Contents

List of Figures

List of Tables

Abstract

In this thesis, we shall consider a stock's volatility estimation

modelling ARCH(1) stochastic process for financial time series of in combination with following three models, namely, an ARCH(1), a GARCH(1,1) and a Markov regime switching model.

More precisely, we attempt to use various modern numerical solutions, techniques and tools to generate the respective answers.

MS-ARCH(p, q) has become a growing area of interest

Fubini, in process,... We use MCMC methodology and Bayesian Inference to estimate a vector of unknown parameters for stochastic ARCH process Markov Switching Model. We use modern statistical computing to simulate based on Gibbs Sampling. We maximize convex likelihood for ARCH(1) model.

Keywords and phrases: TIME SERIES MODELLING; AUTOREGRESSIVE CONDITIONAL HETEROSCEDASTICITY; ARCH MODELS; MARKOV CHAIN MONTE CARLO; MCMC; MARKOV CHAIN METHODS OF SAMPLING; GIBBS SAMPLING; METROPOLIS ALGORITHM, METROPOLIS HASTINGS ALGORITHM; METROPOLIS WITHIN GIBBS ALGORITHM; STATISTICAL COMPUTING; MODERN STATISTICS PARAMETER ESTIMATION; TIME VARYING TRANSITION PROBABILITIES; HIDDEN MARKOV MODELS; HMM; REGIME CHANGES; MARKOV REGIME SWITCHING MODELS; MSM; MRSM; MARKOV REGIME-SWITCHING ARCH MODEL; STOCHASTIC SIMULATION; PROBABILISTIC PROGRAMMING FRAMEWORK; COMPUTATIONAL STATISTICS; STOCK PRICES; STOCK VOLATILITY; MACHINE LEARNING; NON-STATIONARY; NONLINEAR FILTERING; BAYESIAN STATISTICS; BAYESIAN INFERENCE; BAYESIAN PHILOSOPHY; BAYESIAN DATA ANALYSIS; BAYESIAN COMPUTATIONS.

1 Introduction

The theory that we introduce throughout the thesis, will be illustrated through an application on on two publicly traded stocks with daily observations.

We want to take a closer look at, one popular and widely used approach in academia to modelling volatility of various economic and financial time series (such as, stock prices, an empirical analysis of a business cycle, etc. . .), which is Autoregressive Conditional Heteroscedasticity Model's specifications and theoretical aspects, that was introduced by (Engle, 1982); cf., inter alia (Hamilton and Susmel, 1994, p. 308).

to modelling volatility of various

The rest of the thesis is organized in the following way. Section 1 sets out the basic Fubini model. In section 2 we will present a general introduction of time series models,. . . Followed by discussions of Fubini in section 3 focuses on the FUBINI,. . . .. Then, in section 4 FUBINI,. . . application of a conventional hidden Markov model in stocks' data performed and related experimental results discussed in section 5. Finally, in section 6, we provide a summary discussion. In section 7, we provide an application of the proposed methods to financial time series for two stocks. Finally, in Section FUBINI we discuss issues of the practical implementation.

2 Financial Time Series

Fubini, in process. . . We are interested in analysis of financial time series.

2.1 Fubini

Fubini, in process. . .

2.1.1 Fubini

Fubini, in process. . .

2.2 Conditional Heteroscedastic Models

The absolutely first econometric model that was successful to modify the traditional unrealistic assumption about a constant one-period forecast variance, was proposed by (Engle, 1982). Robert Engle in his paper introduced a new class of stochastic processes, the so-called autoregressive conditional heteroscedastic (ARCH) processes. He has also provided specifications and systematic framework for a setup of dynamic volatility modelling. Later, (Bollerslev, 1986) characterized a general class of ARCH as generalized ARCH (GARCH(m, n)).

The ARCH-effects, which is also known as the conditional heteroscedasticity, is a well-known phenomenon. It represents tendency of volatility clustering of a stock market data. Such that, a large(small) change has a tendency to replicate itself, such that it is followed by somewhat larger(smaller) relative changes. Now, ARCH and GARCH models have such an ability to model this phenomenon. These kind of models widely used and in variety of academic papers and is a popular approach in financial applications (Canarellar and Pollard, 2007).

2.2.1 Properties of ARCH(m) Models

We begin our study with ARCH(m) models. Here m denotes the order of ARCH process, which is, the number of lagged states that matter of interest in a specific model, see more (Tsay, 2010, p. 115, p. 119), (Hamilton and Susmel, 1994, p. 308) and the code comments in SWARCH Maxseek function.

The fundamental structure of ARCH models assumes that a *shock* a_t (also known as an *innovation* or *current error term*) is serially uncorrelated process with mean zero, and variance that depends on the sum of squared recent past shocks $\{a_{t-i}^2\}_{i=1}^m$. More specifically, in ARCH(m) model we assume that a variable a_t at time t is governed by

$$a_t = \sigma_t \epsilon_t,$$

such that, the conditional variance of a_t depends on a quadratic function of its past realizations

$$\begin{aligned} \sigma_t^2 &= \alpha_0 + \alpha_1 a_{t-1}^2 + \alpha_2 a_{t-2}^2 + \cdots + \alpha_{m-1} a_{t-m-1}^2 + \alpha_m a_{t-m}^2 \quad (1) \\ &= \alpha_0 + \sum_{i=1}^{m} \alpha_i a_{t-i}^2, \quad (2) \end{aligned}$$

where $\{\epsilon_t\}_{t=1}^T$ is a sequence of Gaussian white noises[1], and

$$\underline{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_{t-m-1}, \alpha_m)$$

is a vector of unknown parameters, with $\alpha_0 > 0$ and $\alpha_i \geq 0$ conditions for $i > 0$. As a result, this guarantees the conditional variances σ_t^2 to be positive for all t.

The simple first-order model ARCH(1) can be represented, as the set of following equations

$$a_t = \sigma_t \epsilon_t, \ \sigma_t = \sqrt{\alpha_0 + \alpha_1 a_{t-1}^2}, \ \sigma_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2, \quad (3)$$

where $\alpha_0 > 0$, $\alpha_1 \geq 0$, ϵ_t is independent of the past, and σ_t is known function value of the prior information at time $t-1$.

[1] *innovations* in most econometric models modelled by standard normal distribution, though (Hamilton and Susmel, 1994) in their paper discuss the advantage of Student t distribution

First, the unconditional mean of a_t remains zero, because

$$\begin{aligned} E(a_t) &= E\left[E(a_t|F_{t-1})\right] = E\left[E(\sigma_t\epsilon_t|F_{t-1})\right] \\ &= E\left[\sigma_t\right]E\left[(\epsilon_t|F_{t-1})\right] = E\left[\sigma_t\right]E(\epsilon_t) = E\left[\sigma_t\right]0 = 0. \end{aligned}$$

Second, assuming that a_t is a stationary process with $E(a_t) = 0$, $Var(a_t) = E(a_t^2) = Var(a_{t-1}) = E(a_{t-1}^2)$, we can obtain the *unconditional variance* of a_t

$$\begin{aligned} Var(a_t) &= E(a_t^2) = E\left[E(a_t^2|F_{t-1})\right] = E\left[E(\sigma_t^2\epsilon_t^2|F_{t-1})\right] \\ &= E\left[\sigma_t^2 E(\epsilon_t^2)\right] = E\left[1\sigma_t^2\right] = E\left[\alpha_0 + \alpha_1 a_{t-1}^2\right] \\ &= \alpha_0 + \alpha_1 E\left[a_{t-1}^2\right] = \alpha_0 + \alpha_1 E\left[a_t^2\right] \\ &= \alpha_0 + \alpha_1 Var(a_t), \end{aligned}$$

here it is very interesting to observe, that the left hand side of the equation, equals the right hand side. Where, the rhs is a sum of one constant, and a product of another constant and a lhs' value. If $\alpha_1 \geq 1$ then $Var(a_t) < Var(a_{t-1})$, which means ascending variance. Therefore, the inequality remains true, as long as α_1 constraint is less then 1. Solving, with respect to the unconditional variance, we can obtain

$$\begin{aligned} Var(a_t) &= \alpha_0 + \alpha_1 Var(a_t) \Leftrightarrow \\ \alpha_0 &= Var(a_t)(1-\alpha_1) \Leftrightarrow \\ Var(a_t) &= \frac{\alpha_0}{1-\alpha_1}. \end{aligned}$$

Third, and this is especially in some financial applications, we may be interested in the kurtosis m_4 of a_t. Hence, there are some additional constraints that α_1 must follow. Assume $m_4 = E(a_t^4) = E(a_{t-1}^4)$ is finite. By applying the normality assumption of $\{\epsilon_t\}$ in eqn 1, we get

$$\begin{aligned} m_4 &= E(a_t^4) = E\left[E(a_t^4|F_{t-1})\right] = E\left[E(\sigma_t^4\epsilon_t^4|F_{t-1})\right] \\ &= E\left[\sigma_t^4 E(\epsilon_t^4)\right] = E\left[3\sigma_t^4\right] = 3E\left[(\sigma_t^2)^2\right] \\ &= 3E\left[(\alpha_0 + \alpha_1 a_{t-1}^2)^2\right] \\ &= 3E\left[\alpha_0^2 + 2\alpha_0\alpha_1 a_{t-1}^2 + \alpha_1^2 a_{t-1}^4\right] \\ &= 3(\alpha_0^2 + 2\alpha_0\alpha_1 E\left[a_{t-1}^2\right] + \alpha_1^2 E\left[a_{t-1}^4\right]) \\ &= 3(\alpha_0^2 + 2\alpha_0\alpha_1 Var(a_t^2) + \alpha_1^2 E\left[a_t^4\right]) \\ &= 3(\alpha_0^2 + 2\alpha_0\alpha_1 \frac{\alpha_0}{1-\alpha_1} + \alpha_1^2 m_4) \\ &= 3\alpha_0^2\Big(1 + 2\frac{\alpha_1}{1-\alpha_1}\Big) + 3\alpha_1^2 m_4 \\ &= 3\alpha_0^2\Big(\frac{1+\alpha_1}{1-\alpha_1}\Big) + 3\alpha_1^2 m_4, \end{aligned}$$

where we can see, that the rhs can only be equal lhs, if $3\alpha_1^2 < 1 \Leftrightarrow \alpha_1 = \sqrt{1/3}$. Otherwise, the fourth moment will be growing. Consequently, solving with respect to m_4, we get the following result

$$\begin{aligned} m_4(1-3\alpha_1^2) &= \frac{3\alpha_t^2(1+\alpha_1)}{(1-\alpha_1)} \quad \Leftrightarrow \\ m_4 &= \frac{3\alpha_0^2(1+\alpha_1)}{(1-\alpha_1)(1-3\alpha_1^2)}. \end{aligned}$$

The m_4 is always positive, that is why the α_1 constraint must satisfy the condition $0 \leq \alpha_1 < \sqrt{\frac{1}{3}}$. The *unconditional kurtosis* of a_t is found by

$$\begin{aligned} \frac{E(a_t^4)}{[Var(a_t)]^2} &= 3\frac{\alpha_0^2(1+\alpha_1)}{(1-\alpha_1)(1-3\alpha_1^2)} / \frac{\alpha_0^2}{(1-\alpha_1)^2} \\ &= 3\frac{\alpha_0^2(1+\alpha_1)}{(1-\alpha_1)(1-3\alpha_1^2)} \cdot \frac{(1-\alpha_1)^2}{\alpha_0^2} \\ &= 3\frac{(1+\alpha_1)(1-\alpha_1)}{1-3\alpha_1^2} = 3\frac{1-\alpha_1^2}{1-3\alpha_1^2}. \end{aligned}$$

It is fairly easy to see, the unconditional kurtosis $3\frac{1-\alpha_1^2}{1-3\alpha_1^2} > 0$, is bigger than the kurtosis of the standard normal distribution. Therefore, we conclude the ARCH(1) process, do generate data with heavier tails than the normal distribution.

Luckily, the above mentioned mathematical properties for ARCH(1) continue to hold for higher order ARCH(m). However, mathematical formulas often difficult, and if not impossible to derive explicitly. In these situations, modern statistical parameter estimation might serve as a reasonable approximation to an "exact" solution. Note that for simplicity of this thesis, we only take a closer look at most simple ARCH(m) models. In spite of that, we have heard that successive extension to more complex ARCH(m) models, should be fairly straightforward. Though, this task is left as an optional hands-on exercise to the interested reader.

Nevertheless, it is still though interesting to see that the specification of a Gaussian GARCH(m, n)

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{m} \alpha_i u_{t-i}^2 + \sum_{i=1}^{n} \beta_i \sigma_{t-i}^2$$

looks like, the ARCH(m) specification when $n = 0$ in eqn **??**. fuckhead

2.2.2 Markov-Switching ARCH(1) Model

In the following short discussion, we want to briefly introduce a simple Markov-Switching ARCH(1) model. The reason why we do it here, is mainly because of the interconnectivity of Markov regime switching and an explicit representation of a likelihood function. The reader may want to skip this chapter, at first, and then come back after reading Section 6.

Imagine, we have an excess return series $r_t = \sigma_t \epsilon_t$, where $\epsilon_t \sim N(0,1)$ are independent and identically distributed standard normal random variables, with variance is $\sigma_t^2 = \alpha_{0,S_t} + \alpha_{1,S_t} r_{t-1}^2$. Notice that the non-constant (i.e. varying) conditional variance of r_t given r_{t-1} is Gaussian $r_t \mid r_{t-1} \sim N(0,\ \alpha_0 + \alpha_1 r_{t-1}^2)$. The latent Bernoulli variable $S_t \in \{0,1\}$ at time t is a historical sequence of states $\{S_t\}_{t=1}^T$, and it represents the unobserved (hidden or embedded) true state of the system. We assume that, the state probabilities are constant over the sample period, and S_t follows a first-order Markov process for state transitions. In other words, the transition probabilities are defined by the current state of S at time $(t-1)$. For a convenience representation, let the upper-case S denote a random state variable, and the lower-case s refer to a particular realization, then we can express it more informative, as $P(S_t = s_t \mid S_{t-1} = s_{t-1}) = P(S_t = s_t \mid S_{t-1} = s_{t-1}, \ldots, S_0 = s_0)$.

Now, we can illustrate an economic example for this simple model. Suppose that there is a data analyst who is interested in a stock's implied volatility of respective returns. Then let a Markov chain with a two-state space process represent a sequence of various successive realisations, that we can write as following transition probabilities $p_{ij} = P(s_t = j \mid s_{t-1} = i)$. That is, the process characterizes a switch, from a regime j-th to a regime i-th, and vice versa (i.e. we assume reversibility here). Hence, we can define a general conditional probability matrix $\underline{\underline{P}}_{2\times 2}$ like this

$$
\begin{aligned}
Pr\left[S_t = 0 \mid s_{t-1} = 0\right] &= p \\
Pr\left[S_t = 1 \mid s_{t-1} = 0\right] &= 1 - p \\
Pr\left[S_t = 1 \mid s_{t-1} = 1\right] &= q \\
Pr\left[S_t = 0 \mid s_{t-1} = 1\right] &= 1 - q
\end{aligned}
$$

In this particular case, we can choose to associate the state $i = 0$ to be a low-volatility regime, and the state $i = 1$ to a high-volatility regime. Accordingly, p_{00} is the probability of remaining in the low-volatility regime at time t, when the state of the system was in the low-volatility regime in the previous period $t-1$. Consequently, the probability of switching from low- to high-volatility regime is p_{01}, the probability of remaining in a high-volatility regime is p_{11}, and finally the probability of moving out high- to low-volatility is p_{10} .

The mathematical properties for the conditional probabilities p_{ij} in the transition matrix $\underline{\underline{P}}_{k\times k}$ for a respective Markov chain always in the interval $0 \le p_{ij} \le 1$ for $1 \le i,j \le k$, and $\sum_{j=1}^{k=2} p_{ij} = 1$, for $1 \le i \le k$. We further assume that $\underline{\underline{P}}_{k\times k}$ is irreducible with finite state space $S = \{1,\ldots,k\}$, so that we can find an initial probability distribution $\phi_0(i) = Pr\left\{S_0 = i\right\}$.

We can define the joint conditional distribution as a product of

$$
f(r_1,\ldots,r_T \mid S_1,\ldots,S_T) = \prod_{t=2}^{T} f(r_t \mid r_{t-1}, S_t) f(r_1 \mid S_1).
$$

Thus, under the distributional assumption the conditional normality gives following (i.e., intuitively looks very Gaussian) density

$$f(r_1, \dots, r_T \mid S_1, \dots, S_T) = \prod_{t=2}^{T} \frac{1}{\sigma_t \sqrt{2\pi}} \exp\left(-\frac{r_t^2}{2\sigma_t^2}\right) f(r_1 \mid S_1), \qquad (4)$$

where a corresponding joint density is given by

$$f(r_1, \dots, r_T) = f(r_1, \dots, r_T \mid S_1, \dots, S_T) f(S_1, \dots, S_T),$$

is a product of various conditional densities $\{r_t\}_t^T$ given respective $\{S_t\}_t^T$, and a corresponding marginal density for $f(\{r_t\}_t^T)$. We thus see that our *likelihood function* for the randomly observed stochastic process r_t, can be represented as a summation of products

$$f(r_1, \dots, r_T) = \sum_{\mathbf{S} \in \{0,1\}^T} f(r_1, \dots, r_T) f(S_1, \dots, S_T),$$

where $f(S_1, \dots, S_T)$ is a Monte Carlo probability function. In a discrete variables case (i.e. same concepts can be transformed to a continuous case) we express

$$f(S_1, \dots, S_T) = \prod_{t=2}^{T} f(S_t \mid S_{t-1}) f(S_{t-1}).$$

Summarizing the above, the resulting conditional likelihood function

$$L(\alpha_0, \alpha_1 \mid r_1, \dots, r_T; S_1, \dots, S_T) = \prod_{t=2}^{T} f_{\alpha_0, \alpha_1}(r_t \mid r_{t-1}, \dots, r_T)$$

where $\underline{\alpha} = (\alpha_0, \alpha_1)$ exponentially distributed with the respective conditional densities $f_{\underline{\alpha}}(\cdot|\cdot)$.

Last but not least, at this point the natural question is how to estimate these quantity values of the latent parameters p_{ij}, which govern the process of the transition between states i and j

$$\begin{aligned} &Pr\{S_t = 1 \mid s_{t-1}\}^{S_t} \cdot Pr\{S_t = 0 \mid s_{t-1}\}^{1-S_t} \\ &\qquad = \begin{cases} p_{01}^{S_t} \cdot p_{00}^{1-S_t} & \text{if } S_t = 0 \\ p_{11}^{S_t} \cdot p_{10}^{1-S_t} & \text{if } S_t = 1 \end{cases} \end{aligned}$$

which namely are the respective transition probabilities in a hidden Markov Chain.

For more interesting discussions and further modelling specifications, please see (Cai, 1994, p. 310), (Durland and McCurdy, 1994, p. 280), (Lawler, 2010, pp. 10–16), (Li and Lin, 2003, p. 125), and (Tsay, 2010, pp. 280–286).

2.3 Data Set

We use Yahoo's data service `www.quote.yahoo.com` to download our time series. Originally, we have downloaded 25 randomly selected blue chip stocks. But, we choose to continue only with two of them. The Bank of America (BAC:NYSE) and General Electric Company (GE:NYSE). We believe that the mathematical properties of these stocks, can be categorized (according to Engle (1982)) as ARCH(m) stochastic processes.

2.3.1 Data Specifications

Our sample period is from January 2000 to April 2014, where the sample's size is $T = 1867$ of daily adjusted closing prices (i.e., quotes are adjusted for possible dividends and splits). From Figure 1 we see that, even though the stocks come from two different financial sectors, they seem to be very well correlated. The values of the stocks drastically decrease from 1992-2001 and 2009-2012. Were there some kind of financial crises during these two periods? This question is beyond this thesis.

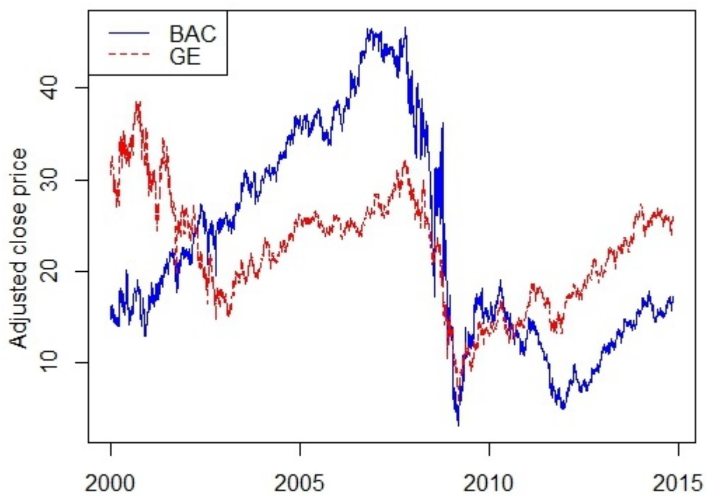


Figure 1: Time plot of daily stock returns (in US dollar) of BAC and GE, from January 2000 to November 2014. The blue line represents Bank of America, and the dashed red line represents General Electric Inc.

2.3.2 Data Properties

In Figure 2 and 3 we see BAC's average price is 22.943 USD, and it fluctuates in range of 3.053 USD and 46.582 USD. The average log return is 0.0000245, and the log returns fluctuates in range of -0.3420578 and 0.3020965.

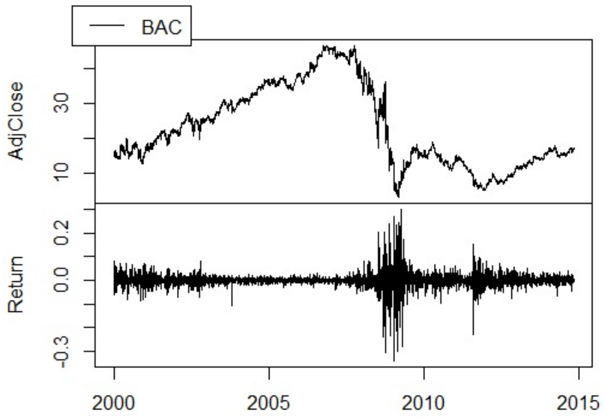


Figure 2: Upper sub-panel is the time plot of daily stock returns (in US dollar) of GE from January 2000 to November 2014. Lower sub-panel represents the respective log stock returns.

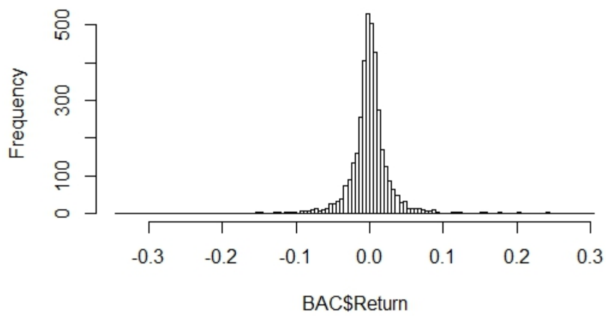


Figure 3: Histogram of the log returns for Bank of America.

In Figure 4 and 5 we see GE's average price is 21.855 USD, and it fluctuates in range of 5.479 USD and 37.866 USD. The average log return is -0.0000584, and in between -0.1368414 and 0.1798441.

A normal quantile-quantile (henceforth, Q-Q) plot shows a relation of residuals against fitted values. The Q-Q plots, for both stocks, reveals some pretty heavy tails. This indicates that that the data frames that we are looking at, definitely are not normally distributed. Luckily for us, this is actually desirable in our case, because ARCH(m) processes do have heavier tails than a normal distribution, cf. 2.2. We proceed with our analysis.

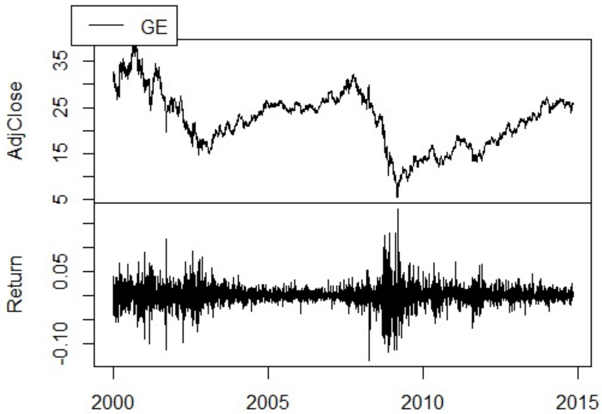


Figure 4: Upper sub-panel is the time plot of daily stock returns (in US dollar) of GE from January 2000 to November 2014. Lower sub-panel represents the respective log stock returns.

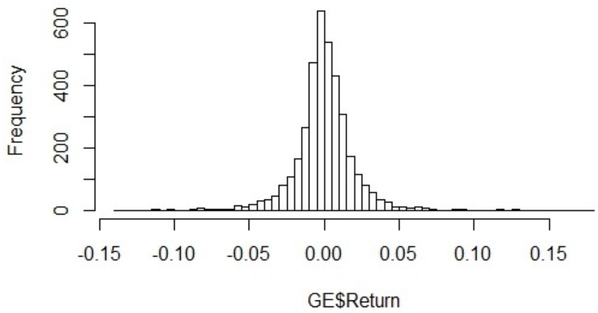


Figure 5: Histogram of the log returns for General Electric.

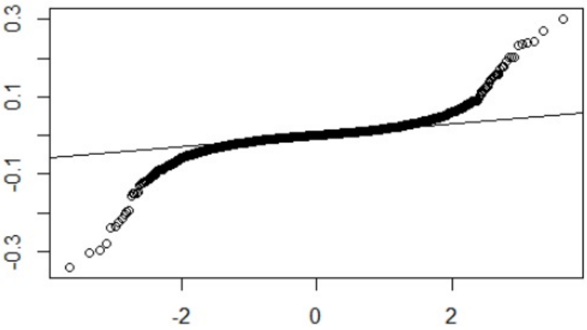


Figure 6: The Normal Q-Q plot for BAC.

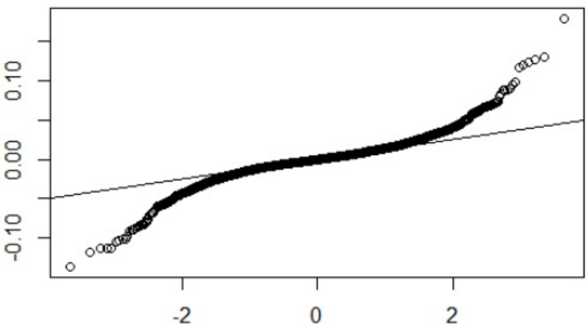
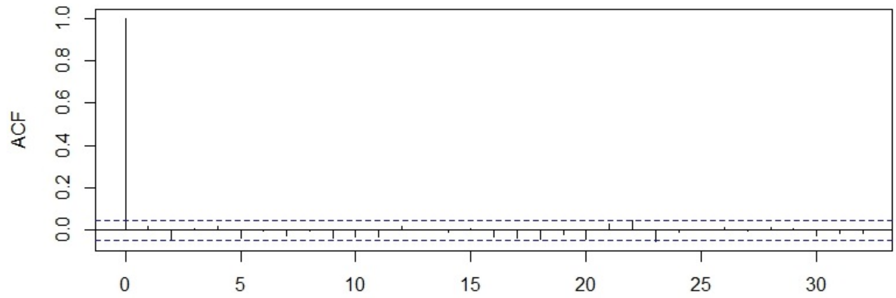


Figure 7: The Normal Q-Q plot for GE.

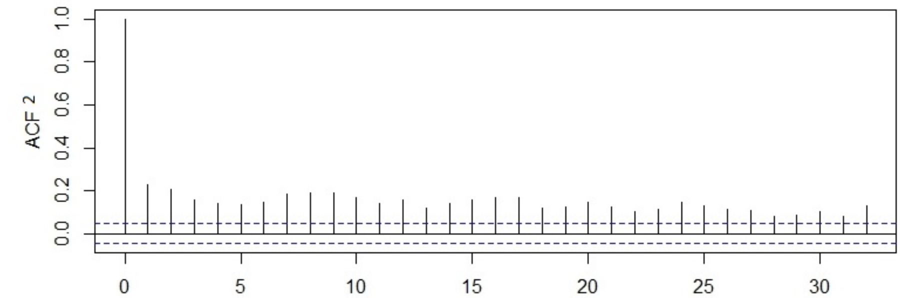
2.3.3 Identifying ARCH Process

The identification (or testing) of ARCH(m) process is a two-step procedure. The first step, is to verify if the time series has no (or, in some minor order) serial correlation between its past values. The second step, is to confirm that these series, actually have conditional heteroscedasticity in one of its sub-populations.

In Figure 8a and Figure 9a, that both series have no serial correlations (i.e., the log returns are within the two standard-error lines, on the magic 5% significance level).



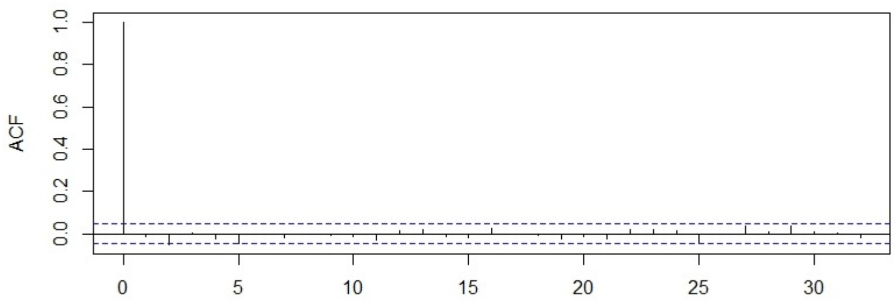
(a) The sample autocorrelation function of the log return series for BAC



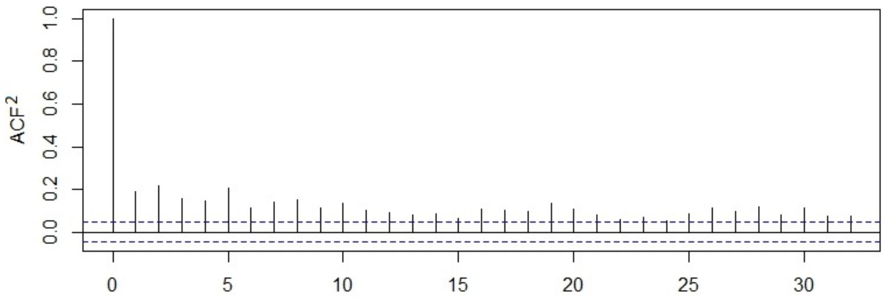
(b) The sample ACF of the squared log returns series for BAC

Figure 8: The sample ACF of two various functions of daily log stock returns for Bank of America, where the sample period is from January 2000 to April 2014.

In Figure 8b and Figure 9b, we see that there is somewhat exponential decay pattern, for both series. Although, the patterns are not like a perfect textbook example of an exponential decay pattern, we choose to proceed with testing our null hypotheses.



(a) The sample autocorrelation function of the return series for GE



(b) The sample ACF of the squared log returns series for GE

Figure 9: The sample ACF of two various functions of daily log stock returns for General Electric, where the sample period is from January 2000 to April 2014.

2.3.4 Testing for Autocorrelation

The Ljung-Box $Q(m)$ statistic for BAC log returns is $Q(8) = 9.2565$ with p-value 0.3211. The test is not significant (on a 5% level), and we cannot reject our hypothesis that there is no serial correlation in the time series. The Ljung-Box $Q(m)$ statistic for GE log returns is $Q(8) = 10.9671$ with p-value 0.2036. The tests is not significant, and we cannot reject our hypothesis that there is no serial correlation in the time series. Since, the series do not have any significant serial correlations, we continue to testing for ARCH Effect.

2.3.5 Testing for ARCH Effect

Once again, we use usual Ljung-Box $Q(m)$ statistics, but this time to the squared time series r_t^2. The Ljung-Box $Q(m)$ statistic for BAC has p-value $< 2.2e - 16$. The test is significant, and we conclude that BAC has some conditional heteroscedasticity. The Ljung-Box $Q(m)$ statistic for GE has p-value $< 2.2e - 16$. We find this test, also to be significant, and we conclude that GE has some conditional heteroscedasticity effect.

3 Bayesian Statistics

3.1 Introduction

Reverend Thomas Bayes (1701 - 7 April 1761) was an English statistician, philosopher and Presbyterian minister. Bayes by strong mathematical reasoning proved the latter famous Bayes' theorem. His work was posthumously published in 1763 by his friend Sir Richard Price. Some time after, independently of the publication, a French mathematician and astronomer Pierre-Simon Laplace generalized Bayes' theorem (but mostly in words). Price and Laplace have met in Paris, and talked together, and Laplace did recognized that Bayes was the first who mathematically have proved the theorem. Laplace was the pioneer in his vision and have popularized Bayesian probability even further [2]. Many years later, or more precisely in the early 1990s, Bayesian inference reached mainstream statistics.

Bayesian statistics is an alternative approach to the classical frequentist approach, that most students have to learn in both graduate and undergraduate statistics courses. The main difference, is in the philosophical view and aspects of Universe and its mutual interconnectivity. Thus, the Bayesian statisticians believe that a world they choose to model, has dependent probabilities. Such that, these conditional probabilities, somehow and in some way, always conditioned on something. This assumption makes it possible, to update a subjective knowledge and experience about a phenomenon, by recalculating personal a priori probability density function.

The general nature of a Bayesian framework is a systematic updating statistical technique of various conditional probabilities. We think that the one of the most fascinating thing about this theorem is that tedious repetitive calculations, that in most cases very difficult and if not impossible computational tasks, now can be (to a certain degree) easily calculated with use of a modern laptop computer. We believe that Bayes statistics and use of it, will only become more and more popular in the future, with further growth in modern computer processing power.

To gain a better understanding and more profound insight into Bayesian statistics, the interested reader might want to consult, for example (Lee, 2004, chap. 2.1) or (Brandimarte, 2014, pp. 241-243).

3.2 Bayes' Theorem

Bayes' theorem (also known as Bayes' rule, Bayes' law, or Bayes' formula) is one of basic logical reasoning pillars in both probability and statistics. Alternatively, formulated and often referred to as a *theorem on the probability of causes.* Although Bayes' theorem easily can be deduced with a pencil on a piece of paper (Schiller et al., 2009, pp. 7-17), we do believe that it very well may take several years to truly understand the theorem and various applications of it.

[2] http://en.wikipedia.org/wiki/Thomas_Bayes

Suppose there is a sample space S with only two but mutually exclusive events, i.e. A and B such that $P(B) > 0$. Furthermore, let $P(A \mid B)$ denote respective conditional probability of A, given that B is known and has occurred. Since we can observe a value of A when event occurs, we can update the original S_t with a new sample space S_{t+1}. Hence, we can write $P(A \mid B) = P(A \cap B)/P(B) \Leftrightarrow P(A \cap B) = P(B)P(A \mid B)$. Due to symmetry, we can write same differently $P(B \mid A) = P(A \cap B)/P(A) \Leftrightarrow P(A \cap B) = P(A)P(B \mid A)$. Because of the equality we can write $P(A \cap B) = P(A)P(B \mid A) = P(B)P(A \mid B)$. Thus by dividing with $P(B)$ we can obtain Bayesian theorem

$$P(B \mid A) = \frac{P(A)P(A \mid B)}{P(B)}.$$

The denominator $P(B)$ can be calculated (mostly in simple cases) by using the law of total probability. In practise, in some financial applications, the event B is a result in one of mutually exclusive events $A_1, A_2, \ldots, A_T$. That is, in a discrete case can be written as

$$\begin{aligned} P(B^*) &= P(A_1)P(B \mid A_1) + \cdots + P(A_T)P(B \mid A_T) \\ &= \sum_{t=1}^{T} P(A_t)P(B \mid A_T) \end{aligned}$$

It is worth noting that, in most simple cases we can integrate $P(B^*)$ analytically, but in more complex cases (i.e., multidimensional state spaces) we can only simulate an approximative solution by using e.g. Markov Chain Monte Carlo techniques.

3.3 Posterior is prop to prior times likelihood

Consider the problem where a stock analyst observes daily returns for various publicly traded stocks. Sooner or later, the stock analyst may realize that some of the stocks have clear characteristic repetitive cyclical patterns (due to e.g. a seasonality within a seasonality). Based on relevant data, solid mathematical apparatus and a good portion of scientific epistemological intuition, the stock analyst will eventually generate own trading ideas and investment strategies. Such that, gradually through time, an investment philosophy will be born, evolved and various trading rules defined and written.

Now, without further ado, let the so-called trading rules, mathematically very well fit into the following pattern. Furthermore, assume that these trading rules can be represented with a random vector of some exogenous variables. We can denote this latent vector θ of unknowns as

$$\theta = (\theta_1, \theta_2, \ldots, \theta_k),$$

where $k \geq 1$ is the number of respective parameters. In other words, θ represents quantity-values of unknown implicit parameters, (which usually either integers and/or real numbers), that we wish to estimate, because these parameters characterize a distribution of interest for the respective stock returns. This is the main reason why, we are so interested in estimating θ, as it characterizes pdf for a random sample of X of returns. Suppose, that we can write the vector X of random variables as a sequence

$$X = (X_1, X_2, \ldots, X_T),$$

where $T \geq 1$ is the total number of observations in a data set series. In other words, the respective density function of a stock quotes is dependent on θ_k latent parameters.

3.3.1 Prior Beliefs

Assume that the analyst, in advance, makes various observations on a stock market, before even s/he have collected any evidence and relevant data. We can express this subjective a priori knowledge as pdf

$$p(\theta) = p(\theta_1, \theta_2, \ldots, \theta_k),$$

which is also called a priori distribution, that expresses the analyst's beliefs about parameters in the $p(\theta)$ of θ.

Now, here it is worth noting that the more empirical data becomes available to the stock analyst, the less weigh her/his prior beliefs. But then again, as the Oracle of Omaha Warren Buffett says: "We are all created equal, but we do not all have an equal opportunity." In other words, in context of this thesis, in real life "big data" costs money and not every stock analyst can afford it.

3.3.2 Likelihood Function

The likelihood function $L(\theta \mid X)$ measures how likely is an observation i in $X = \{r_i\}_{i=1}^{T}$ for a given value of parameter θ. Hence, by using likelihood function, we can express the conditional density function $f(X \mid \theta)$ as

$$\begin{aligned} L(\theta \mid X) &= f(X \mid \theta) = f(X \mid \theta_1, \theta_2, \ldots, \theta_k) \\ &= f(r_1, r_2, \ldots, r_T \mid \theta_1, \theta_2, \ldots, \theta_k) \\ &= f(r_1 \mid \theta) f(r_2 \mid \theta) \cdots f(r_T \mid \theta), \end{aligned}$$

where X is a sequence of T observations/realizations with known values, that are conditionally on θ.

3.3.3 Posterior Distribution

We are interested in a distribution of unknown θ given observed $\{r_i\}_{i=1}^T$. We estimate a distribution of $\{r_i\}_{i=1}^T$ for known fixed θ. Hence, we want to invert a conditioning on θ to conditioning on $\{r_i\}_{i=1}^T$. Hence, we use Bayes' theorem and merge the respective likelihood and prior beliefs, such that we get

$$f(\theta \mid r_1, r_2, \dots, r_T) = \frac{1}{Z} f(\ r_1, r_2, \dots, r_T \mid \theta) p(\theta),$$

where Z is a normalizing constant. By applying the total probability theorem, for given $f(X \mid \theta)$ distribution and $p(\theta)$ distribution of prior beliefs, we can find a normalizing constant Z, as the marginal density for $X_1, \dots X_T$

$$Z = \int_{\Theta} f(\{r_t\}_{t=1}^T \mid \theta) p(\theta) d(\theta),$$

where we integrate over area where θ is well defined in domain Θ. Since θ is independent of θ, it does not change the shape of the distribution, and we can disregard from it. For more convenient notation we can write

$$p(\theta \mid D) \quad \propto \quad p(D \mid \theta) p(\theta),$$

where the $p(D \mid \theta)$ is the prior probability distribution $p(\theta)$ of θ. This is just an epistemological intuition that a scientist decision/inference biased to believe about the prior beliefs. For more memorable version of the theorem we can write

$$\text{posterior} \propto \text{prior} \times \text{likelihood},$$

in other words says that we merge likelihood with prior knowledge of observed data.

3.4 Our Proposal Density

We now know that ARCH(1) processes have unknown but positive parameters. So, based on our prior knowledge we can think of two appropriate choices of a priori distributions.

3.4.1 Our Choice of Prior Distribution

In the first case, we can write our a priori distribution, as the product of two independent exponential distributions for both α-parameters

$$p(\alpha_0, \alpha_1) \quad = \quad \frac{1}{\beta_0} \exp{(-\frac{\alpha_0}{\beta_0})} \frac{1}{\beta_1} \exp{(-\frac{\alpha_1}{\beta_1})}, \ \alpha_0, \alpha_1, \beta_0, \beta_1 > 0 \quad (5)$$

where α_0 and α_1 are the two unknown parameters, and β_0 and β_1 are their respective means.

We choose to use this a priori distribution in our numerical implementation of ARCH(1) model.

3.4.2 Alternative Choice of Our Prior Distribution

In the second case, we can write an alternative a priori distribution, where we can choose to believe the ARCH(1) process has a variance. In that particular case, we know that $\alpha_1 < 1$. That is, the alternative a priori distribution must satisfy $\alpha_1 > 0$ and $\alpha_1 < 1$. Hence, the a priori distribution is a product of an exponential distribution for α_0, and uniform distribution $U \sim [0, 1]$ for α_1, is given by

$$p(\alpha_0, \alpha_1) \quad = \quad \frac{1}{\beta_0} \exp{(-\frac{\alpha_0}{\beta_0})} \cdot 1, \text{ where } \alpha_0 \text{ and } \beta_0 > 0.$$

It can be a possibility to use this the alternative a priori distribution in the numerical implementation of ARCH(1) model.

3.5 Our Posterior Distribution

Okay, perfect... progress!.. Finally, we figure out how to we can derive our posterior distribution. By rearranging equations 2 and 3 (where the unknown parameters α are conditioned on a stock return in a times series), we get the following result

$$\pi \left(\alpha_0, \alpha_1 \mid r\right) = \prod_{t=2}^{T} \frac{1}{\sigma_t} \exp\left(-\frac{r_t^2}{2\sigma_t^2}\right) \frac{1}{\beta_0} \exp\left(-\frac{\alpha_0}{\beta_0}\right) \frac{1}{\beta_1} \exp\left(-\frac{\alpha_1}{\beta_1}\right).$$

The respective *log-likelihood (lik)* function, where we are disregarding from a (probably unknown) normalizing constant. Our lik is

$$\begin{aligned} \log\left(\pi\left(\alpha_0, \alpha_1 \mid r\right)\right) \quad &= \quad \sum_{t=2}^{T} \log \frac{1}{\sigma_t} \exp\left(-\frac{r_t^2}{2\sigma_t^2}\right) + \log\left(\frac{1}{\beta_0} \exp\left(-\frac{\alpha_0}{\beta_0}\right)\right) \\ &\quad + \log\left(\frac{1}{\beta_1} \exp\left(-\frac{\alpha_1}{\beta_1}\right)\right) \\ &= \quad \sum_{t=2}^{T} \left(\log \frac{1}{\sigma_t} + \frac{r_t^2}{2\sigma_t^2}\right) + \log\left(\frac{1}{\beta_0}\right) - \frac{\alpha_0}{\beta_0} \\ &\quad + \log\left(\frac{1}{\beta_1}\right) - \frac{\alpha_1}{\beta_1} \\ &= \quad \sum_{t=2}^{T} \left(\log \frac{1}{\sigma_t} + \frac{r_t^2}{2\sigma_t^2}\right) - \frac{\alpha_0}{\beta_0} - \frac{\alpha_1}{\beta_1}. \end{aligned} \tag{6}$$

After, we have been looking at our posterior for quite some time, we are pretty much convinced that our deduced distribution has no explicit solution (at least to us known).

3.5.1 Full Conditionals

In addition to the posterior distribution, we take a brief look at two full conditionals. Such that, the first full conditional for π_{α_0}, that is conditioned on α_1 and r is represented by

$$\pi_{\alpha_0}(\alpha_0 \mid \alpha_1, r) \propto \prod_{t=2}^{T} \frac{1}{\sigma_t} \exp\left(-\frac{r_t^2}{2\sigma_t^2}\right) \frac{1}{\beta_0} \exp\left(-\frac{\alpha_0}{\beta_0}\right),$$

whereas, the second full conditional for π_{α_1}that is conditioned on α_0 and r is given by

$$\pi_{\alpha_1}(\alpha_1 \mid \alpha_0, r) \propto \prod_{t=2}^{T} \frac{1}{\sigma_t} \exp\left(-\frac{r_t^2}{2\sigma_t^2}\right) \frac{1}{\beta_1} \exp\left(-\frac{\alpha_1}{\beta_1}\right).$$

Similarly, as with our posterior distribution, we simply cannot recognize the full conditionals. Intuitively, we are now convinced that neither our posterior or the full conditionals, can be solved explicitly. Unfortunately for us, it looks like we have to get our hands dirty, and do some statistical computations and numerical programming. Alas, first we have to learn more about Markov Chain Monte Carlo methodology. We begin our study with Markov chain theory.

4 Markov Chain Theory

4.1 Introduction

Andrey Andreyevich Markov (1856 - 1922) was a Russian and later a Soviet mathematician. He was a student of another prominent Russian mathematician Pafnuty Lvovich Chebyshev (1821 - 1894) [3]. Pafnuty Lvovich is famous for his mathematical contribution in the fields of probability, statistics, mechanics, and number theory [4]. Andrey Andreyevich is best known for his research in stochastic processes, and to be more precisely, i.e. Markov chain theory. The surname Markov in older works also spelled as Markoff. The oldest source of this kind of spelling that we could verify was found in (Metropolis and Ulam, 1949, p. 341).

Now, as sure as we are and as sure as we can be, such that as much as we want to be Bayesians at the current moment. We are not sure, but we must have heard and/or read somewhere (also during our thesis research process) that the work in stochastic processes of Andrey Andreyevich was inspired by novel in verse "Eugene Onegin" that was written by one of he greatest Russian poet, author and writer Alexander Sergeyevich Pushkin (1799 - 1837) [5]. By the way, various excellent partial translations of Pushkin's poem can be found at Peter M Lee's homepage [6]. We have not been able to find the original work

[3]https://en.wikipedia.org/wiki/Andrey_Markov
[4]https://en.wikipedia.org/wiki/Pafnuty_Chebyshev
[5]https://en.wikipedia.org/wiki/Alexander_Pushkin
[6]http://www-users.york.ac.uk/~pml1/onegin/partial.htm

of Andrey Andreyevich, such that, we could attempt to recreate his original thought processing, although that could be interesting to do that. Nevertheless, since we have been lucky to be "forced" by our primary school teacher [7] to recite of by heart various parts of Onegin, back in old days. We think, or still, as Bayesians qualitatively guessing, that Andrey Andreyevich began to look at the different sequences of consonants and vocals. Then after have been looking in vain at different combination of letters and these letters in words combination, at some point must have zoomed out of the sequence of words, and looked at the whole particular subsections of the poem. The poetry of Pushkin, in our humble opinion, is a close combination of modern hip hop and William Shakespeare's rhyming. It is impossible to recite whole poem if you try to remember all words, key words, pictures, images, etc.... But by remembering phonetics and particular rhythm of subsections, one can recite the poem of by heart for hours. So, that Andrey Andreyevich must have realized that rhythm of next subsection (i.e., future) was independent of all other subsections (i.e., past) except the present subsection (i.e., present). Now, maybe it is not that surprising too see that (hidden) Markov chains are heavily used in speech recognition and digital signal processing as well.

We think that the Markov's research in stochastic processes has contributed a huge impact in modern statistics, where the best of it is there to be discovered.

In this subsection, we will attempt to give relevant exposition of Markov chain theory, that we can justify the use of it in MCMC and hidden Markov models. We will try to keep it short and concise, although not sacrificing rigorous mathematical notation.

4.2 Finite Markov Chains

Assume that we have a d-dimensional continuous state space, where a transition kernel

$$Pr(x, A) =$$

a conditional distribution function is represented with
Fubini et de Finetti,..., logically speaking, thoughts in process

We assume that there exists the unique stationary distribution (the limiting distribution) π, such that the initial distribution (i.e. initial state of a system) a chain $\pi^{(n)}$

Lawler (2010)[pp.]

[7] Nadezhda Nikolajevna Vasilieva in Russian Litterature and Language in school nr. 5, Sibiria, Republic Tuva

4.2.1 Notation

4.2.2 Stationary Distribution

4.2.3 Reversibility of Chains

4.3 Continuous State Space

The continuous time processes case

4.3.1 Transition kernels

4.3.2 Fubini

4.3.3 Fubini

The reader can find more interesting details with rigorous mathematical examples, derivations and proofs, inter alia Chib and Greenberg (1995), Chib (2001)[pp. 3576–3579], Gamerman (1997)[pp. 93-116], and Monahan (2011)[pp. 375-400].

5 Markov Chain Monte Carlo

5.1 The Short Motivation to MCMC

In statistics, or maybe more precisely in statistical modelling, Markov Chain Monte Carlo (henceforth, MCMC) is a very big algorithmic family of various mathematical schemes, computational methods and numerical implementation techniques.

We must admit at once, that there is going on a gigantic research in this vast MCMC field. There are myriads of flourishing academic literature. Such that there are many excellent examples of various pseudo - and implemented algorithms. Various software tools, scientific programming and higher order probabilistic programming languages out there [8], that are now available to any scientist.

We must also admit that we did only a very short, but we must say at the same time quite intense, introduction study to MCMC subject. Gradually it sort of became obvious to us how powerful MCMC techniques are, though we are still in the learning process mode. Nevertheless, as we understand the secret of these MCMC techniques is not only in the rigorous mathematical notation. But also, it is as much in various hands-on numerical challenges of being able to implement on available modern computational facilities.

We think, and we must admit that from now on only subjectively, that MCMC methodology in modern statistics (also in e.g., economics, econometrics, image analysis, machine learning, physics, robotics, social science, etc. . .), is a powerful all-round computational mechanisms that definitely can be used for estimation of a model's latent parameters. Not only that, we also think that MCMC can be used to do

[8] e.g., Anglican, BUGS, Church, Fortran, GAUSS, JAGS, Mathematica, Matlab, OpenBUGS, Python, R, Stan, Venture, WinBUGS, . . .

various strategic Hedge Fund's investments strategies on a volatility's fluctuations, for various publicly traded stocks and financial instruments. Although, to be honest with the reader, we do not really know how to do it, yet. But we have an idea, how it should be done.

We believe, and from now on only subjectively, that we finally became completely Bayesians in our philosophical and statistical points of view. Thus, we now see our numerical world with all available information to us, can and should be modelled with MCMC (even if it is computationally costly and time consuming), mainly because we are believing in future increasing computational power. Thus, in other words, if MCMC techniques wisely used and cleverly implemented, it can certainly be the universal magic hack for finding "exact" solutions to (almost) any physical phenomena (that by the way, cannot be solved with traditional pure mathematics in various branches of nature science and wide variety of situations.

So maybe after all it was not that bad idea to get our hands fully dirty!.. Hence, we with our heads first, jump into Ocean dé MCMC.

5.2 The Not So Short Introduction to MCMC

In this subsection, we will try to give a brief introduction, somewhat short overview and historical background on MCMC. Our subjective discussion is mainly based on Robert and Casella (2011), where it is interesting to read about the inception, revolution and post-revolution era of MCMC and its tremendous impact on modern computational statistics. The reader can skip this subsection without loss of continuity.

The very first rudimentary version of *Monte Carlo* (henceforth, MC) methods, actually was discovered in early 1930s by Italian physicist Enrico Fermi. By the way, who also have invented the analogue device FERMIAC (or the Monte Carlo trolley), which could be used to model neutron transportation in various types of nuclear systems[9]. But since there was no modern electronic computer machinery available at that time, alas he never got any recognition for his visionary work and profound insights in physics.

Nevertheless, it is common to say that the original methodology of MCMC computational techniques, initially was conceived in the secret Los Alamos Scientific Laboratory, Los Alamos, New Mexico during the end of second World War and the beginning of the Cold War's arms race. Although, MC and MCMC have been used in physics for quite some time, its impact first became influential in early 1990s.

The original idea to regular MC methods, can be traced all the way back to the late 1940s. The co-inventors of MC methods were a Polish and later American mathematician Stanislaw Marcin Ulam, and John von Neumann a Hungarian and later American applied mathematician and physicist. At some point, Ulam was looking at an intractable combinatorial computations of winning probability of "solitaire" card game. The main reason was, that the combinatorics and probabilities of the

[9] `http://en.wikipedia.org/wiki/FERMIAC`

game, were much similar in its structure to various problems in physical phenomena. And, with this in mind, von Neumann enthusiastically adapted the idea, and used it on various calculations in thermonuclear and fission problems. At the same time in 1947, Ulam and Neumann have invented MCMC techniques, such as, Bayesian probability inversion modelling and accept-reject techniques.

Soon after, (Metropolis and Ulam, 1949) have published the very first paper about the Monte Carlo method, that presented the idea of using new statistical approach to study integro-differential equations (which are known in the kinetic theory of gases, as the Boltzmann equations). We think that the major breakthrough of the paper was the first mechanized computations of new numerical techniques for approximation of unobtainable "closed form" solutions of various integrals.

So, in other words, in early 1950s, the same group of Los Alamos Laboratory scientists (and plus many prominent others) that were mainly physicists, did some intensive research in mathematical physics. The research, by the way, also known as Top Secret "The Manhattan Project" was sponsored by U.S. military. The main purpose was the development/creation of the first atomic bombs. Which are the Fruits of World War II. During their amazing research process, they managed successfully to simulate MCMC computations, which were combined with the first computer ENIAC[10], they were the first to succeed to generate computer simulations for various MC schemes.

Then, couple of years later, the discovered MC ideas were further studied and presented in the published paper by Metropolis et al. (1953). Metropolis et al. described the modified method of MC integration, where the efficiency was in the improvement of a random walk. It is reasonable to say that the first MCMC method was the Metropolis algorithm (henceforth, MA). The generated numerical calculations were carried out on second Los Alamos MANIAC computer [11]. Believe it or not, it was the big thing back then. Suddenly, it was numerically possible to use so-called fast machines and MCMC to obtain feasible experimental results to statistical mechanical problems, that yet not have been possible to solve before for many-dimensional integrals. Metropolis et al. proved ergodicity, which is sufficient condition for asymptotic convergence towards the stationary distribution of Markov chains. And proposed integrating over a random sampling of points, such that respective stationary distribution of interest would be the Markov chain equilibrium of the state of finite number N of interacting individual molecules. By the way, the "Monte Carlo" name was suggested by Nicolas Metropolis, who was both a physicist and mathematician.

Thereafter, or almost 20 years later the publications by (Hastings, 1970) and his student (Peskun, 1973) have further popularized MCMC methodology. In a sense, they brought MCMC closer to various main-

[10]ENIAC stands for Electronic Numerical Integrator and Calculator, and was built with transistors in February 1946, after three years of construction

[11]MANIAC stands for Mathematical Analyzer, Numerical Integrator and Computer

stream statisticians (i.e., social scientists, data scientists, economists, etc...). So, as mentioned above, MA (that was originally proposed by Metropolis et al.) is a sampling method (on a digital computer) that requires the simulation of a Markov chain sequence with its respective specified stationary distribution. Now, that Hastings has introduced an outline of such a generalized method of the sampling procedure, i.e., a recipe for how to construct and simulate such a Markov chain sequence. Hastings (in his paper) likewise exposed relevant Markov chain theory and presented four examples. He has also discussed a challenging topic, which namely is an assessment of the error of an estimation, produced by using Monte Carlo methods. Then short time after, P. H. Peskun in his Biometrika paper did comparison of two sampling methods that were proposed by Metropolis et al. and Barker. Peskun thus has shown that in a discrete setup, MA sampling method is asymptotically more precise than Barker's independent sampling. We would like to make one more comment that (Peskun, 1981) further provided very intuitive and practical computational guidelines for choosing the transition matrix $\underline{\underline{P}} = p_{ij}$, when using Monte Carlo methods in conjunction with Markov chain sampling.

Then almost a decade later, (Geman and Geman, 1984) have introduced their version of MCMC simulation technique, the alternative approach *Gibbs Sampler* algorithm (henceforth, GS). The algorithm was named after an American physicist Josiah Willard Gibbs (1839-1903) that developed the field of vector analysis and made contributions to crystallography and planetary orbits. Gibbs sampling is a fairly simple and widely applicable MCMC algorithm. It is easy (although in the beginning can be little confusing) to see that GS is a special case of the MH algorithm (Bishop, 2006, pp. 542-543). Now, at this moment, we must note that Gibbs sampling and Metropolis-Hastings algorithm, originate from two different sources, though the mathematical justification via Markov chains is same. So, Donald and Stuart Geman [12] have successfully presented new ideas for a Bayesian restoration solution in image processing problems. They have also introduced the generic description of Gibbs sampling, and proved that like MA, that GS produces a Markov chain with $\underline{\pi}$ as the equilibrium distribution (Geman and Geman, 1984, p. 731).

(Tanner and Wong, 1987) are like Geman and Geman are just one of earlier pioneers of GS. In the article, they have introduced an iterative method for the calculation of posterior distributions. The data augmentation algorithm, with great advantage, can be used in solving maximum likelihood problems.

With all this in mind, as a result, Bayesian inference and MCMC methods have changed entire academic way of thinking and attacking different phenomena and challenges in applied computational tasks. The numerical real-life problems, that could not be solved due to computational limits, now became cracked open like eggs. We agree with (Robert and Casella, 2011) that this can be one of paradigm shifts

[12]The development of BUGS (Bayesian inference Using Gibbs Sampling) software, as early as 1991, was originally initiated by Geman and Geman

that originally defined by Thomas Kuhn. MCMC became a universal key or numerical wrench tool, you (i.e., the reader and the writer) can attack (almost) any numerical problem, even shooting a house sparrow with a cannon. We therefore think that the demonstration of using various sequential mathematical computation steps, that were made with help of the "first" artificial intelligence, laid ground not only to MCMC but also to modern numerical analysis.

The history of MCMC is short, but at the same time long. It is quite interesting history that is full of colourful details, just like one of the paintings of a Dutch painter Vincent van Gogh (1853 - 1890). Thus, without a doubt, we will agree 100 percent with the reader that our *The Not So Short Introduction to MCMC* and its respective chronological description of MCMC events, are insufficiently expressive. As we probably have already mentioned above, there are and were many other prominent and influential research papers concerning MCMC techniques and computations. That is, this is said, then this should not be kept as a secret from the reader, that we do really want to read all available (to us) MCMC nomenclature. And, yes-of-course, we would like to write more about MCMC history and different aspects of it, in the very near future. But, alas, this exciting to-do-on-my-list-task-thing is beyond the scope, in the context of this thesis.

5.3 Thinking MCMC and Bayesian Inference

In section 3 we have introduced our subjective version of Bayesian philosophy of science. We have also tried to expose an introduction to a more detailed picture of an estimation method of various unknown parameters from some observed data series. In plain words, we now can summarize the essence of the MCMC approach, in the following way. The observed posteriori statistical distribution of data observables is actually equivalent to the merge-product (i.e., of iterative nature) of the (subjective nature) prior information and the likelihood of how likely is a phenomenon to occur (on the basis of the current Bayesian belief(s)). So, in Bayesian framework we choose to regard the setup of various unknown parameters (both discrete and/or continuous as random variables, and the respective probability of those random variables, as the prior statistical distribution of interest. So, according to Bayes' theorem, the posterior density is a summary of our knowledge about the parameters, that is merged (i.e., recursively) with a priori empirical evidence.

Now that we are well and heavy armed with the posterior distribution (and its continuously (i.e., recursively iterative iterations) updating technique), we should be able to estimate the unknown parameters of interest. We can do that simply by finding their expected values. And, it is worth noting that there is also possibility to find credible intervals (i.e., Bayesian confidence intervals), and do hypothesis testing of the found parameters. But, there is always that little but in life and modelling. We don't know our normalizing constant in our posterior distribution. Our posterior is proportional to our prior multiplied with likelihood. And, even if we knew the normalizing constant, that is needless to say, the evaluation of multidimensional integrals is very difficult and certainly not easy to compute explicitly. However, there is this another possibility that we have read so much about, in such relatively short time. The key idea is that we can sample from the posterior distribution. This way we do not need to know the normalizing constant at all. Luckily for modellers like us, we utilize MCMC methods as sampling approaches to tackle the this normalizing constant that typically arises in analytically intractable problems ((Brandimarte, 2014)[pp. 629-630, p. 647]).

Imagine (just like in the John Lennon's song) that you can imagine Bayesian Inference from a personal and epistemological rough knowledge point view. The roughness or smoothness of zooming in and out process, can be controlled as with a matrix in multi dimensions. That is, the processed knowledge based on information's data inputs becomes finer as more longer time Nature is being observed. Likewise originally, when looking (gathering data) at an applied model (through finite period of time of usually equidistant steps) and its respective economic features, that specified by various esoteric variables, a modeller can and should (gradually) be able to learn stochastically from her/his simulated data. Now, a computational challenge of MCMC with interaction of modern digital computers, is the applied implementation of Bayesian inference. That is, as an average business student e.g. like the writing writer faces great difficulty and challenge, that is not only in the pure mathematical notation, economical insight and last but not least the numerical implementation of MCMC thinking with its practical Bayesian Inference in it.

We prioritize our frugal time wisely and continue our focus on the numerical journey of MCMC algorithms!..

5.4 Metropolis-Hastings Algorithm

In order to be able to find mathematical sound solutions for various inference problems, we have to and must apply numerical principles of dynamic programming.

The Metropolis-Hastings Algorithm (henceforth, M-H)...

For a more complete version of the M-H Algorithm see, for example work of, (Chib and Greenberg, 1995), (Gamerman, 1997)[chap. 6], Metropolis et al. (1953), (Lee, 2004)[chap. 9.6] and (Prochazka et al., 1998)[chap. 9.4.4].

Note that via Markov chain theory the mathematical proofs for Metropolis-Hastings based algorithms and Gibbs samplers', mathematically justified in the same way (Casella and George, 1992, p. 1).

The generic features of the Metropolis-Hastings algorithm

An astonishing variety of versions of MH there are out there.

5.4.1 Metropolis Algorithm

MA is a basic algorithm

The MH algorithm is the same a Markov chain procedure

The "Metropolis algorithm" produces a Markov chain

$$\{X(t), t = 0, 1, 3 \cdots\}$$

where the equilibrium of the chain is a stationary distribution π (see (Geman and Geman, 1984, p.731).

Random walk Metropolis explores...

The implementation of the algorithm should be straightforward, but in general, can be difficult and very challenging task to do. The key idea that (business as usual) we want to cycle through the numerical procedure, by using for-loops and/or in some kind of combination with while-loop.

5.4.2 Rejection Sampling Description

As far as we understand (Lee, 2004, p. 267) says that, the *acceptance-rejection sampling* is an important method, and a good place to start with Metropolis-Hastings algorithm. Although, this method does not make any use of Markov chain methods, it can be used directly in situations where a posterior distribution has an unknown normalising constant Z. In fact, assume that we have such a density $p(\theta) = f(\theta)/Z$ of interest, where $f(\theta) = x^{\alpha-1}(1-x)^{\beta-1}$ is given by a beta distribution. Now, assume also, that there exist a respective *candidate density* $h(\theta)$, such that we able to simulate samples from. Then, by simulating random variables X from the $h(\theta)$, we can estimate a variate $\tilde{\theta}$ with respective $p(\theta)$. If one have no idea of what $h(\theta)$ should be, then a uniform distribution can be a good initial guess. Let us assume that $h(\theta) \sim U(0, 1)$. A more rigorous description of the acceptance-rejection sampling looks like this:

1. Generate a variate y from $h(\theta)$.
2. Generate a value $u \sim U(0, 1)$.
3. From step 1 and step 2, use generated values u and y to compare an equality $u \leq f(y)/kh(y)$, where $k > 0$ is a known constant.
   (a) Then if the condition in step 3 is true, then accept the variate y, assign it to vector x of accepted values, and return to step 1
   (b) Otherwise, return to step 1.

The algorithm can be implemented with a *for-loop* with a *while-loop* in it. The for-loop goes e.g. $N = 1000$ times, and for every iteration our while-loop calculates all 3 steps until the acceptance of y is happened. When y is accepted, jump out of while-loop, increment for-loop with 1 and begin while-loop again.

We can argue (both in the discrete and the continuous case) that the θ value in N trials is on $Nh(\theta)$ occasions. That is, the number of times that $Nf(\theta)$ retained is proportional to $f(\theta)$.

5.5 Metropolis-Hastings Algorithm

Fubini..., We can summarize the Metropolis-Hastings in the following algorithmic form:

Fubini..., for technically masterful explanations, see for example, Peter M Lee p. 270

Fubini..., something like that: Why do we want to use sampling, and what do we achieve from it?

Metropolis-Hastings acceptance rule

an iterative algorithms that cycles through

for each time step t we compute the acceptance test ratio corresponding $\frac{\text{TargetDist}}{PropDist}$ a wide variety of situations

the MH sampling method is an iterative Monte Carlo procedure

5.5.1 Definition and Properties

The computation task is to find

5.6 Gibbs Sampling

Okay, so why do we want to do sampling?

numerical integration is impossible

To see this, consider the

The computational approach

5.6.1 Introduction

Fubini, thoughts in process...

The reader who is interested in more theoretical details of GS's rigorous mathematical foundation, can for example see (Brandimarte, 2014, chap. 14), Casella and George (1992), (Gamerman, 1997)[chap. 5], (Geman and Geman, 1984, p. 731), (Lee, 2004, chap. 9), (Prochazka et al., 1998)[chap. 9.4.2], (Tanner and Wong, 1987) and (Shumway and Stoffer, 2010, p. 388).

5.6.2 Definition and Properties

The sampling strategy of a Gibbs algorithm, is an iterative scheme, where each respective component updated individually. To illustrate the point, we can imaging a vector with three implicit parameters. That is to say, that we need at least three respective for-loops, where in each for-loop, we generate new step of a random walk and calculate an acceptance ratio.

pre compute invariant quantity

5.6.3 Implementation and Optimization

Fubini, in process...

A vector of parameters

$$\theta = (\theta_1, \theta_2, \dots, \theta_k)^T.$$

θ_{-j} is the vector of parameters with θ_j removed

$$\theta = (\theta_1, \theta_2, \dots, \theta_k)^T.$$

Algorithm Gibbs sampler

- Let $f(\theta)$ be the joint density and $f_j(\theta_j \mid \theta_{-j})$ be the conditional density for component $t = 1, \dots, T$.
- Given the current state θ^k at step k, we generate the next state θ^{k+1}
  - Sample θ^{k+1} tilde $f_1(\theta_1 \mid \theta_2^k, \dots, \theta)$
  - Sample ...
  - Sample θ^{k+1} tilde $f_1(\theta_1 \mid \theta_2^k, \dots, \theta)$

5.7 MCMC Hybrid Algorithms

We choose to implement a Metropolis-within-Gibbs (henceforth, MwG) algorithm. We do that for several reasons that mainly were based on logical thinking. Those thinkings we try to describe here. So, we have a pretty fair, nice and easy, and very simple MCMC set-up. And there are many articles and books that describe MwG in both a theory and an applied implementation of it. But then again, there comes this little but. Our implementation of MwG is pretty simple,

We have a fair simple set-up that is very well describe in many articles and books.

MwG is a hybrid method

A very nice expository introduction to MCMC Hybrid algorithms is also provided by

5.7.1 Metropolis within Gibbs Algorithm

Fubini,... in process

The Metropolis within Gibbs Algorithm (henceforth, MwG)

the algorithm is motivated by the following iterative procedure

Fubini, ..., the model algorithm for our model goes here, this numerical recipe is "guldkorn" from Soeren Feodor Nielsen (i.e., see the paper)

Fubini,...

So, the proposal distributions must generate positive numbers, and this can be achieved if we generate exponential random variates $X \sim exp(\lambda)$ that has density $f(x) = \lambda e^{-\lambda x}, \quad x \geq 0$.

It is worth noting about a rate parameter definition for the exponential distribution definition in R

The algorithm with its algorithmic details, may be summarised formally as follows

5.8 OUR MCMC Computational Results

With a bit of luck, the Lady Luck will smile back to you, too. We have been lucky to be able to calculate and generate following sequential complex numerical and experimental results.
Some thoughts about the practical implementation of the MwG algorithm.

In table FUBINI we have this kind and this kind of thing, so bla bla bla bla bla bla bla bla bla bla bla After, looking long time (i.e., hours and hours of debugging and compilation) at the code and its respective computational results and graphical representations. We can (or maybe must) conclude and agree that the results are wrong. conclude that they are (or must be) partially wrong.

we have realized that our results mathematically do not sound (i.e., just like in music) intuitively right.

Intuitively, based on the available a priori knowledge and nomenclature that we have already read (but not necessarily fully understood). To the reader we confess, that as young Bayesian philosophers, scientists and economical modellers, I must admit that the results that were generated subjectively infer that there is something not right there

we do not understand As the young MCMC modellers, we have encountered a banal trial and error experience in implementation of sound mathematical experimental results.

Nevertheless, especially due to time factor and various weather conditions, as a young Bayesian modeller

we must continue our numerical journey, magic-study-hacking and thesis writing. Such that, we immediately advancing closer to the state-of-the-dynamical-art-simulations that can represent various economic factors.

and like in old stories about some brave Vikings in "big" ships, we continue our sailing of numerical journey

It is worth emphasizing that

5.8.1 Warmup phase

The *warmup phase* is also known as *burn in* time (or more precisely a number of iterations) that the algorithm is warmed up to collect information (i.e., store calculated values into a multi-dimensional matrix or a finite vector)

The probabilistic programming language Stan We adopt the definition of of the *burn in* time probabilistic Stan of a burn-in phase, as a warmup phase. The

The essential feature of the warmup phase

The reader can see further discussions of the warmup phase, e.g.

6 Markov Regime Switching Models

The hidden Markov models, as we now understand it, is basically an extension of Markov chains and/or Markov random fields, that can be represented by graphical probabilistic modelling

Markov models of that can graphically represent various . The (discrete and/or continuous) values of various parameters, that econometricians may be interested, unfortunately most of the times simply is just hidden. It is very difficult for a new Bayesian scientist to figure out on her/his own

observables and hidden (latent) states (factors) can be measured with either discrete or continuous latent variables. The un- and observable component(s) we as modellers can attempt to estimate with probabilistic modelling tools and techniques.

6.1 Introduction

Bayesian analysis of the ARCH(1) model with Markov regime switching model.

A hidden Markov model (henceforth, HMM) is a statistical tool that is based on a Markov model, where we want modelling various stochastic Markov processes that are dependent on latent regime/state levels.

The underlying system is hidden

the hidden system would be either "bull" or "bear" market

Basically when we have an HMM problem then, as a scientist and/or a modeller, we usually interested in three situations. The first one is the evaluation problem, which we amongst can use for isolated (word) recognition. The second situation is the decoding problem, and can be used for e.g. the continuous recognition and data segmentation. And, the third situation is the learning problem (i.e. Viterbi algorithm) of training HMM for some kind of recognition tasks [13].

HMMs are common method in speech recognition, machine learning, FUBINI

We will agree with most of nomenclature that a good introduction to HMM is kindly provided by (Rabiner, 1988). We immediately begin our study of HMM with this article. It seems like nomenclature agrees that academic can agrees on the best introduction to exhaustive

[13] `http://jedlik.phy.bme.hu/~gerjanos/HMM/node6.html`

6.2 Literature Review

(henceforth, MRSM)

,as we now discuss.

6.3 MSM Structure, Properties and Methods

The prominence of Markov regime switching numerical modelling is greater than ever. The computational power of fast machines of early 1990s can be easily achieved with ordinary personal computer in 2015. It is even possible to lease online computational power nowadays to do some fast and heavy computations for numerical MSM modelling.

For the sake of a logical, computational and numerical tractability, we briefly define and discuss following assumptions and mathematical details.

The key idea of of Markov Regime Switching model's technical specifications is FUBINI

$\theta = (\mu_{s_t}, \sigma_{s_t}, \alpha_0, \alpha_1, p_{11}, p_{22})$ where $s = 1, 2$ denotes number of states.

Fubini, in process,... Markov switching model

We wish to draw the reader's attention to...

To illustrate these ideas, let us

The forward-backward algorithm that is also known as the Baum-Welch algorithm.

HMM is a statistical tool that can be used for modelling...

Fubini, the latent states of the system is governed by by a Markovian process... fubini

We choose to differentiate between two different periods - namely, regime states of high volatility and low volatility.

to capture regime change

Intuitively it makes sense that the higher the order of HMM the higher the complexity of model building.

Imagine that we have a dependent variable, and in our case it is equivalent to a stock's return. Suppose that this dependent variable follows ARCH(1) stochastic process that have s characteristic of switching between various latent (discrete/continuous) regimes.

we can attempt to estimate the parameters of hidden Markov models

HMM

As modellers we utilize HMM to do Bayesian inference of various latent states.

and describe the past of a sequence of various observables, sun that we can parametrize model, based on evolution of that latent states evolve

The parameters of HMM that we are interested in, can be obtained from the transition and emission matrices.

conventional hidden Markov model (with only two possible states)

We choose to utilize an implementation for modelling Markov Switching Models, using the Stan's Hamiltonian algorithm.

So, if we want to detect the specific switching component between states, then we have to do statistical sampling to estimate various discrete latent regimes. We can e.g. sample from the transition matrix, which we can also use to do Bayesian inference about nature of the state probabilities.

6.3.1 Forward Algorithm

6.3.2 Backward algorithm

So basically, backward algorithm is doing same calculations as forward algorithm does. The only difference that instead of starting in the beginning of the array we begin with the next last element of the array (i.e., $T - 1$).

6.3.3 Viterbi's Algorithm

As the modellers we may be interested in imputations of the most probabilistic path the "waves" of hidden states come in. Imagine that there is a two dimensional stage, that can be represented by "bull," and "bear" markets. That is if the sequence of regimes can be inferred from data set then one may statistically not beat the market by predicting the order of the sequence. But maybe the duration of the respective regimes. In a way, it is like with the stock volatility, but then again it's not the amplitude of the volatility, but it is the speed one may be interested in. Then To calculate the evolution of the order of the states, we can deduce which latent state is most dominant in the event array. Bull Bull Bull Bull, Bear, Bull, Bear, Bull, Bull, etc. . . The Viterbi's algorithm is the tool that we can utilize in this situation.

that we may be interested in imputations of the most probabilistic path for this

for every δt

that we may be interested in the most probable path that we cycle through

the smoothed probabilities calculation routine. . .

6.4 ARCH specifications with Changes in Regime

Fubini, see Hamilton and Susmel article for further model specifications. . .

More fundamentally,

Hence, by using structural breaks due to some exogeneous factors, we are interested in capturing the dynamics of the dataset.

7 HMM with Application to Stock Data

In this section, we present and discuss our experimental results. We reflect on our numerical programming techniques and strategic implementation strategies. To be honest with the reader, when mathematically speaking, we find statistical modelling and numerical computing an intellectually quite challenging task. In some way or another, we think of it as a funny, but at the same time very tricky business. Such that, we fully agree with (Team, 2015, p. 177) that no matter how experienced modeller is, s/he can (and maybe should) always run into problematic and ill-behaved posteriors in practise. Without further ado, let us take a closer look at our generated world of numbers.

econometric literature

as a programmer we can obtain great deal of flexibility in handling big data computations and other information

We want to do the Bayesian data analysis of stochastic ARCH(1) process with MSM.

"Computational tools are an inescapable component of modern economic research. While there is an argument to be made that economists should focus on their comparative advantage (and not on programming), I believe that when it comes to coding, most of us remain in the region of increasing returns." by Chad Fulton [14]

7.1 Definition and Properties

Suppose that we can imagine following specifications of the HMM model, such as a dependent variable is an observation of a respective state. Suppose that we can imagine a dependent variable that follows ARCH(1) stochastic process with regime (also state) switching states of nature.

7.2 Parameter Estimation and Empirical Results

The results suggests that,

Initialization Problem

R is an open source project that was developed by Robert Gentlemen and Robert Ihaka.

optimal estimates of latent state variables

numerical illustrations

7.2.1 Overview and Data Preparation

Fubini, in process...

We perform following powerful and versatile numerical sequence of computations.

We run 10,000 runs with 2,500 warmup steps, with 4 chains total, where we are thinning out in every 5th step. Using FUBINI which have yielded satisfactory numerical results for (i) sampling from the distribution XXXL, (ii) estimation of hyper-parameters and (iii) the respective standard errors, blah blah blah

But as a matter of fact,

[14] pages.uoregon.edu/cfulton/code.html

7.2.2 Model Estimation

Fubini, in process...

In the actual simulation, we will only consider case where we focus FUBINI

7.2.3 Results and Analysis

After some reflections, however, we believe that the numerical results...

regime latent indicator

7.3 Convergence Diagnostics

Fubini, in process..., diagnostics of convergence

Fubini, in process..., here goes more on analysing results of the MwG algorithm

CODA package is one of many available software tools that can be used for a performance measurement of various MCMC convergence diagnostics.

7.3.1 Visual Inspection

7.3.2 Gelman and Rubin Diagnostic

7.3.3 Geweke Diagnostic

7.3.4 Raftery and Lewis Diagnostic

7.3.5 Heidelberg and Welch Diagnostic

8 Concluding Remarks

Fubini, in process!.. In this thesis, an estimation of a simple HMM based on the FUBINI algorithm has been developed.

Fubini..., The conclusion is Fubini... The concluding remarks

Overall, we feel that the mathematical reasoning and estimates that were generate during our learning process can be regarded as reasonably well founded.

We believe the Fubini estimates are sufficiently robust for

Furthermore, as explained Fubini

This thesis has definitely various inaccuracies and is biased in any possible way.

All in all, the FUBINI is an interesting field of study. As it seems at the present moment FUBINI

Last, but not least, all the errors remain mine and we still need to do some serious work!..

8.1 Proposal for Future Research

The reader may argue that the ARCH(1) model with MSM is a simple and model. We would fully agree with the reader on that point. Though we hope that the reader would agree with us that it is better to understand and master a simple model, that investment decision are based on sound reasoning that an investor can base her/his decisions. Then, it is not really understand a more complex model, and base various investment decisions on something not being able to do decent inferences, during a modelling process.

In the near future we would like to extend MSM to Autoregressive hidden Markov Models.

9 Acknowledgements

The writing writer is very grateful to his best friend Lars Skovlund for interesting discussions, continuous feedback and invaluable suggestions about the modern scientific perspective of Bayesianism. And, the writing writer also Emcee Sun of a Gun 50 Pence Music Production Copenhagen Company

References

Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer Science Business Media, LLC, 2006.

Tim Bollerslev. Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31:307–327, 1986.

Paolo Brandimarte. *Handbook in Monte Carlo Simulation.* John Wiley and Sons, 2014.

Jun Cai. A Markov Model of Switching Regime ARCH. *Journal of Business & Economic Statistics*, 12(3):309–316, 1994.

Giorgio Canarellar and Stephen K. Pollard. A Switching ARCH (SWARCH) model of stock market volatility: some evidence from Latin America. *International Review of Economics*, 54:445–465, 2007.

George Casella and Edward I. George. Explaining the Gibbs Sampler. *America Statistical Association*, 46(3):167–174, 1992.

Siddhartha Chib. *Handbook of Econometrics.* Elsevier Science B.V., 2001.

Siddhartha Chib and Edward Greenberg. Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4): 327–335, 1995.

J. Michael Durland and Thomas H. McCurdy. Duration–Dependent Transitions in a Markov Model of U.S. GNP Growth. *Journal of Business & Economic Statistics*, 12(3):279–288, 1994.

Robert F. Engle. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, 50(4):987–1007, 1982.

Dani Gamerman. *Markov Chain Monte Carlo*. Chapman and Hall, 1997.

Stuart Geman and Donald Geman. Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images. *The IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 6 (6):721–741, 1984.

James D. Hamilton and Raul Susmel. Autoregressive Conditional Heterosckedasticity and Changes in Regime. *Journal of Econometrics*, 64:307–333, 1994.

W. Keith Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109, 1970.

Gregory F. Lawler. *Intoduction to Stochastic Processes*. Chapman & Hall, 2010.

Peter M. Lee. *Bayesian Statistics An Introduction*. Arnold, a member of the Hodder Headline Group, third edition, 2004.

Ming Yuan Leon Li and Hsiou Wei William Lin. Examining the Volatility of Taiwan Stock Index Returns via a Three-Volatility-Regime Markov-Switching ARCH Model. *Review of Quantitative Finance and Accounting*, 21(21):123–139, 2003.

Nicholas Metropolis and Stanislaw Ulam. The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, and Augusta H. Teller. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Phisics*, 21(6):1087–1092, 1953.

John F. Monahan. *Numerical Methods of Statistics*. Cambrindge University Press, second edition, 2011.

P. H. Peskun. Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, 60(3):607–612, 1973.

P. H. Peskun. Guidelines for Choosing the Transition Matrix in Monte Carlo Methods Using Markov Chains. *Journal of Computational Physics*, 40:327–344, 1981.

Ales Prochazka, Jan Uhlir, and P.W.J. Rayner. *Signal Analysis and Prediction*. Springer Science+Business Media New York, first edition, 1998.

Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of IEEE*, 2: 257–286, 1988.

Christian Robert and George Casella. A Short History of Markov Chain Monte Carlo: Subjective Recollections from Incomplete Data. *Statistical Science*, 26(1):102–115, 2011.

John Schiller, Murray R. Spiegel, and R. Alu Srinivasan. *Probability and Statistics*. McGraw Hill, third edition, 2009.

Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications With R Examples*. Springer, third edition, 2010.

Martin A. Tanner and Wing Hung Wong. The Calculation of Posterior Distributions by Data Augmentation. *Journal of the American Statistical Association,*, 82(398):528–540, 1987.

Stan Development Team. *Stan Modeling Language:User's Guide and Reference Manual*. Stan, 2.6.2 edition, 2015.

Ruey S. Tsay. *Analysis of Financial Time Series*. John Wiley & Sons, third edition, 2010.

10 Appendix

10.1 Software Tools and Solutions

The numerical results that have been generated in this thesis, were done by a parsimonious repertoire of different software tools, numerical techniques and scientific programming languages. Nevertheless, the cautious reader must be warned that these applied numerical solutions that we provide along with the thesis, as if and with no warranty of any kind. Use at own risk!.. The writing author cannot be responsible of any financial loss of whatsoever. The writing writer support the idea of free software and open philosophy. In other words, anyone can and/or may copy following code and trade secrets, in any possible and impossible way without any written permission. If the interested reader might have any question to various possible practical issues, implementations and/or procedures, then s/he should not hesitate write to me. Fell free to contact me at cphniceguy@gmail.com

10.2 GAUSS™

The commercial statistical software GAUSS™ is a mathematical and computer system, that has a fast matrix programming language which is especially designed for computationally intensive tasks. GAUSS is perfect for most scientists that do not wish to write a whole program from the bottom. It is highly flexible or powerful enough to perform complicated statistical analysis or to work on big data problems. GAUSS provides industry solutions, on-site training, webinars and has an excellent customer service and support. For more information, please consult `http://www.aptech.com`.

We use the free license of student version GAUSS Light 15 for Windows 64. That has a nice automatic setup wizard, but has a matrix size limitation of 10,000 elements and no debugger.

10.2.1 GAUSS - Programming Source Code

Fubini, thoughts in process,.. here goes the different implementations details and issues of the computations

10.3 Python

Fubini, thoughts in process,.. we want to talk little about the background of the language

10.3.1 Python - Programming Source Code

Fubini, thoughts in process,.. here goes the different implementations details and issues of the computations

10.4 R

The R project is a free software and statistical computing environment, that can be used for statistical computing and graphics. R have rich variety of built-in functions, libraries, plug-ins, tools and various possible developer extensions. If you want to learn more about R and see various possibilities in it, then go to `https://www.r-project.org/`.

10.4.1 R - Programming Source Code

Fubini, thoughts in process,.. here goes the different implementations details and issues of the computations

10.5 Stan

Stan is an imperative higher-order probabilistic programming language that is supported by Microsoft Research. Stan is coded in C++ library and runs almost on all major platforms. It is an open-source and freedom-respecting software. Stan can be used for semi-automatic programming implementations that have further interface extensions RStan, PyStan, MATLAB and JuliaStan. We think that Stan is well-suited for Bayesian data analysis, MCMC sampling, statistical inference and optimization. To get more information about Stan and use of it (i.e., such as tutorials, user group, user guide manual, etc ...), one may consult Stan's official homepage at `http://mc-stan.org/`.

10.5.1 Stan - Programming Source Code

Fubini, thoughts in process,.. here goes the different implementations details and issues of the computations

FUBINI, fix me later!...

Data: this text
Result: how to write algorithm with LaTeX2e
initialization;
while *not at end of this document* **do**
    read current;
    if *understand* **then**
        go to next section;
        current section becomes this one;
    else
        go back to the beginning of current section;
    end
end

Algorithm 1: How to write algorithms