduced by Arthur Dempster, Nan Laird, and Donald Rubin (Dempster, Laird, Rubin, 1977) to discover maximum a posterior (MAP) estimates of parameters in the models. In HMM, the posterior distribution is $P(C_t|X_t)$, and the MAP would be $\hat{v}(\mathrm{t}) = \underset{\theta}{\operatorname{argmax}}\, P\,(C_t|X_t)$. The EM algorithm for HMM iterates between performing an expectated posterior parameter (E-step) and a maximum value of parameter by maximizing the expected value of log likelihood function of posterior distribution (M-step) until log likelihood values converge.

- E step: for each point, it computes the conditional expectation of posterior probability given the observation $X^{(T)}$ as well as the current parameter estimates
    - $\hat{v}_j(\mathrm{t}) = \Pr\left(\mathrm{C}_{t=j}\middle|X^{(T)}\right) = \alpha_t(j)\beta_t(j)/L_T$
    - $\hat{v}_{ij}(\mathrm{t}) = \Pr\left(\mathrm{C}_{t-1=j}, \mathrm{C}_{t=k}\middle|X^{(T)}\right) = \alpha_{t-1}(j)\gamma_{jk}p_k(x_k)\beta_t(k)/L_T$
- M step: Maximize complete-data log-likelihood (CDLL)

- $arg\max_{\theta} \log\left(\Pr\left(X^{(T)}, C^{(T)}\right)\right) = \sum_{j=1}^{m} \mu_j\,(1) log\delta_j +$

  $\sum_{j=1}^{m}\sum_{k=1}^{m}(\sum_{t=2}^{m} v_{jk}(t)) log\gamma_{jk} + \sum_{j=1}^{m}\sum_{t=1}^{T} \mu_j(t) logp_j(x_t)$

The application of the EM algorithm will be shown in the R example.

Applying to the S&P 500 dataset, we start with two random Gaussian distribution, a bull distribution with mean=μ_a, variance= ${\sigma_a}^2$ and a bear distribution with mean=μ_b, variance= ${\sigma_b}^2$.

- E step: for each point, it computes the probability the point came from the blue distribution or the other.
  - $b_i = P(b|X_i) = \frac{P(X_i|b)P(b)}{P(X_i|b)P(b)+P(X_i|a)P(a)}$, where

    $P(X_i|b)=\frac{1}{\sqrt{2\pi\sigma_b}} \exp\left(-\frac{(X_i-\mu_b)^2}{2{\sigma_b}^2}\right)$
  - $a_i = P(a|X_i) = 1 - b_i$
- M step: Once it computed these assignments, it is going to recalculate the means and variances to enhance the point assignments.
  - $\mu_b = \frac{b_1X_1+b_2X_2+\cdots+b_nX_n}{b_1+b_2+\cdots+b_n}$
  - ${\sigma_b}^2 = \frac{b_1(X_1-\mu_1)^2+\cdots+b_n(X_n-\mu_n)^2}{b_1+b_2+\cdots+b_n}$
  - $\mu_a = \frac{a_1X_1+a_2X_2+\cdots+a_nX_n}{a_1+a_2+\cdots+a_n}$
  - ${\sigma_a}^2 = \frac{a_1(X_1-\mu_1)^2+\cdots+a_n(X_n-\mu_n)^2}{a_1+a_2+\cdots+a_n}$
- This process iterates until converge

III. Seminal Research Paper

As Nguyen stated in Hidden Markov Model for Stock Trading, they used the monthly prices of S&P 500 to compute the corresponding values for 120 HMM's parameter calibrations in order to select the best Hidden Markov Model (HMM) among two to six states (2018). By comparing the Akaike information criterion (AIC), the Bayesian information criterion (BIC), the Hannan–Quinn information criterion (HQC), and the Bozdogan Consistent Akaike Information Criterion (CAIC) criteria, 4-state HMM has been selected due to the lower AIC, BIC, HQC, CAIC values, which indicate the better performance of model fitting. Applying the backward algorithm, probability of observed output (closing price) given model parameter (transition matrix, emission matrix and unconditional probability of hidden states) is computed. Moreover, they proposed the future closing price can be predicted by previous closing price plus the residual of closing prices difference between the time T*+1 and T* where T* is the time period that gave out the most similar probability distribution as that at time period T.

$O_{T+1}{}^{(4)} = O_T{}^{(4)} + \left(O_{T^*+1}{}^{(4)} - O_{T^*}{}^{(4)}\right) \times sign\ (\mathrm{P}(O|\lambda) - \mathrm{P}(O^*|\lambda))$ (Nguyen, 2018)

where (4) denotes the monthly close price of the S&P 500

In the other article - Application of Markov Chain to stock trend: A study of PT HM Sampoerna, tbk written by Fitriyanto and Lestari, Markov model is used to predict the stock trend. They used dataset from January 2017 to December 2017 to fit a 4-state Markov Chain and got the result of the stock price movement probability.

Comparing these 2 methods, HMM is more precise in forecasting stock trend by taking the hidden states (general market performance) into consideration, Markov model only takes stock prices in account. However, HMM might not be a good model for the stock price prediction. For

time T+2 stock price prediction, the model uses predicted price at time T+1, which gives the formula:

$$O_{T+2}{}^{(4)} = O_{T}{}^{(4)} + \left(O_{T^*+1}{}^{(4)} - O_{T^*}{}^{(4)}\right) \times sign\left(\mathrm{P}(O|\lambda) - \mathrm{P}(O^*|\lambda)\right) + \left(O_{T^{**}+1}{}^{(4)} - O_{T^{**}}{}^{(4)}\right) \times sign\left(\mathrm{P}(O|\lambda) - \mathrm{P}(O^{**}|\lambda)\right)$$

As prediction time period increases, the error term accumulates as time goes on.

IV. Documentation of R Code

Four libraries have been used in the R code:

depmixS4: A framework for specifying and fitting dependent mixture models (HMM)

quantmod: Quantitative Financial Modelling and Trading Framework for R

TTR: Technical Trading Rules

forecast: A generic function for forecasting from time series or time series models.

In the application of HMM, the two state HMM perform well, which assigned high posterior probabilities for Regime #1 during the global financial crisis of 2007 to 2008, where the markets were extremely volatile. However, the results from three states are not stable, the model switching behavior between Regime #1 and Regime #2 during 2004 to 2007, where the market performance were calmer. The model indicates high posterior probabilities for Regime #1 in the period of 2008 to 2010, where market is more volatile. Besides, the model switching behavior between Regime #1, Regime #2 and Regime #3 for the later time. Therefore, the hidden state is not obvious.

The detailed R code can be found in Appendix.

Bibliography

Baum, L. E., & Petrie, T. (1966). *Statistical Inference for Probabilistic Functions of Finite State Markov Chains*. Retrieved from https://projecteuclid.org/DPubS/Repository/1.0/Disseminate?handle=euclid.aoms/1177699147&view=body&content-type=pdf_1

Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). *A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains*. The Annals of Mathematical Statistics, 41(1), 164-171. doi:10.1214/aoms/1177697196

Dempster, A.P.; Laird, N.M.; Rubin, D.B. (1977). *Maximum Likelihood from Incomplete Data via the EM Algorithm*. Journal of the Royal Statistical Society, Series B. 39 (1): 1–38. JSTOR 2984875. MR 0501537.

Fitriyanto, A., Lestari, T.E. (2018). *Application of Markov Chain to stock trend: A study of PT HM Sampoerna, tbk.* IOP Conf. Ser.: Mater. Sci. Eng. 434 012007

Lavrenko, V. (2014, January 19). *Expectation Maximization: How it works*. Retrieved from https://www.youtube.com/watch?v=iQoXFmbXRJA

Mcalpine, K., Miranda, E., & Hoggar, S. (1999). Making Music with Algorithms: A Case-Study System. *Computer Music Journal,23*(2), 19-30. doi:10.1162/014892699559733

Nguyen, N. (2018). *Hidden Markov Model for Stock Trading*. International Journal of Financial Studies, 6(2). https://doi.org/10.3390/ijfs6020036

Zucchini, W., MacDonald, I. L., & Langrock, R. (2016). *Hidden markov models for time series: An introduction using R.* Boca Raton: CRC Press, Taylor & Francis Group.

Appendix

Hidden Markov Models for Regime Detection

```
library(depmixS4) # A framework for specifying and fitting dependent mixture models, otherwise known as hid
den or latent Markov models.
```

```
## Warning: package 'depmixS4' was built under R version 3.5.3
```

```
## Loading required package: nnet
```

```
## Loading required package: MASS
```

```
## Loading required package: Rsolnp
```

```
## Warning: package 'Rsolnp' was built under R version 3.5.3
```

```
library(quantmod) # Quantitative Financial Modelling and Trading Framework for R
```

```
## Warning: package 'quantmod' was built under R version 3.5.3
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
library(TTR)# Technical Trading Rules
library(forecast) # A generic function for forecasting from time series or time series models
```

```
## Warning: package 'forecast' was built under R version 3.5.2
```

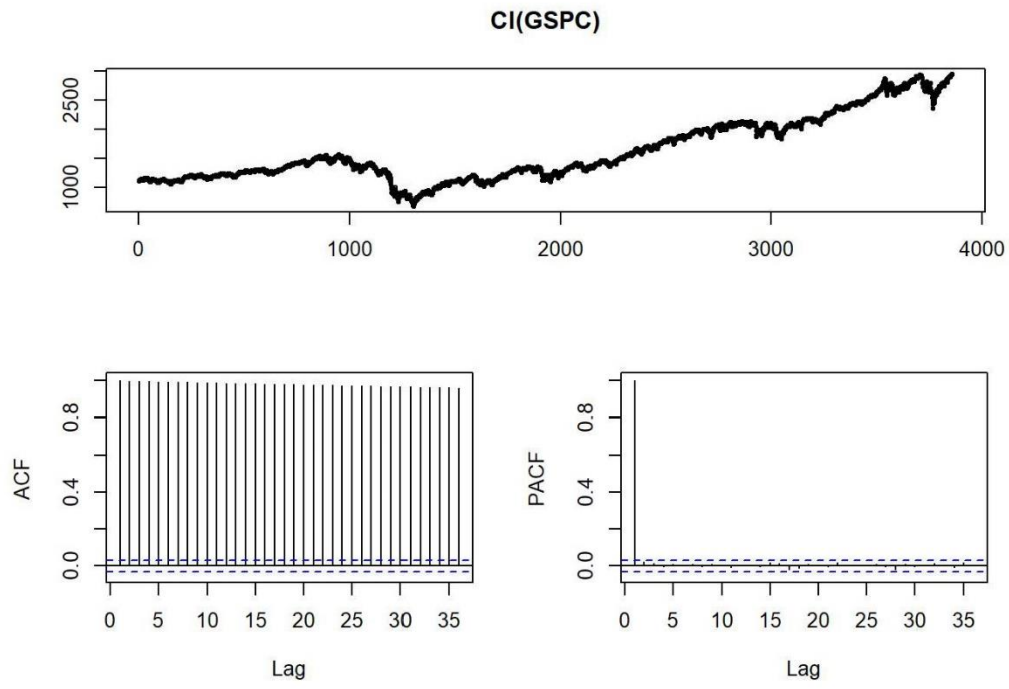
```
# Obtain S&P500 data from 2004 up to current
getSymbols( "^GSPC", from="2004-01-01" )
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

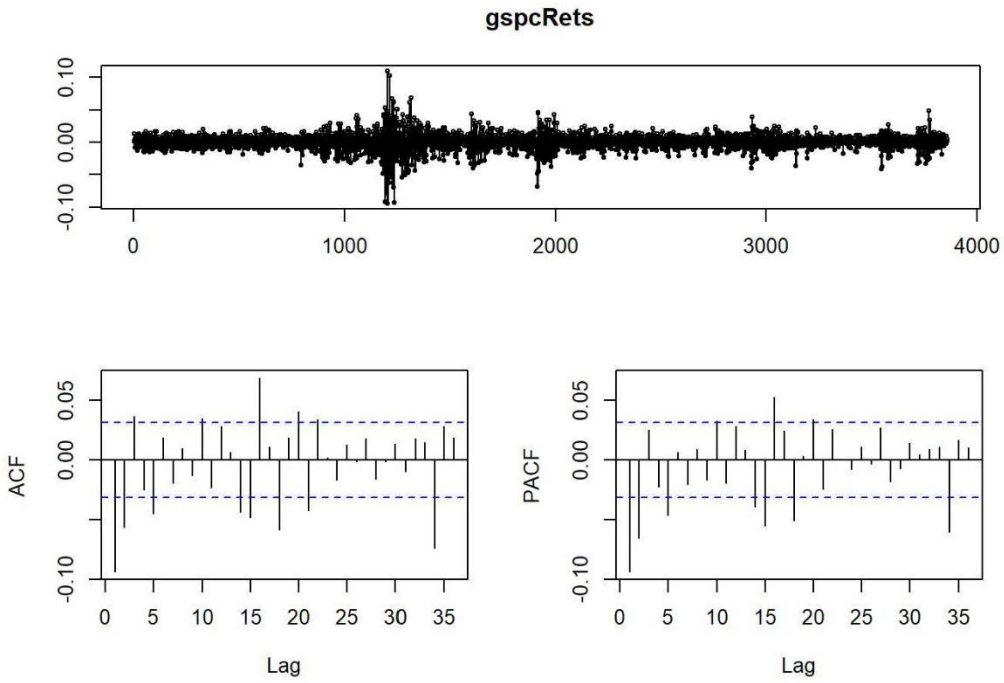
```
## [1] "^GSPC"
```

```
# Log transformation and one difference is needed for converting the close price to growth rate and data be
comes more stable after transformation.
gspcRets = diff(log(Cl(GSPC)))
returns = as.numeric(gspcRets)
```

```
tsdisplay(Cl(GSPC)) # upward trend
```




```
tsdisplay(gspcRets) # In order to make the sequence more stationary, we did log transformation and take 1 d
ifference
```




```
# Fit a Hidden Markov Model with two states to the S&P500 returns stream
hmm <- depmix(returns ~ 1, family = gaussian(), nstates = 2, data=data.frame(returns=returns))
hmmfit <- fit(hmm)
```

```
## iteration 0 logLik: 11819.76
## iteration 5 logLik: 12450.45
## iteration 10 logLik: 12558.49
## iteration 15 logLik: 12650.12
## iteration 20 logLik: 12688.75
## iteration 25 logLik: 12693.58
## iteration 30 logLik: 12693.88
## iteration 35 logLik: 12693.9
## converged at iteration 38 with logLik: 12693.9
```

```
# Initial probabilities
summary(hmmfit, which = "prior")
```

```
## Initial state probabilties model
## pr1 pr2
##   0   1
```

```
# Transition probabilities
summary(hmmfit, which = "transition")
```

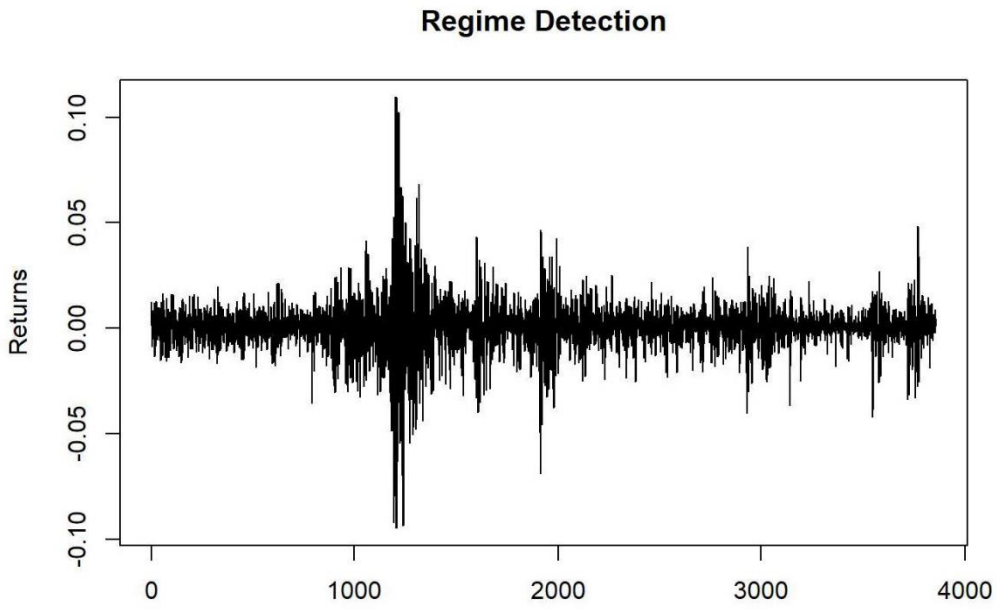
```
## Transition matrix
##         toS1  toS2
## fromS1 0.965 0.035
## fromS2 0.012 0.988
```

```
# Reponse/Emission function
summary(hmmfit, which = "response")
```

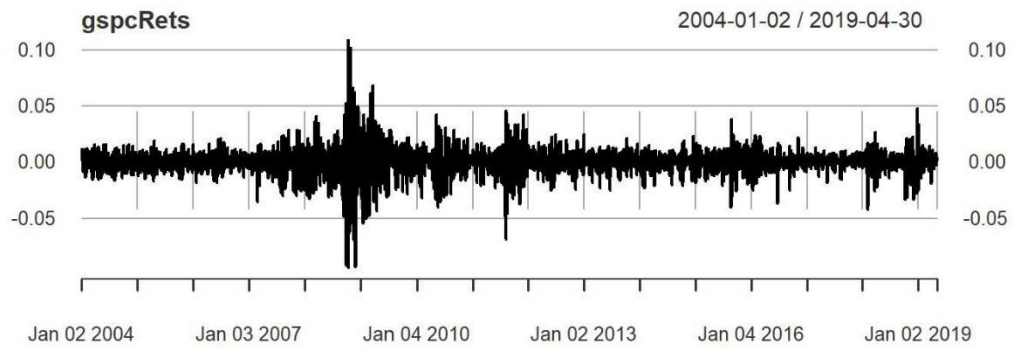
```
## Response parameters
## Resp 1 : gaussian
##     Re1.(Intercept) Re1.sd
## St1          -0.001  0.020
## St2           0.001  0.007
```

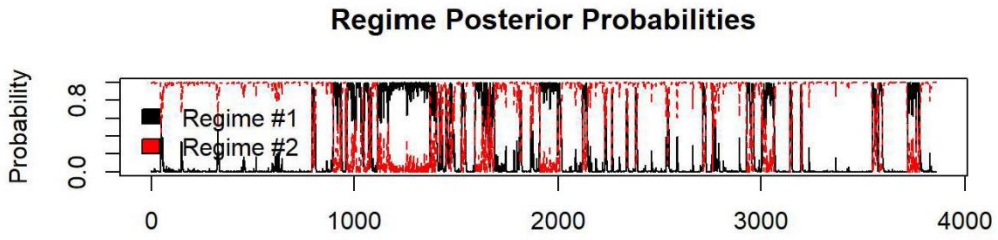
```
post_probs <- posterior(hmmfit)

# Plot the returns stream and the posterior probabilities of the separate regimes
layout(1:1)
plot(returns, type='l', main='Regime Detection', xlab='', ylab='Returns')
```




```
par(mfrow=c(2,1))
plot(gspcRets)
matplot(post_probs[,-1], type='l', main='Regime Posterior Probabilities', ylab='Probability')
legend(x='bottomleft', c('Regime #1','Regime #2'), fill=1:2, bty='n')
```






```
# Fit a Hidden Markov Model with three states to the S&P500 returns stream
hmm <- depmix(returns ~ 1, family = gaussian(), nstates = 3, data=data.frame(returns=returns))
hmmfit <- fit(hmm)
```

```
## iteration 0 logLik: 11887.78
## iteration 5 logLik: 12504.33
## iteration 10 logLik: 12642.1
## iteration 15 logLik: 12752.64
## iteration 20 logLik: 12770.84
## iteration 25 logLik: 12774.07
## iteration 30 logLik: 12777.29
## iteration 35 logLik: 12780.79
## iteration 40 logLik: 12784.8
## iteration 45 logLik: 12789.65
## iteration 50 logLik: 12795.72
## iteration 55 logLik: 12803.26
## iteration 60 logLik: 12812.35
## iteration 65 logLik: 12823.49
## iteration 70 logLik: 12840
## iteration 75 logLik: 12858.14
## iteration 80 logLik: 12865.6
## iteration 85 logLik: 12866.83
## iteration 90 logLik: 12866.92
## iteration 95 logLik: 12866.92
## converged at iteration 96 with logLik: 12866.92
```

```
# Initial probabilities
summary(hmmfit, which = "prior")
```

```
## Initial state probabilties model
## pr1 pr2 pr3
##   1   0   0
```

```
# Transition probabilities
summary(hmmfit, which = "transition")
```

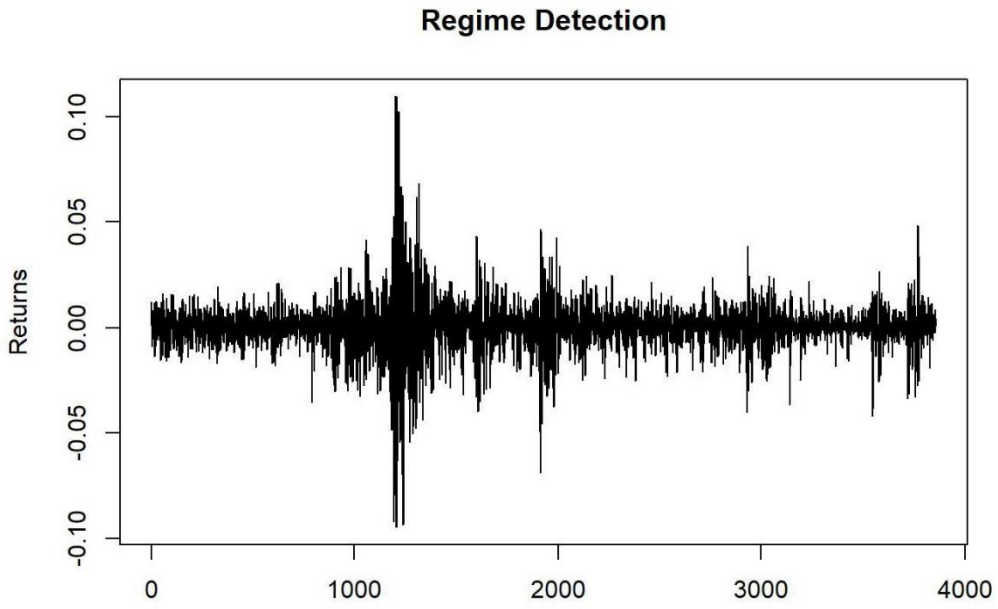
```
## Transition matrix
##         toS1  toS2  toS3
## fromS1 0.957 0.038 0.005
## fromS2 0.025 0.975 0.000
## fromS3 0.024 0.000 0.976
```

```
# Reponse/Emission function
summary(hmmfit, which = "response")
```

```
## Response parameters
## Resp 1 : gaussian
##     Re1.(Intercept) Re1.sd
## St1           0.000  0.011
## St2           0.001  0.005
## St3          -0.002  0.028
```

```
post_probs <- posterior(hmmfit)

# Plot the returns stream and the posterior probabilities of the separate regimes
layout(1:1)
plot(returns, type='l', main='Regime Detection', xlab='', ylab='Returns')
```




```
par(mfrow=c(2,1))
plot(gspcRets)
matplot(post_probs[,-1], type='l', main='Regime Posterior Probabilities', ylab='Probability')
legend(x='bottomleft', c('Regime #1','Regime #2', 'Regime #3'), fill=1:3, bty='n')
```