A Bayesian Semi-parametric Model for Realized Volatility

A BAYESIAN SEMI-PARAMETRIC MODEL FOR REALIZED VOLATILITY

BY

TIAN FENG, B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF MATHEMATICS AND STATISTICS

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

Master of Science (2013) McMaster University

(Department of Mathematics and Statistics) Hamilton, Ontario, Canada

TITLE:	A Bayesian Semi-parametric Model for Realized Volatility
AUTHOR:	Tian Feng B.Sc., (Mathematics and Statistics) McMaster University, Hamilton, Canada
SUPERVISOR:	Dr. John Maheu
NUMBER OF PAGES:	vii, 90

Contents

List of Figures

To Xiumei Tian

Abstract

Due to the advancements in computing power and the availability of high-frequency data, the analyses of the high frequency stock data and market microstructure has become more and more important in econometrics. In the high frequency data setting, volatility is a very important indicator on the movement of stock prices and measure of risk. It is a key input in pricing of assets, portfolio reallocation, and risk management. In this thesis, we use the Heterogeneous Autoregressive model of realized volatility, combined with Bayesian inference as well as Markov chain Monte Carlo methods to estimate the innovation density of the daily realized volatility. A Dirichlet process is used as the prior in a countably infinite mixture model. The semi-parametric model provides a robust alternative to the models used in the literature. I find evidence of thick tails in the density of innovations to log-realized volatility.

Acknowledgements

I will take this opportunity to thank my extraordinary supervisor Dr. John Maheu. It was a wonderful experience to work under his supervision. Also, I want to thank the supervisory committee members, Dr Shui Feng, and Dr. Aaron Childs for their support and recommendations, Professor Narayanaswamy Balakrishnan for his nice suggestions, my parents for their encouragement and my friends Hon Yiu So, Cong Zhou, Xiaojun Zhu, Tao Tan, Song Mao, Xiaolin Wang, Youzhou Zhou and all the other research students for their kind assistance.

Notation and abbreviations

ARMA . . . Autoregressive Moving Average

ARFIMA . . . Autoregressive Fractionally Integrated Moving Average

CRP . . . Chinese Restaurant Process

DP . . . Dirichlet Process

DPM . . . Dirichlet Process Mixture

HAR-RV . . . Heterogeneous Autoregressive Model on Realized Volatility

i.e. . . . that is to say

i.i.d. . . . independent, identically distributed

IV . . . Integrated Variation

OLS . . . Ordinary Least Squares

p.d.f. . . . Probability Density Function

p.m.f. . . . Probability Mass Function

QV . . . quadratic variation

RV . . . realized volatility

SDE . . . stochastic differential equation

MCMC . . . Markov chain Monte Carlo

Chapter 1

Introduction

Econometric and Mathematical Finance are heavily dependent on quantified data to explain the relationship between different variables and economic phenomena. For example, in finance, usually the historical data of stock prices is used to analyze the stock market behavior and make predictions on the future stock prices. Recently, due to the advancements in computing power and the availability of high-frequency data, research in these areas has become popular. Therefore, the ability to trade investment assets at any time of the day makes the analysis of the high frequency stock data and the micro-market structures important.

In high-frequency data, volatility (variance) of the price process is the key input in asset pricing, portfolio allocation and risk management, (Andersen *et al.*, 2007). Estimation of ex-post volatility from high-frequency data leads to improved forecasts of volatility and returns, (Maheu and McCurdy, 2011). Realized volatility or realized variance is a measure of ex-post variation of the return process. According to Andersen and Benzoni (2008), realized volatility is a nonparametric *ex-post* estimate

of return variation since very few assumptions are made on the underlying stochastic process. The basic estimate of realized volatility is defined as the sum of the intraperiod squared returns over a fixed time period. This can be an inconsistent estimator and have biases if market micro-structure dynamics are present, (Russell and Bandi, 2004), (Zhang *et al.*, 2003). Since the continuous price recording is not available, and measurable error is unavoidable, in addition, market microstructure noise including the existence of price reporting errors, bid-ask bounce, different trading markets, strategic order flows and price discreteness may cause serial correlation in the price process. The bias may offset the benefit gain with respect to high frequency price process sampling. Some adjustments of realized volatility to correct for this are suggested by Hansen and Lunde (2006).

There are several models related to volatility (variance) modeling. The most basic and widespread one is the GARCH model of (Engle, 1982) and (Bollerslev, 1986). For example, if r_t is the daily return, then the GARCH(1,1) model is

$$r_t = \mu + \epsilon_t \tag{1.1}$$

$$\epsilon_t = \sigma_t z_t,\ z_t \sim iid(0,1) \tag{1.2}$$

$$\sigma_t^2 = w + \alpha\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2. \tag{1.3}$$

This is a particular parametric model that implies the conditional variance σ_t^2 is a function of all past squared return innovations, ϵ_t^2. The exponential general autoregressive conditional heteroskedastic (EGARCH) model is another benchmark parametric model, see Maheu and McCurdy (2011).

Unlike the GARCH model using realized volatility provides "data" on daily volatility without the underlying parametric assumption. Now standard time series models can be used to model realized volatility data. In modeling time series with long memory, fractional difference operators are introduced as in the ARFIMA model of realized volatility. To capture the long-memory dependence of the high frequency data, the simple tri-variate vector autoregressive (VAR-RV) model is suggested by Andersen *et al.* (2003) and the comparison with real data shows that it outperforms most models. Furthermore, the adaption of Heterogeneous AutoRegressive (HAR) functions of lagged log RV is proposed by Corsi (2009) and Andersen *et al.* (2007). The HAR model on RV (HAR-RV) have a noticeable improvement when compare to the VAR model, and at the same time, the HAR model has less parameters, straightforward to understand and simple to implement.

The purpose of this thesis is to estimate a Heterogeneous model of realized volatility with an unknown innovation distribution. A Dirichlet process is used as the prior of a countably infinite mixture model. The semi-parametric model provides a robust alternative to the models used in the literature. We find evidence of thick tails in the innovation to the log-realized volatility. The predictive density is leptokurtic and has a smaller slope in the tail than the normal distribution. This is probability due to the present of jumps in realized volatility.

First, we start with discussing the theoretical framework and models of realized volatility in Chapter 2. Then, we go on to describe Dirichlet Process and various modeling methods in Chapter 3. In Chapter 4, an example with real data will be given to illustrate the model. Finally, we will conclude our study and give some ideas for further study.

Chapter 2

Modeling and Forecasting of Realized Volatility

2.1 Introduction

In the Econometric and Mathematical Finance area, volatility is a very important indicator on the movement of the stock price. A good estimator of volatility becomes more and more attractive in pricing of assets, reallocating of assets, and managing risk in the derivative market and many other financial areas. A large amount of research has been done to find a good volatility estimate since volatility is not a directly observable variable. As the development of data management technology and more frequent trading in the financial market, estimation of volatility from high-frequency data draws much attention by scholars during recent decades. Analysis of the high-frequency data introduces some new challenges since it has some unique characteristics.

In the financial market, some of the historical observations were not expected

to happen as the way they have happened. It is probably due to effect of non-synchronous trading, bid-ask spread, gradual response of prices to a block trade, regional disparity, price discreteness and rounding. Thus the prediction based on these historical observations may not truly represent the operation of market, we refer this as microstructure noise in practice. And adjustment is introduced to eliminate the bias. In this chapter, a theoretical framework of the realized volatility and some useful models of realized volatility are introduced.

2.2 Theoretical Framework

Assume we are in a frictionless market and without microstructure noise. For a given time interval $[0, T]$, and $0 \leq t \leq T$. First, let us consider the standard continuous time process without jump components, (Back, 1991),

$$\frac{dS(t)}{S(t)} = \mu(t)dt + \sigma(t)dW(t), \quad 0 \leq t \leq T, \tag{2.1}$$

where

$S(t)$ is the level of the instantaneous price.

$\mu(t)$ is the cadlág finite variation, can treat as the instantaneous conditional mean of return.

$W(t)$ is a standard Brownian motion.

$\sigma(t)$ is a stochastic process, and it is independent of $W(t)$, can treat as the instantaneous volatility of return.

To eliminate arbitrage opportunities, the log of the stock price should follow the

semi-martingale, (Protter, 1990). In a standard arithmetic Brownian motion followed by the stock price for one day $T = 1$, we have

$$s(t) = logS(t). \tag{2.2}$$

Applying Ito's Lemma,

$$ds(t) = (\alpha - \delta - 0.5\sigma^2)dt + \sigma dZ(t), \quad 0 \leq t \leq 1. \tag{2.3}$$

where

$Z(t)$ as a standard Brownian motion.

α as the continuously compounded rate of return for the stock.

δ as the continuously compounded dividend yield.

σ as the spot volatility.

Denote the intraday return observations for day t as

$$r_{t,i} = s(t + \frac{i}{n}) - s(t + \frac{i-1}{n}), \quad t = 1, 2, \ldots, n. \tag{2.4}$$

where $s(t)$ is the logarithm of the instantaneous price. Here the price process is sampled at equal intervals but the results hold for non-equal intervals as well. The realized volatility is the sum of squared returns over a specific time interval.

$$RV(t) = \sum_{j=1}^{n} r_{t,j}^2. \tag{2.5}$$

The quadratic variation (also known as Integrated Variation) can be found in the

following way,

$$QV(t) = \int_{t-j}^{t} \sigma^2(s)ds. \tag{2.6}$$

where $\sigma(s)$ is a stochastic process of square root of volatility. As the sampling frequency increases, $n \to \infty$, realized volatility RV_t converges to quadratic variation (Andersen *et al.*, 2007),

$$RV(t) \xrightarrow{n\to\infty} QV(t). \tag{2.7}$$

Note that this result holds without saying anything about the $\sigma^2(t)$ process. The volatility process can contain long-memory, jumps or breaks.

Jumps, Bipower Variation, High frequency data

Jumps allow for discontinuities in the price process and capture abrupt changes in stock prices. For example, important microeconomic news or firm specific news events can result in large instantaneous changes in a stock price, (Andersen *et al.*, 2007). The previous section assumed no jumps. Adding a jump component to the stochastic differential equation (SED), the model, and QV becomes

$$ds(t) = \mu(t)dt + \sigma(t)dW(t) + J(t)dq(t), \tag{2.8}$$

$$QV(t) = \int_{t-k}^{t} \sigma^2(s)ds + \sum_{t-k\le s\le t} J^2(s), \tag{2.9}$$

where $J(t)$ is the size of discrete jumps in the logarithmic price process, and $q(t)$ is a counting process with possibly time-varying intensity. The realized bi-power variation

is introduced by (Barndorff-Nielsen and Shephard, 2004) have the following form,

$$BV(t,n) = \frac{\pi}{2}\sum_{i=2}^{n}|r_{t,i}||r_{t,i-1}|. \tag{2.10}$$

As the sampling frequency n increases, we have the following approximation, (Andersen and Benzoni, 2008),

$$BV(t,k) \overset{n\to\infty}{\longrightarrow} \int_{t-k}^{t}\sigma^2(s)ds. \tag{2.11}$$

$$RV(t,k,n) \overset{n\to\infty}{\longrightarrow} QV(t,k). \tag{2.12}$$

$$RV(t,k,n) - BV(t,k) \overset{n\to\infty}{\longrightarrow} \sum_{t-k\le s\le t} J^2(s). \tag{2.13}$$

Therefore, it is possible to estimate the jump component of QV.

2.3 Estimation of Realized Volatility in Practice

Intraday return is usually denote as $r_{t,i}$, where t is the index of the trading day, and $i = (1, 2, ..., n)$ is the number of intraday returns in that day t. The 5 minutes grids normally has $N = 78$.

The usual estimator of RV_t is the sum of squared returns over a specific time interval denoted as,

$$RV_{t,u} = \sum_{i=1}^{n} r_{t,i}^2. \tag{2.14}$$

Assuming no market microstructure, realized volatility is an unbiased estimator and the property of this estimator are discussed by (Barndorff-Nielsen and Shephard, 2002).

However, realized volatility can be biased and inconsistent under certain situations, for example, from market microstructure noise and discretization error. In this case, we observe the price process with error. For example, the existence of price reporting errors, bid-ask bounce, different trading markets and price discreteness may cause serial correlation in the price process. Especially in high frequency price process analysis, the bias and inconsistent may offset the benefit gain with respect to more frequent sampling. Hansen and Lunde (2006) recommend the following adjusted estimators to account for bias,

$$RV_{t,q} = w_0 \sum_{i=1}^{n} r_{t,i}^2 + 2 \sum_{j=1}^{q} w_j \sum_{i=1}^{n-j} r_{t,i} r_{t,i+j}, \quad j = 1, 2, \ldots, q; \; q = 1, 2, 3. \tag{2.15}$$

where the Bartlett weights is

$$w_j = 1 - \frac{j}{q+1}, \quad j = 1, 2, .., q. \tag{2.16}$$

The Bartlett weights ensure the estimate is always positive. The properties of this modified estimator have been investigated by Ole E. Barndorff-Nielsen (2008).

2.4 Models for RV

In the section, several models including Autoregressive-moving-average (ARMA) model, Autoregressive fractionally integrated moving average (ARFIMA) model, and Heterogeneous Autoregressive model of realized volatility (HAR-RV) are stated.

2.4.1 ARMA model

In the time series analysis, Autoregressive-moving-average (ARMA) model is commonly used in the volatility modeling. It contains less number of parameters compared to Autoregressive (AR) or Moving-average (MA) model. And it gives a better description on dynamic structure of the data. The general ARMA model was introduced by Peter Whittle (1951). For a time series data X_t, ARMA model with parameters p, q can be described as ARMA(p, q)

$$X_t = c + \epsilon_t + \sum_{i=1}^{p} \psi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}. \tag{2.17}$$

where s is a constant, ψ_i, θ_i are parameters of the model, ϵ_i is the white noise error terms, p is the order of AR part and q is the order of MA part. An alternative definition of ARMA(p, q) with the lag operator L is

$$(1 + \sum_{i=1}^{p} \phi_i L^i) X_t = (1 + \sum_{i=1}^{q} \theta_i L^i) \epsilon_t. \tag{2.18}$$

The simplest case is ARMA$(1, 1)$,

$$X_t - \psi_1 X_{t-1} = c + \epsilon_t + \theta_1 \epsilon_{t-1}. \tag{2.19}$$

with AR component on the left hand side, and MA component on the right hand side. In empirical work, $X_t \equiv RV_t$ or $X_t \equiv log(RV_t)$. It is found that the ARMA models do not capture all the persistence in the data. This has lead to alternative models.

2.4.2 ARFIMA model

Autoregressive fractionally integrated moving average (ARFIMA) model is a generalization of Autoregressive integrated moving average (ARIMA) model, and ARIMA model is a generalization of ARMA model in time series analysis. For a time series data $X_t \equiv log(RV_t)$, ARFIMA model with parameters p, d, q can be described as ARFIMA(p,d,q)

$$(1 - \sum_{i=1}^{p} \phi_i B^i)(1 - B)^d X_t = (1 + \sum_{i=1}^{q} \theta_i B^i)\epsilon_t. \tag{2.20}$$

where ψ_i, θ_i are parameters of the model, ϵ_i is the white noise error terms, p is the order of AR part, q is the order of MA part, d is the order of integrated part, and B is the backshift operator. This model implies an ACF that has a hyperbolic rate of decay in contrast to the ARMA which has an exponential decay. However, these models are complicated to estimate due to the likelihood function.

2.4.3 HAR-RV model

Denote daily realized volatility for day t as RV_t^d, d means daily. The Heterogeneous Autoregressive model of realized volatility (HAR-RV)is given by (Andersen *et al.*, 2007)

$$RV_{t+1}^d = \mu + \beta^d RV_t^d + \beta^w RV_t^w + \beta^m RV_t^m + \varepsilon_{t+1d}. \tag{2.21}$$

where ε_{t+1d} is the white noise error terms, and

$$RV_t^w = \frac{1}{5}(RV_t^d + RV_{t-1}^d + RV_{t-2}^d + RV_{t-3}^d + RV_{t-4}^d). \tag{2.22}$$

$$RV_t^m = \frac{1}{22}(RV_t^d + RV_{t-1}^d + RV_{t-2}^d + \ldots + RV_{t-22}^d). \tag{2.23}$$

A s-step past averaging of the RV can be used to find the weekly mean and monthly mean, RV_t^w is the weekly mean of realized volatility at day t, w stands for week. RV_t^m is the monthly mean of realized volatility at day t, m stands for monthly. This model is also valid when we replace RV as log(RV). This model implies a restricted AR(22), but is very parsimonious with only four regression parameters. Andersen *et al.* (2007) and others show this model to work well and closely approximate forecasting results from ARFIMA specifications.

Our study will focus on this HAR-RV model with a more general innovation distributions.

Chapter 3

Dirichlet Process

3.1 Introduction

In probability theory, the Dirichlet process (DP) is a random process, which is widely used in Bayesian non-parametrics. In the mixture model analysis, how to determine the number of mixtures is a fundamental problem. A pre-specified number of clusters may cause minor bias. The traditional method including model selection based on different number of clusters may over-fit or under-fit of the data. While a Bayesian non-parametric model side-steps this part, it allows the data set to determine its own clusters without actually doing the model selection. Suppose the observation follows an unknown distribution which we try to investigate, the Bayesian method defines a known prior distribution for the underlying unknown distribution. And the prior distribution in the non-parametric approach is in the space of distributions. The Bayesian method is computationally expensive, but this problem has been successfully solved with the development of Makov chain Monte Carlo (MCMC) technology. In this chapter, the Dirichlet distribution and five general interpretations of Dirichlet

process are introduced.

3.2 Dirichlet distribution

The Dirichlet Distribution always denote as $Dir(\alpha)$, where α is called "concentration parameter" or "scaling parameter" with positive scalars. In Bayesian statistics, Dirichlet Distributions serve as a conjugate prior for multinomial distributions and categorical distributions.

Let $X = (X_1, X_2, \cdots, X_k)$ be a vector with $X_j > 0, j = 1, 2, \ldots, k$ and $\sum_{j=1}^{k} X_j = 1$ distributed as a Dirichlet distribution of order $k \geq 2$ with parameter $\alpha = (\alpha_1, \cdots, \alpha_k)$ denote as $X \sim Dir(\alpha)$. This is a distribution over multinomials, and the p.d.f. is given by

$$f(x|\alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{k} x_i^{\alpha_i - 1}, 0 < x_i < 1, \forall i, \sum_{i=1}^{k} x_i = 1. \tag{3.1}$$

Define $\alpha_0 = \sum_{i=1}^{k} \alpha_i$, the normalizing constant is the multinomial Beta function

$$B(\alpha) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma(\alpha_0)} \tag{3.2}$$

and $\Gamma(z)$ is the gamma function with an improper integral

$$\begin{aligned} \Gamma(z) &= \int_0^{\infty} t^{z-1} e^{-t} dt, \quad z > 0 && (3.3) \\ &= (z-1)! \text{ for a positive integer } z && (3.4) \end{aligned}$$

3.2.1 Mean and Variance

$$\begin{aligned} E(x_i) &= \int_0^1 \frac{\Gamma(\alpha_0)}{\prod_{j=1}^k \Gamma(\alpha_j)} \prod_{j=1}^k x_j^{\alpha_i - 1} x_i dx_i \\ &= \frac{\Gamma(\alpha_0)}{\prod_{j=1}^k \Gamma(\alpha_j)} \int_0^1 x_1^{\alpha_1 - 1} \cdots x_i^{\alpha_i} \cdots x_k^{\alpha_{k-1}} dx_i \\ &= \frac{\Gamma(\alpha_0)}{\prod_{j=1}^k \Gamma(\alpha_j)} \frac{\Gamma(\alpha_1) \cdots \Gamma(\alpha_i + 1) \cdots \Gamma(\alpha_k)}{\Gamma(\alpha_0 + 1)} \\ &= \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_i)} \frac{\Gamma(\alpha_i + 1)}{\Gamma(\alpha_0 + 1)} = \frac{\alpha_i}{\alpha_0}. \end{aligned} \tag{3.5}$$

$$\begin{aligned} E(x_i^2) &= \int_0^1 \frac{\Gamma(\alpha_0)}{\prod_{j=1}^k \Gamma(\alpha_j)} \prod_{j=1}^k x_j^{\alpha_i - 1} x_i^2 dx_i \\ &= \frac{\Gamma(\alpha_0)}{\prod_{j=1}^k \Gamma(\alpha_j)} \int_0^1 x_1^{\alpha_1 - 1} \cdots x_i^{\alpha_i + 1} \cdots x_k^{\alpha_{k-1}} dx_i \\ &= \frac{\Gamma(\alpha_0)}{\prod_{j=1}^k \Gamma(\alpha_j)} \frac{\Gamma(\alpha_1) \cdots \Gamma(\alpha_i + 2) \cdots \Gamma(\alpha_k)}{\Gamma(\alpha_0 + 2)} \\ &= \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_i)} \frac{\Gamma(\alpha_i + 2)}{\Gamma(\alpha_0 + 2)} = \frac{(\alpha_i + 1)\alpha_i}{(\alpha_0 + 1)\alpha_0}. \end{aligned} \tag{3.6}$$

$$V(x_i) = E(x_i^2) - (E(x_i))^2 = \frac{(\alpha_i + 1)\alpha_i}{(\alpha_0 + 1)\alpha_0} - (\frac{\alpha_i}{\alpha_0})^2 = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)}. \tag{3.7}$$

Figure 3.1 is sample p.m.f. drawn from the Dirichlet distribution with different precision parameters $\alpha = (\alpha_1, \alpha_2, \alpha_3)$ in R^3. From left to right, we choose $\alpha = (100, 100, 100)$, $\alpha = (1, 1, 1)$, $\alpha = (0.05, 0.05, 0.05)$ (upper row), and $\alpha = (0.05, 1, 100)$, $\alpha = (1, 100, 100)$, $\alpha = (1, 1, 0.05)$ (lower row) respectively. If the precision parameter is vector with value one, mass point tend to be uniformly distributed. If precision parameter is a constant vector, mass point is symmetric, otherwise mass point is not

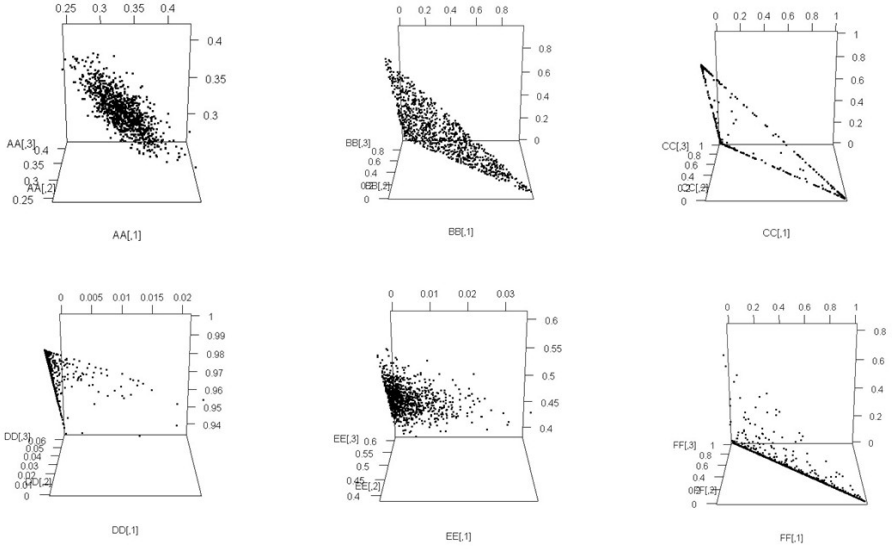

Figure 3.1: Plots of sample pmfs (Dirichlet distribution).

symmetric.

3.2.2 Generalization of the Beta

Notice that the Dirichlet distribution is a multidimensional generalization of the Beta distribution. In fact, the Beta distribution is a special case of a Dirichlet distribution with dimension two, i.e. $k = 2$.

Define $\hat{\alpha} = (\alpha_1, \alpha_2) = (\alpha, \beta)$, the p.d.f. of $Y \sim Beta(\alpha, \beta)$ is

$$f(y; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} y^{\alpha-1}(1-y)^{\beta-1}, \quad 0 \leq y \leq 1. \tag{3.8}$$

That is to say,

$$X = (Y, 1-Y) \sim Dir(\hat{\alpha}), \quad Y \sim Beta(\alpha, \beta). \tag{3.9}$$

3.2.3 Posterior of Multinomial distribution

Suppose $Y = \{Y_1, Y_2, \cdots, Y_k\}$ follows Multinomial distribution with probability $X = \{X_1, X_2, \cdots, X_k\}$ for n independent events, and X follows Dirichlet distribution with parameter $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_k\}$, and the p.m.f. of $Y|X$ is,

$$f(y_1, y_2, \cdots, y_k | x_1, x_2, \cdots, x_k) = \frac{n!}{y_1! y_2! \cdots y_k!} \prod_{i=1}^{k} x_i^{y_i} \tag{3.10}$$

Applying Bayes' rule, the posterior is

$$\begin{aligned} f(x|y) &\propto f(y|x) f(x) \\ &\propto \left(\frac{n!}{y_1! y_2! \cdots y_k!} \prod_{i=1}^{k} x_i^{y_i}\right)\left(\frac{\Gamma(\alpha_0)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} x_i^{\alpha_i - 1}\right) \\ &\propto \prod_{i=1}^{k} x_i^{\alpha_i + y_i - 1} \\ &= \frac{\Gamma(\alpha_0 + \sum_{i=1}^{k} y_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i + y_i)} \prod_{i=1}^{k} x_i^{\alpha_i + y_i - 1} \text{ (with normalizing constant)} \\ &\sim Dir(\alpha + y) \end{aligned} \tag{3.11}$$

Hence, a multinomial prior still gives a Dirichlet posterior distribution. Dirichlet distributions are the conjugate prior for multinomial distributions.

3.2.4 Aggregation Property

An essential property for Dirichlet distribution is the"aggregation property". For example, if we have a partition $\{P_1, ..., P_q\}$ over $\{1, ..., k\}$, and $X = \{X_1, ..., X_n\}$

follows Dirichlet distribution with parameter $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_k\}$, then

$$(\sum_{i \in P_1} X_i, ..., \sum_{i \in P_q} X_i) \sim Dir(\sum_{i \in P_1} \alpha_i, ..., \sum_{i \in P_q} \alpha_i). \tag{3.12}$$

3.3 Dirichlet process and Basic Properties

A Dirichlet process (DP) is a stochastic process that is a distribution over distributions. It is the finite dimensional generalization of Dirichlet distribution based on the concept of measures. For example, suppose we ask students about their favorite subject, and limit the answer to History, Math, and English. Student may have different choice within the limitation. Therefore, we could model the probability of a particular choice as a p.m.f., hence model the student as a p.m.f. over the three subjects, which is a draw from a Dirichlet distribution over the set of three subjects. If we allow students provide any answers beyond the limit, we have to model a distribution over distribution over an infinite sample space, and this can be reached by Dirichlet process.

In a parameter space Θ, suppose there exists a base distribution or base measure G_0, and a positive scaling parameter α, if G is drawn from a DP denoted as $G \sim DP(\alpha, G_0)$, then G is a random probability measure that has the same support as G_0. Suppose for any measureable partition $\{A_1, A_2, \cdots, A_k\}$ of Θ, we have

$$(G(A_1), G(A_2), \cdots, G(A_k)) \sim Dir(\alpha G_0(A_1), \alpha G_0(A_2), \cdots, \alpha G_0(A_k)). \tag{3.13}$$

Then, we say G follows a Dirichlet process,

$$G \sim DP(\alpha, G_0). \tag{3.14}$$

This definition was first introduced by (Ferguson, 1973). Now, lets have a look at some basic properties of the DP.

3.3.1 Mean and Variance

The parameter α and G_0 play important roles in the DP, which can be illustrated by looking at the mean, variance, and p.m.f of various value of α.

Following the mean (3.5) and variance (3.7) derived from Dirichlet distribution and equation (3.13), we may get the mean distribution,

$$E[G(A_k)] = \frac{\alpha G_0(A_k)}{\sum\limits_{k'} \alpha G_0(A_{k'})} = \frac{\alpha G_0(A_k)}{\alpha} = G_0(A_k). \tag{3.15}$$

and the variance distribution,

$$\begin{aligned} V[G(A_k)] &= \frac{\alpha G_0(A_k)(\sum\limits_{k'} \alpha G_0(A_{k'}) - \alpha G_0(A_k))}{(\sum\limits_{k'} \alpha G_0(A_{k'}))^2(\sum\limits_{k'} \alpha G_0(A_{k'}) + 1)} \\ &= \frac{\alpha^2 G_0(A_k)(1 - G_0(A_k))}{\alpha^2(\alpha + 1)}, \text{since} \quad \sum_{k'} G_0(A_{k'}) = 1 \\ &= \frac{G_0(A_k)(1 - G_0(A_k))}{1 + \alpha}. \end{aligned} \tag{3.16}$$

From above analysis, we may say the base distribution G_0 is the mean measure of DP. Furthermore, as the precision parameter α goes to infinity, G approaches the

base distribution G_0, i.e., the mass in the Dirichlet process will be more concentrate around the mean value for larger α. In addition, the variance will decrease as the increase value of α, this is the reason for some scholars referred α as the inverse variance parameter. Note that $G \mapsto G_0$ pointwise. For instance, G will be a discrete distribution with probability one even if G_0 is a continuous distribution.

3.3.2 Posterior Dirichlet Process of Multinomial distribution

Consider $G \sim DP(\alpha, G_0)$, and for partition $A = \{A_1, A_2, \cdots, A_k\}$,

$$(G(A_1), G(A_2), \cdots, G(A_k)) \sim Dir(\alpha G_0(A_1), \alpha G_0(A_2), \cdots, \alpha G_0(A_k)). \tag{3.17}$$

Suppose $\phi = \{\phi_1, \phi_2, \cdots, \phi_n\}$ are a sequence of independent draws from G. Then for each ϕ_j, $j = 1, 2, \cdots, n$, we have

$$P(\phi_j \subset A_i | G) = G(A_i), \quad j = 1, 2, \cdots, n \tag{3.18}$$

That is to say,

$$\phi | G \sim G. \tag{3.19}$$

Suppose we define a mass point at A_i as,

$$\delta_{\phi_j}(A_i) = \begin{cases} 1 & \text{if } \phi_j \in A_i;\ i = 1, 2, \cdots, k;\ j = 1, 2, \cdots, n \\ 0 & \text{otherwise} \end{cases} \tag{3.20}$$

And define

$$n_i = \sum_{j=1}^{n} \delta_{\phi_j}(A_i),\ i = 1, 2, \cdots, k,\ j = 1, 2, \cdots, n \tag{3.21}$$

The posterior distribution is slightly different from the original Dirichlet distribution by adding mass weights for each component, which can be written as

$$G(A_1), \cdots, G(A_k)|\phi_1, \cdots, \phi_n \ \sim \ Dir(\alpha G_0(A_1) + n_1, \cdots, \alpha G_0(A_n) + n_k). \quad (3.22)$$

Then the posterior process is still a DP with a new concentration parameter and base distribution

$$G|\phi \sim DP(\alpha', G_0') \quad (3.23)$$

where

$$\alpha' = \alpha + n \quad (3.24)$$

$$G_0' = \frac{\alpha}{\alpha + n} G_0 + \frac{\sum_{j=1}^{n} \delta_{\phi_j}}{\alpha + n} \quad (3.25)$$

The base distribution can be recognized as a weighted average between G_0 and the empirical distribution of n events $\frac{\sum_{j=1}^{n} \delta_{\phi_j}}{n}$. As α goes to infinity, $G_0^{'}$ approaches to G_0. As α goes zero or n goes to infinity ($n \gg \alpha$), $G_0^{'}$ approaches the empirical distribution, and the Dirichlet process approaches the true underlying distribution.

3.3.3 Predictive Distribution

Suppose $X = \{X_1, \ldots, X_n\}$ are i.i.d. draws from G, and $G \sim DP(\alpha, G_0)$.

For any $i = 1, \ldots, n$, define $X_{-i} = \{X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n\}$.

The conditional distribution for $X_i|X_{-i}$ is

$$X_i|X_{-i} \sim \frac{\alpha}{n - 1 + \alpha} G_0(X_i) + \frac{\sum_{j \neq i} \delta_{X_j}(X_i)}{n - 1 + \alpha}, \quad (3.26)$$

i.e. X_i is a new draw from the base distribution G_0 with probability $\frac{\alpha}{n-1+\alpha}$, and X_i is remain the same as before with probability $\frac{n}{n-1+\alpha}$. Therefore, there is positive probability that the first n cases can be classified to $k \leq n$ clusters, denote X_j^*, $j = 1, \ldots, k$. And the conditional distribution for $X_{n+1}|X$ is

$$X_{n+1}|X \sim \frac{\alpha}{n+\alpha} G_0(X_{n+1}) + \frac{\sum_{j=1}^{n} \delta_{X_j}(X_{n+1})}{n+\alpha} \tag{3.27}$$

$$\sim \frac{\alpha}{n+\alpha} G_0(X_{n+1}) + \frac{\sum_{j=1}^{n} \delta_{X_j^*}(X_{n+1})}{n+\alpha} \tag{3.28}$$

This means that a new draw from a DP based on the previous draws can be a new draw from the base measure G_0 with probability $\frac{\alpha}{n+\alpha}$, or be the same draw as X_j with probability $\frac{n_j}{n+\alpha}$.

3.3.4 Exchangeability

Suppose there is a sequence of draws $(X_{i_1}, X_{i_2}, \cdots, X_{i_n})$ from a DP, then for any permutation $(X_{j_1}, X_{j_2}, \cdots, X_{j_n})$ we have

$$P(X_{i_1}, X_{i_2}, \cdots, X_{i_n}) = P(X_{j_1}, X_{j_2}, \cdots, X_{j_n}) \tag{3.29}$$

This "exchangeability" property for draws is a fundamental property for DP, which can be proven by using De Finetti (1931). It is a weaker assumption than "independent, identically, distributed" assumption. Every i.i.d. sequence is exchangeable, but every exchangeable sequence need not be i.i.d..

3.4 Construction

In order to construct a DP, or draw some samples from this given DP, there are three further interpretations of this random process: Chinese Restaurant Process (CRP), Polya Urn scheme (also known as Blackwell-MacQueen urn scheme), and Stick-Breaking construction (referred to GEM process).

3.5 Chinese Restaurant Process (CRP)

Suppose there are infinite tables in a Chinese restaurant, and infinitely many customers can sit at one table, and a table only offers one type of dish. Suppose customers are willing to have a meal in this restaurant. For the first customer, denoted as X_1, he can seat in any table in this restaurant and choose his meal, mark this table as the first table. His meal is drawn from G_0. When the second customer comes, there is one table already with a customer. The second customer, denoted as X_2, can choose to seat in either the first table and have the same dish as customer one or a new table depending on whether he likes the dish or not in the first table. Similarly, when the $n+1$-th customer, denoted as X_{n+1}, comes, there are k tables already have customers, say, there are $n_1, n_2, \cdots, n_k$ customers respectively with k different types of dishes. The $n+1$-th customer can have a look at all the dishes on the table, and see which one he likes, for example, the i-th table. If none of the dishes satisfied his taste, he can find a new table and order a new dish, which again his dish is drawn form G_0.

We assume the customer is a normal people, and the probability of choosing a

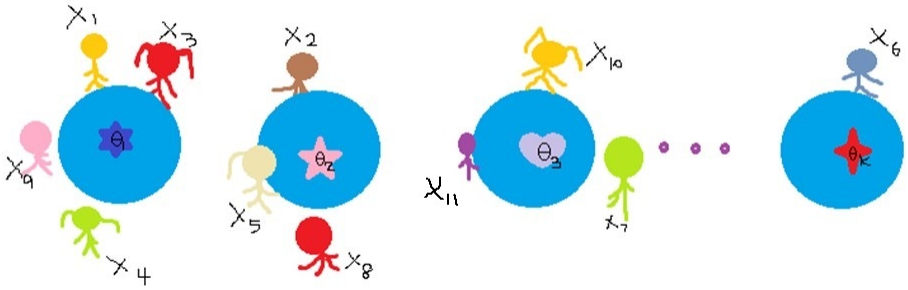

Figure 3.2: Chinese Restaurant Process

particular table from the first k table should be proportional to the number of customers in that table, i.e. the more customers on the table, the higher the probability the next customer would like to seat on this table. This is so called "rich-gets-richer" phenomenon or "self-reinforcing" property of the DP. In mathematical notation, we say he can choose to seat in the ith table with probability $\frac{n_i}{\alpha+n}$ to share the dish in the ith table if he likes, or a new table G_0 with probability $\frac{\alpha}{\alpha+n}$, where $1 \leq i \leq k$ to choose his own meal, i.e.

$$X_{n+1}|X_1, \cdots, X_n = \begin{cases} \sim G_0 & \text{probability} \frac{\alpha}{\alpha+n} \\ X_k^* & \text{probability} \frac{n_k}{\alpha+n} \end{cases} \tag{3.30}$$

For the first n customers in this case, we have k tables with different number of customers, in other words, we classify these n customers into k clusters based on their favorite dishes, which is a partition of the space $\Theta = \{1, 2, ..., n\}$. This is usually known as the "Cluster Effect" of DP. This Chinese restaurant construction of DP does not only define a distribution of partitions Θ, but also defines a distribution over the permutation of Θ as a result of exchangeability since Chinese restaurant usually has round tables .

3.6 Polya Urn Model

Polya Urn Model is the same as Chinese Restaurant Process. Suppose we have an urn but without any ball in it. Assume G_0 is a distribution over colors. Now we pick a color with probability proportional to α from a distribution G_0, record the color as X_1, and put the colored ball in the urn.

$$X_1 \sim G_0, \tag{3.31}$$

Next, we either select a ball from the urn and put two same colored ball into the urn or we pick a new color from the distribution G_0, color a ball and put the ball in the urn, record the color as X_2.

$$X_2|X_1 \sim \frac{\alpha}{\alpha+1}G_0 + \frac{\delta_{X_1}}{\alpha+1}, \tag{3.32}$$

Repeat this process, we can see that the probability of selecting a ball with a particular color is $\frac{n_i}{\alpha+n}$, where n_i is the number of the ball with this color, n is the total number of ball back in the urn. The probability of putting a new ball in is $\frac{\alpha}{\alpha+n}$.

$$X_{n+1}|X_1,\ldots,X_n \sim \frac{\alpha}{\alpha+n}G_0 + \frac{\delta_{X_1}}{\alpha+n} + \ldots + \frac{\delta_{X_n}}{\alpha+n}, \tag{3.33}$$

Table in the CRP is corresponding with the color in the Polya urn processes, choosing a new table is similar with putting a new ball into the urn.

3.7 Stick-Breaking Construction

The stick breaking model is constructed by Sethuraman (1991), and can be understood in the following way. Suppose we have a stick with length one, we may pick a part X_1 with length π_1, for the rest of the stick, we may get another part X_2 with length π_2, repeat the same process, we may get a part X_n with length π_n, and sum for all the π_i should be the length of the stick.

The constructive definition of the DP is

$$\begin{aligned} p_1 &\sim Beta(1,\alpha) && (3.34)\\ p_2 &\sim Beta(1,\alpha) && (3.35)\\ &\vdots && (3.36)\\ p_i &\sim Beta(1,\alpha), && (3.37) \end{aligned}$$

and weight parameter for each breaking part can be constructed as

$$\begin{aligned} \pi_1 &= p_1 && (3.38)\\ \pi_2 &= p_2(1-p_1) && (3.39)\\ &\vdots && (3.40)\\ \pi_i &= p_i \prod_{1\le i'<i} (1-p_{i'}), \quad i>1 && (3.41) \end{aligned}$$

The part $X_1, X_2, \ldots, X_n, \ldots$ can be clustered in to $X_1^*, X_2^*, \ldots, X_i^*, \ldots$ by DP as

$$X_i^* \overset{i.i.d.}{\sim} G_0 \tag{3.42}$$

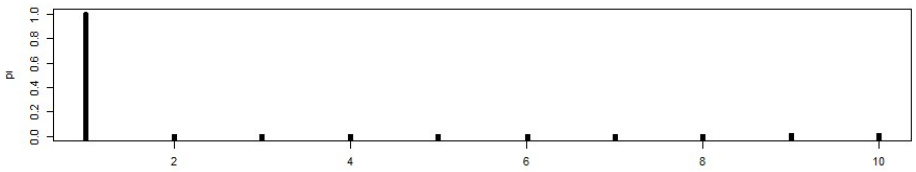

Figure 3.3: breaking a stick of one into ten pieces with $\alpha = 0.1$.

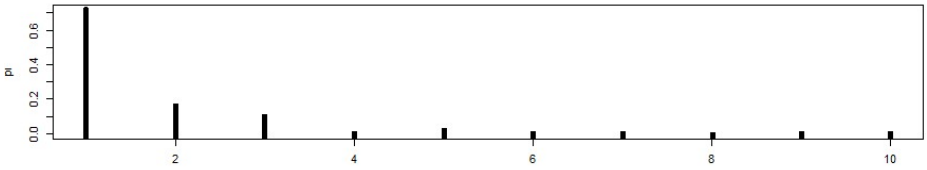

Figure 3.4: breaking a stick of one into ten pieces with $\alpha = 1$.

$$G = \sum_{i=1}^{\infty} \pi_i(p)\delta_{X_i^*} \tag{3.43}$$

Then $G \sim \mathrm{DP}(\alpha, \mathrm{G}_0)$.

To show that $\sum_{j=1}^{\infty} \pi_j = 1$, we have

$$\begin{aligned} 1 - \sum_{j=1}^{\infty} \pi_j &= 1 - \pi_1 - \pi_2 - \cdots - \pi_i - \cdots \\ &= 1 - p_1 - p_2(1-p_1) - \cdots - p_i \prod_{1 \le i' < i} (1 - p_{i'}) - \cdots \\ &= \prod_{j=1}^{\infty} (1 - p_i) \\ &\to 0 \end{aligned} \tag{3.44}$$

see (Ishwaran and James, 2001) for detail. The construction of π can be written as $\pi \sim GEM(\alpha)$ with the letter stand for Griffiths, Engen and McCloskey.

Figure 3.3, 3.4, and 3.5 are the visualization of the DP as stick breaking model. We simulate one stick for each value of α and get the value of each component. The

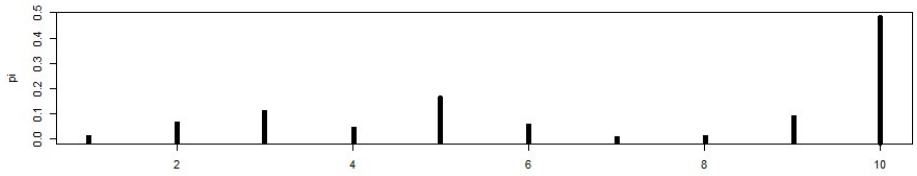

Figure 3.5: breaking a stick of one into ten pieces with $\alpha = 10$.

weights become more spread out as α gets larger.

3.8 Dirichlet Process Mixture (DPM) model

Dirichlet Process may cluster data by using a mixture model. For a set of observations $\{y_1, y_2, \cdots, y_n\}$, suppose they are independent, and conditionally distributed, given $\phi = \{\phi_1, \ldots, \phi_n\}$, where ϕ_i come from some prior distributions for $i = 1, \ldots, n$

$$y_i|\phi_i \sim f_i(y_i|\phi_i), \quad i = 1, 2, ..., n \tag{3.45}$$

$$\phi_i|G \sim G \tag{3.46}$$

If G is uncertain and modeled as a DP,

$$G|\alpha, G_0 \sim DP(\alpha, G_0). \tag{3.47}$$

We say the data come from a DPM model.

The distribution of the future observation y_{n+1} is a mixture distribution.

Denote the distinct values of the elements of ϕ as $\{\theta_1, \ldots, \theta_k\}$ with $k \leq n$, and

$n_i = \{\text{mumber of } j|\ \theta_j = \phi_i\}$

$$n_1 + \ldots + n_k = n. \tag{3.48}$$

The goal of Baysian non-parametrics is the density function. Conditional on ϕ, the predictive density for y_{n+1} is

$$P(y_{n+1}|\phi) = \frac{\alpha}{\alpha+n}\int f(y_{n+1}|\theta_{k+1})g_0(\theta_{k+1})d\theta_{k+1} + \sum_{i=1}^{k}\frac{n_i}{\alpha+n}f(y_{n+1}|\theta_i) \tag{3.49}$$

This is a weighted average of base distribution and the empirical distribution. As α goes to infinity, y_{n+1} would more likely to be a new draw from G_0. As α goes to zero or n goes to infinity ($n \gg \alpha$), y_{n+1} would more likely to be a same point from previous draw, and the Dirichlet process approaches the true underlying distribution. In particular, f_i can be a normal density with mean μ_i and variance σ_i^2 for $i = 1, \ldots, n$. If $\mu|\sigma^2$ follows a normal distribution and σ^2 follows a inverse-gamma distribution, then y_{n+1} becomes a mixture of normal and student-t distributions. We say the data come from a Dirichlet Process normal mixture model (Escobar and West, 1994),(Mike West and Escobar, 1994),(Escobar, 1988),(Ferguson, 1983),(Mike West and Escobar, 1994).

3.9 Markov chain Monte Carlo method

MCMC algorithm can be used based on CRP, stick-breaking process, and DPM model. The most common methods in MCMC are Gibbs sampling and Metropolis-Hastings algorithm.

In the computation on basic integration, suppose the given p.d.f. of θ is f, and

$\{\theta_1, \cdots, \theta_N\}$ are sampled from distribution f, we can approximate the expectation by using law of large numbers

$$\frac{1}{N}\sum_{i=1}^{N} h(\theta_i) \xrightarrow{p} \int f(\theta)h(\theta)dx = E_f(h(\theta)) \tag{3.50}$$

The following section is going to introduce the Gibbs sampling with conjugate priors. For an unknown p.d.f. of θ, we may generate a Markov chain $\theta_1, \ldots$ with stationary distribution f. In the DPM model, suppose we have observations $Y_1, \ldots, Y_n$ conditional on $\phi_1, \ldots, \phi_n$,

$$Y_i|\phi_i \sim f_i(Y_i|\phi_i), \quad i = 1, \ldots, n \tag{3.51}$$

$$\phi_i|G \sim G \tag{3.52}$$

$$G|\alpha, G_0 \sim DP(\alpha, G_0) \tag{3.53}$$

draws follows Polya urn method, and hence with positive probability we can classify the observations into $k \leq n$ clusters, i.e. $Y_1^*, \ldots, Y_k^*$ conditional on $\theta = \{\theta_1, ..., \theta_k\}$. Define the configuration as $s_j = j$ if $\phi_i = \theta_j$ and collect these in $s = \{s_1, \ldots, s_n\}$. Define $\phi_{-i} = \{\phi_1, ..., \phi_{i-1}, \phi_{i+1}, ..., \phi_n\}$. Let $n_{j,-k^i}$ be the number of $\{s_{m,-i}|s_{m,-i} = j\}$ in the set ϕ_{-i}. The Gibbs sampling iterate for one sweep is based on the following steps,

Step 1

Sample $\phi_i, s_i|\alpha, \phi_{-i}, k_{-i}, y$

$$\phi_i, s_i|\alpha, \phi_{-i}, k_{-i}, y = \begin{cases} \theta^*_{k_{-i}+1} \sim G_i, s_i = k_{-i}+1 & \text{with prob } c\alpha h_i(y_i) \\ \theta^*_1, s_i = 1 & \text{with prob } cn_{j,-1}\alpha f_1(y_i|\theta_1) \\ \theta^*_2, s_i = 2 & \text{with prob } cn_{j,-2}\alpha f_2(y_i|\theta_2) \\ \vdots & \\ \theta^*_{k_{-i}}, s_i = k_{-i} & \text{with prob } cn_{j,-k_{-i}}\alpha f_{k_{-i}}(y_i|\theta_{k_{-i}}) \end{cases} \tag{3.54}$$

Step 2

Sample $\theta^*_j|s, k, Y,\ j = 1, \ldots, k$

$$P(\theta^*_j|s, k, Y) \propto \prod_{i:s_i=j} f_i(y_i|\theta_j)g_0(\theta_j) \tag{3.55}$$

Mike West and Escobar (1994) recommended the second step to speed up convergence of the chain. After one sweep in the Gibbs sampling, we save the corresponding θ^*, s, k, and the predictive density is given by the Poly-urn prediction rule.

$$P(y_{n+1}|Y, k, s, \theta) = \frac{\alpha}{\alpha+n}\int f(y_{n+1}|\theta_{k+1})g_0(\theta^*_{k+1})d\theta_{k+1} + \sum_{i=1}^{k}\frac{n_i}{\alpha+n}f(y_{n+1}|\theta_i). \tag{3.56}$$

After R iteration, the Gibbs sampling gives the approximation of the predictive density,

$$P(y_{n+1}|Y) \approx \frac{1}{R}\sum_{\phi', s', k'} P(y_{n+1}|\phi', s', k'). \tag{3.57}$$

which has all parameter and distributional uncertainty integrated out.

3.9.1 Burn-in period

A burn-in period refers to a period with observations at the beginning of the Monte Carlo Markov Chain Simulation. Although after a large amount of iterations, the process will eventually converges to the desired (stationary) distribution, but initial samples may follow a very different distribution. This is because successive samples are not independent with each other. It is common to drop an initial sample, called a burn-in, and use the remainder to compute posterior quantities.

Chapter 4

Simulation study and Illustrative Example

4.1 Data

Data set comes from Price-Data. It is 5 minute data on SPY Exchange Traded Fund (ETF) from 9:35 GMT to 16:00 GMT, and covers 14 years from August 5th 1997 to May15th 2012. The SPY is designed to track the S&P 500 portfolio. As such, it provides a way to trade the index. The field order of the five minute data are: Date, Time, Open, High, Low, Close. For accuracy, data points from slow trading days were removed. These days usually are the major holidays such as Christmas, New Year's Day, or day after or before, Good Friday, Easter Monday, Labor Day, and so on. On average, the number of intraday return is 78.

Figure 4.1 provides a time series plot of daily return, RV($q = 1$), and logRV ($q = 1$) August 5th 1997 to May 15th 2012 for the SPY Exchange Traded Fund.

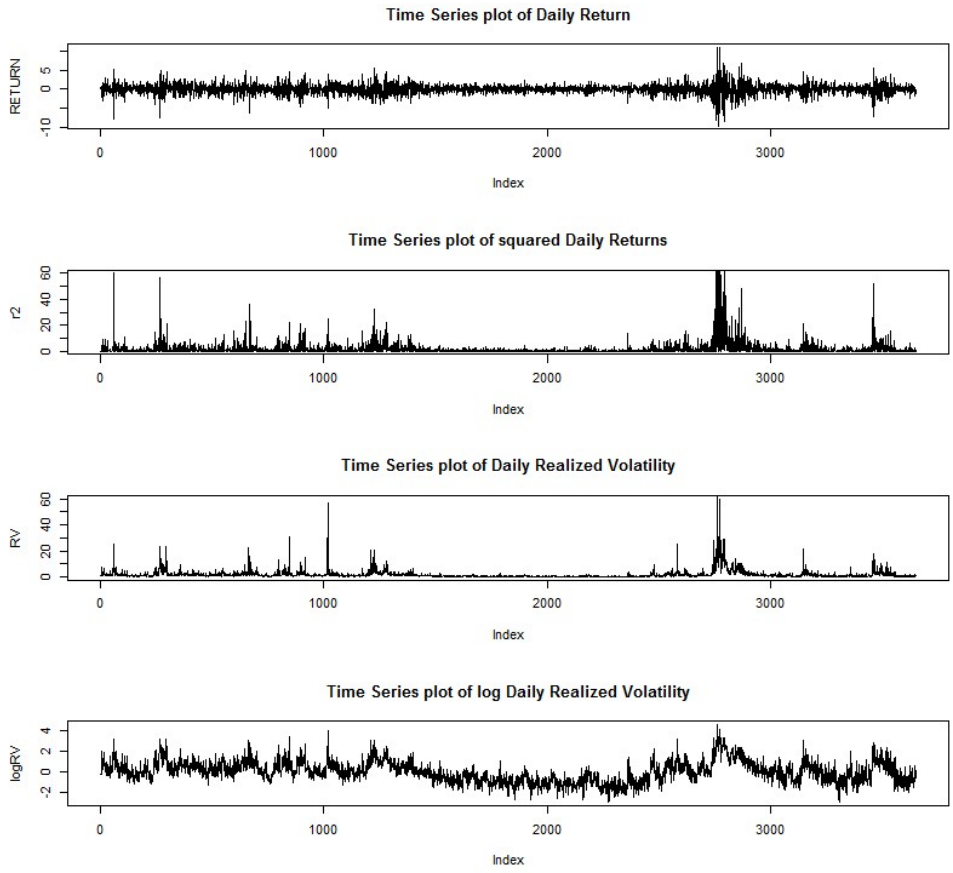

Figure 4.1: Time Series plot of Daily Return, Daily Realized Volatility, squared Realized Volatility, and log RV

variable	mean	variance	skewness	kurtosis	minimum	maximum
Return	0.00812829	1.845945	-0.1357056	6.404853	-9.470928	10.86478
Return^2	1.845506	28.62124	10.61657	169.5352	0	118.0434
RV(daily)	1.891949	13.49269	10.84123	210.417	0.05870844	103.288
RV (q=1)	1.792344	12.89174	11.16272	225.2714	0.04811407	103.6737
RV (q=2)	1.736698	11.91658	10.10289	175.8777	0.03900571	89.85874
RV (q=3)	1.703472	11.27304	9.587828	158.5268	0.03791243	85.67059

Table 4.1: Summary Statistics

variable	Mean	Variance	95% Density Interval
k	11.0992	11.40604	(4.4797250,17.71867)
α	0.9401094	0.168912	(0.1345709,1.745648)

Table 4.2: Summary Statistics for k and α

Table 4.1 shows the summary statistics for daily return and adjusted value of RV from August 5th 1997 to May 15th 2012. The daily RV is calculated from the 5 minute return data by Equation (2.15). Compare the variance of return with mean of RV, $q = 0$ has bias, $q = 1$ is improved, therefore we choose $q = 1$ for RV in our example. The variance of squared returns is a a lot larger than the RV. The sample autocorrelation function (ACF) for a series gives correlations between the time series and lagged values of the series for lags of $1, 2, 3, \ldots$. Figure 4.2 shows there appears to be significant autocorrelation in the Daily Realized Volatility and log RV. The partial autocorrelation plot in Figure 4.3 for log RV shows clear statistical significance for lags 1 and 6. The lags after 6 are at the borderline of statistical significance.

We first apply Bayesian semi-parametric estimation to daily return r_t for SPY. Figure 4.4 and 4.5 shows that the Bayesian density approaches for the predictive value gives thicker tails than normal distribution. Table 4.2 illustrates the return data are clustered in to 11 group on average. The next section will introduce the algorithm for the Bayesian semi-parametric approach in detail.

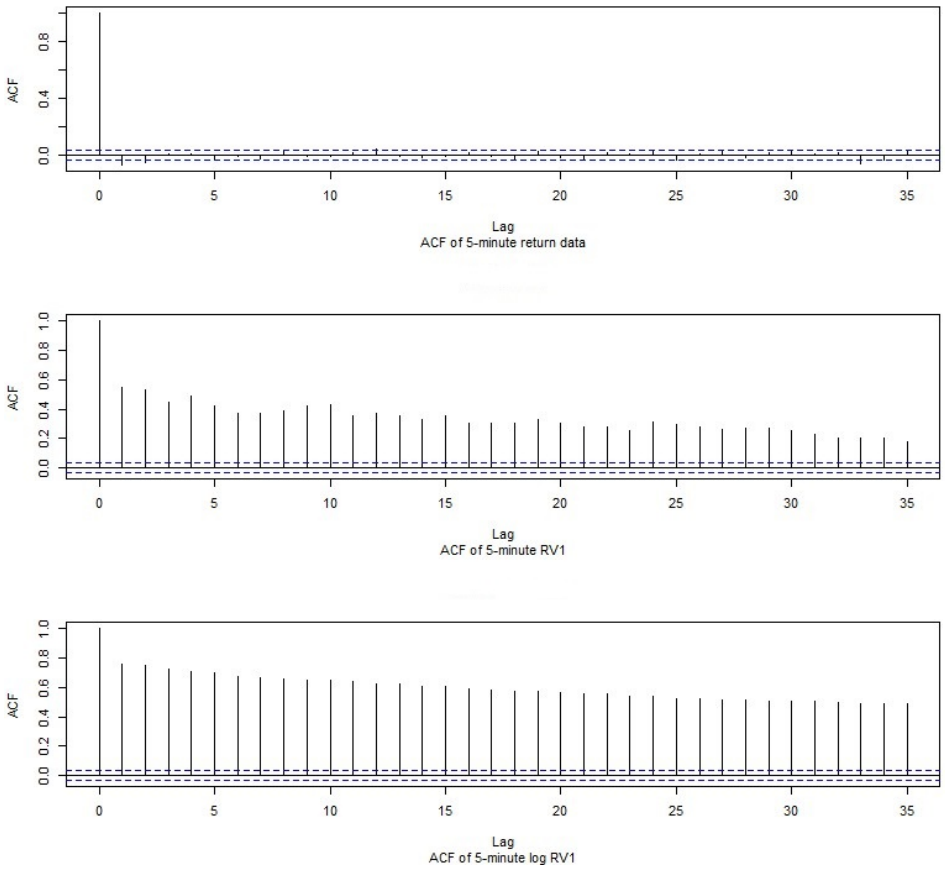

Figure 4.2: Plot of Autocorrelation Function on Return, Daily Realized Volatility, and log RV

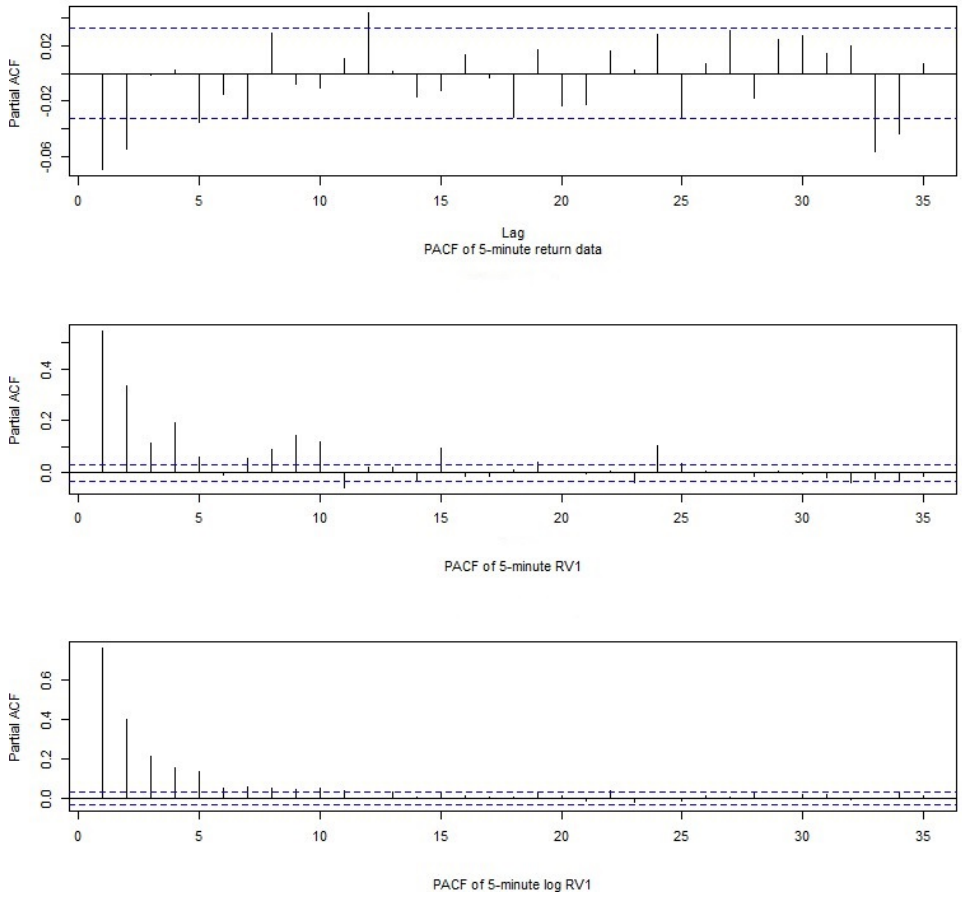

Figure 4.3: Plot of Partial Autocorrelation Function on Daily Realized Volatility, and log RV

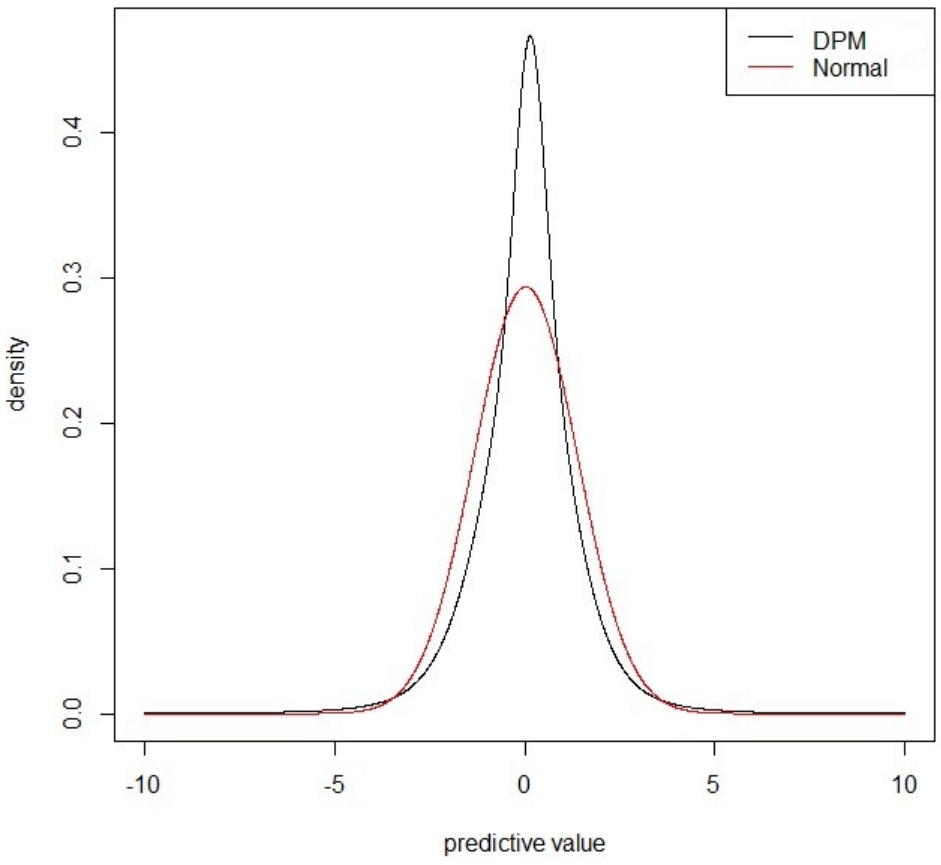

Figure 4.4: Comparison of normal (red), and DPM (black) pdfs of Return

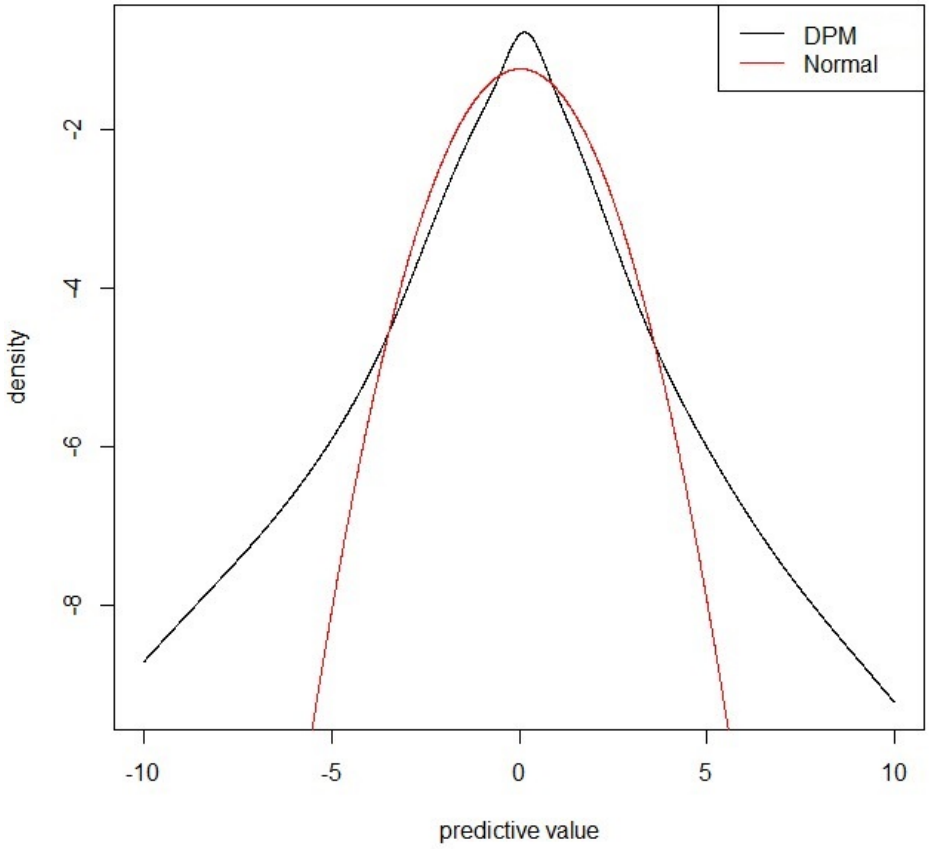

Figure 4.5: Comparison of normal (red) and DPM (black) log(pdfs) of Return

4.2 Estimation Method

4.2.1 HAR-DPM model

The univariate observations $Y = \{y_1, y_2, ..., y_n\}$ can be generated from HAR-DPM model, assume each observation y_i follows the time series model,

$$\begin{aligned} y_t &= \mu_t + X_t B + \sigma_t z_t, \quad z_t \sim N(0,1). \\ \phi_t|G &\sim G, \quad \phi_t = \{\mu_t, \sigma_t^2\}. \\ G|\alpha &\sim \mathrm{DP}(\mathrm{G}_0, \alpha). \\ \alpha &\sim \mathrm{Gamma}(\mathrm{a}, \mathrm{b}). \\ B &\sim N_k(b_0, B_0). \\ G_0 &\equiv NG(m, \tau, v_h, s_h). \end{aligned} \tag{4.1}$$

where

$$X_t = [y_{t-1}, \frac{1}{5}\sum_{i=1}^{5} y_{t-i}, \frac{1}{22}\sum_{i=1}^{22} y_{t-i}] \tag{4.2}$$

$$B = (\beta_1, \beta_2, \beta_3)^t \tag{4.3}$$

and $NG()$ denote a normal-gamma distribution such that

$$h_t \sim G(\frac{v_h}{2}, \frac{s_h}{2}) \tag{4.4}$$

$$\mu_t|h_t \sim N(m, \tau^{-1}h_t) \tag{4.5}$$

where $h_t = \sigma_t^{-2}$. In our study, we have Y to be the log-realized volatility, and hope to get the predictive density for the log-realized volatility. In this model, we learn about α from the data. The regression parameter B is assumed fixed but μ_t, σ_t^2 change over clusters.

For a set of given parameters $\mu = \{\mu_1, ..., \mu_n\}$, $\sigma = \{\sigma_1, ..., \sigma_n\}$, define $\phi_i = (\mu_i, \sigma_i)$ with $\phi = \{\phi_1, \phi_2, ..., \phi_n\}$. Draws from the Dirichlet process can be identical, and hence with positive probability the observations can be reduced into $k \leq n$ distinct clusters $\theta = \{\theta_1, ..., \theta_k\}$ with $\theta_t = (\mu_{s_t}, \sigma_{s_t})$. Define the configuration as $s_j = j$ if $\phi_i = \theta_j$, and then collect these in $s = \{s_1, ...s_n\}$. For a given value of B, and specified s_t, the time series model for observations $Y = \{y_1, y_2, ..., y_n\}$ can be rewritten as

$$y_t - X_t B = \mu_{s_t} + \sigma_{s_t} z_t, \quad z_t \sim N(0, 1) \tag{4.6}$$

Set $\zeta_t = y_t - X_t B$, then

$$\zeta_t = \mu_{s_t} + \sigma_{s_t} z_t, \quad z_t \sim N(0, 1) \tag{4.7}$$

which is the standard DPM model of the previous section. Consider univariate data $\zeta = \{\zeta_1, \zeta_2, ...\zeta_n\}$ are generated from the DPM model with

$$\zeta_i | \phi_i \sim f_i(\zeta_i | \phi_i) \tag{4.8}$$

$$\phi_i | G \sim G \tag{4.9}$$

$$G | \alpha, G_0 \sim DP(G_0, \alpha). \tag{4.10}$$

Therefore all the precious Gibbs sampling steps can be used on this model.

To sample B, note that for a given configuration s_t, a simple transformation for the time series model gives

$$\frac{y_t - \mu_{s_t}}{\sigma_{s_t}} = \frac{X_t}{\sigma_{s_t}} B + z_t, \quad z_t \sim N(0,1). \tag{4.11}$$

If we set $y_t^* = \frac{y_t - \mu_{s_t}}{\sigma_{s_t}}$, $X_t^* = \frac{X_t}{\sigma_{s_t}}$, the above equation becomes

$$y_t^* = X_t^* B + z_t, \quad z_t \sim N(0,1). \tag{4.12}$$

That is to say,

$$y_t^* \sim N_n(X_t^* B, I_n) \tag{4.13}$$

And the modified observed data $Y^* = \{y_1^*, y_2^*, ..., y_n^*\}$ have the distribution

$$Y^* \sim N_n(X^* B, I_n) \tag{4.14}$$

with prior distribution

$$B \sim N_k(b_0, B_0) \tag{4.15}$$

As a consequence of these assumptions,

$$B|Y^* \sim N_k(\bar{b}, B_1) \tag{4.16}$$

where

$$B_1 = [X^{*'} X^* + B_0^{-1}]^{-1} \tag{4.17}$$

$$\bar{b} = B_1[X^{*'} Y^* + B_0^{-1} b_0] \tag{4.18}$$

The conditional posterior distribution is standard, the posterior distribution of $(B|Y^*, X^*)$ can be find by using Gibbs sampler.

In algorithm form,

4.2.2 Algorithm: MCMC algorithm

The steps are based on Mike West and Escobar (1994).

Step 1

Choose a starting value of $\beta^{(0)}, \phi^{(0)}$, $\alpha^{(0)}$, get $k^{(0)}$, $\theta^{(0)}$, $s^{(0)}$ from $\phi^{(0)}$.

Step 2

At gth iteration,

1. sample $[\beta^{(g)}|Y^{(g-1)*}, X^{(g-1)*}]$.

2. sample $[\alpha^{(g)}|\alpha^{(g-1)}, k^{(g-1)}]$.

3. sample $[\phi^{(g)'}|\beta^{(g)}, \phi^{(g-1)}, s^{(g-1)}, \alpha^{(g)}, k^{(g-1)}, \zeta^{*(g-1)}]$, and get $k^{(g)'}$, $\theta^{(g)'}$, $s^{(g)'}$ from $\phi^{(g)'}$.

4. sample $[\theta^{(g)}|\beta^{(g)}, \theta^{(g)'}, s^{(g)'}, \alpha^{(g)}, k^{(g)'}, \zeta^{*(g-1)}]$, and get $k^{(g)}$, $\phi^{(g)}$, $s^{(g)}$ from $\theta^{(g)}$.

Step 3

Repeat step 2 until $g = R$, where R is the desired sample size.

Details in Step 2

- $\alpha^{(g)}$ can be sampled at each stage of the simulation in the following two steps,

1. sample $\eta^{(g)}$, where

$$(\eta^{(g)}|\alpha^{(g-1)}, k^{(g-1)}) \sim B(\alpha^{(g-1)} + 1, n) \tag{4.19}$$

2. sample $\alpha^{(g)}$ from the mixture of two gamma densities

$$\alpha^{(g)}|\eta^{(g)}, k^{(g-1)} \sim \begin{cases} G(a + k^{(g-1)}, b - \log(\eta^{(g)})) & \text{with prob } \pi_{\eta}^{(g)} \\ G(a + k^{(g-1)} - 1, b - \log(\eta^{(g)})) & \text{with prob } 1 - \pi_{\eta}^{(g)} \end{cases} \tag{4.20}$$

where

$$\pi_{\eta}^{(g)} = \frac{n(b - \log(\eta^{(g)}))}{a + k - 1 + n(b - \log(\eta^{(g)}))} \tag{4.21}$$

- $\phi^{(g)'}$ can be sampled at each stage of the simulation (for $g = 2, ..., R$) in the following steps. For simplification, we ignore the upper notation (g) for α, k, θ, ϕ since they are from one swap in g-th iteration. Define $\phi_{-i} = \{\phi_1, ..., \phi_{i-1}, \phi_{i+1}, ..., \phi_n\}$, and similarly $s_{-i} = \{s_1, ..., s_{i-1}, s_{i+1}, ..., s_n\}$. Removing ϕ_i and s_i may reduce the number of clusters by 1 if ϕ_i is the only member in this cluster θ_{s_i}. Sample element of ϕ sequentially by drawing from the distribution of

$$[\phi_i|\phi_{-i}, G], \quad i = 1, ..., n \tag{4.22}$$

That is to say,

$$\phi_i, s_i|\alpha, \phi_{-i}, k_{-i}, \zeta = \begin{cases} \theta_{k_{-i}+1} \sim G_i, & \\ dG_i \propto f(\zeta_i|\phi)dG_0(\phi), s_i = k_{-i}+1 & \\ & \text{with prob } c\alpha h_i(\zeta_i) \\ \theta_1, s_i = 1 & \text{with prob } cn_{j,-1}\alpha f_1(\zeta_i|\theta_1) \\ \theta_2, s_i = 2 & \text{with prob } cn_{j,-2}\alpha f_2(\zeta_i|\theta_2) \\ \vdots & \\ \theta_{k_{-i}}, s_i = k_{-i} & \text{with prob } cn_{j,-k_{-i}}\alpha f_{k_{-i}}(\zeta_i|\theta_{k_{-i}}) \end{cases} \tag{4.23}$$

where c is a normalizing constant; the density function is $h_i(\zeta_i) = \int f_i(\zeta_i|\phi_i)g_0(\phi_i)d\phi_i$; and $n_{j,-i}$ is the number of $\{s_{m,-i}|s_{m,-i} = j\}$ in the set ϕ^i, which can be understood as the number of observations in each cluster.

- $\theta^{(g)}$ can be sampled at each stage of the simulation (for $g = 2, ..., G$) as

$$p(\theta_j^{(g)}|\theta^{(g)'}, s^{(g)}, \alpha^{(g)}, k^{(g)}, \zeta^g) \propto \prod_{i:s_i=j} f_i(\zeta_i|\theta_j^{(g)})g_0(\theta_j^{(g)}) \tag{4.24}$$

In the simulation study, given our conjugate prior we have $\zeta_i|\mu_i, \sigma_i^2 \sim N(\mu_i, \sigma_i^2)$ with $\phi_i = (\mu_i, \sigma_i^2)$, and $h_i = \sigma_i^{-2}$. If we set

$$h_i \sim G(\frac{v_h}{2}, \frac{s_h}{2}) \tag{4.25}$$

$$\mu_i|h_i \sim N(m, \tau^{-1}h_j^{-1}) \tag{4.26}$$

For a given set of hyperparameters v_h, s_h, τ, m, we may have

$$\zeta_i \sim \text{Student t}(\text{mean} = m, \text{scale parameter} = \frac{\tau^{-1}+1}{s_h/v_h}, \text{df} = v_h) \tag{4.27}$$

And the posterior distribution can be recognized as (Geweke,2005)

$$h_i|\zeta \sim G(\frac{\bar{v}_h}{2}, \frac{\bar{s}_h}{2}) \tag{4.28}$$

$$u_i|h_i, \zeta \sim N(\bar{\mu}, (\bar{\tau}h_i)^2) \tag{4.29}$$

with

$$\bar{\tau} = \tau + n_i \tag{4.30}$$

$$\bar{\mu} = (\tau + n_i)^{-1}(\tau m + n_i\bar{\zeta_i}) \tag{4.31}$$

$$\bar{v}_h = v_h + n_i \tag{4.32}$$

$$\bar{s}_h = s_h + s_i^2 + (\bar{\mu} - \bar{\zeta_i})^2 n_i + (\bar{\mu} - m)^2\tau \tag{4.33}$$

$$\bar{\zeta_i} = \frac{1}{n_i}\sum_{j=1}^{n} \text{I}\{\zeta_j|s_j = i\} \tag{4.34}$$

where n_i is the sample size of these obeservations, s_j is the sum of squared errors of $\{\zeta_j|s_j = i\}$. We choose $v_h = 5,\ s_h = 4,\ \tau = 10,\ m = 0$, set the initial parameters as $\alpha^{(0)} = 0.5,\ \phi_i^{(0)} = (0.007, 0.5)$, for $i = 1, ..., \lfloor\frac{n}{2}\rfloor$, and $\phi_i^{(0)} = (0.009, 1.5)$, for $i = \lfloor\frac{n}{2}\rfloor + 1, ..., n$ and $\bar{b}^{(0)} = 0,\ B_1^{(0)} = 10I$ in the simulation study.

Burn-in Period

We set $G = 5000$ in the MCMC sampling and drop the first 200 observations as burn-in.

4.2.3 Prediction

The predictive density for each iteration in the Gibbs sampling is given by the following mixture,

$$p(\zeta_{n+1}|\theta, s, k, y) = \frac{\alpha}{\alpha+n}\int f(\zeta_{n+1}|\theta_{k+1})g_0(\theta_{k+1})d\theta_{k+1} + \sum_{i=1}^{k}\frac{n_i}{\alpha+n}f(\zeta_{n+1}|\theta_i) \quad (4.35)$$

This implies a predictive density for y_{t+1}

$$\begin{aligned} p(y_{t+1}|\theta, s, k, y, \beta, X_t) &= \sum_{i=1}^{k}\frac{n_i}{\alpha+n}f(y_{t+1}|X_t\beta + \mu_{s_t,-i}, \sigma^2_{-i}) \quad (4.36)\\ &\quad + \frac{\alpha}{\alpha+n}\int\int f(y_{t+1}|X_t\beta + \mu_{s_t}, \sigma^2)g_0(\mu_{s_t}, \sigma^2)d\mu_{s_t}d\sigma^2 \end{aligned}$$

where the last term is a Student-t density with mean $m + X_t\beta$, scale $\frac{\tau^{-1}+1}{s_h/v_h}$, and degree of freedom v_h. Denote R be the total number of iteration, we may approximate the predictive density as the average of the predicted value (4.35) in each MCMC swap,

$$p(\zeta_{n+1}|\zeta, y) \approx \frac{1}{R}\sum_{\theta', s', k'} p(\zeta_{n+1}|\theta', s', k', y) \quad (4.37)$$

and similarly, the density of (4.37) is

$$p(y_{t+1}|y, X_t) \approx \frac{1}{R}\sum_{\theta', s', k', \beta, X_t} p(y_{t+1}|\theta', s', k', \beta, X_t) \quad (4.38)$$

variable	Mean	Variance	95% Density Interval
k	4.0874	9.487659	(-1.9498, 10.1246)
α	0.3338527	0.1008219	(-0.2885, 0.9562)

Table 4.3: Summary Statistics for k and α

$$y_t = \mu_t + \beta_1 y_{t-1} + \beta_2 \frac{1}{5}\sum_{i=1}^{5} y_{t-i} + \beta_3 \frac{1}{22}\sum_{i=1}^{22} y_{t-i} + \sigma_{s_t} z_t, \quad z_t \sim N(0,1)$$

variable	HAR-DPM, 0.95 Density Interval	OLS, 95% Confidence Interval
μ	...	-0.003991188 (-0.03800895, 0.03002657)
β_1	0.1487661 (0.1423151, 0.1552171)	0.1907763160 (0.120935100, 0.2606175)
β_2	0.5193140 (0.5102717, 0.5283563)	0.5647482570 (0.448933600, 0.6805629)
β_3	0.1809498 (0.1771329, 0.1847666)	0.1915126550 (0.095954740, 0.2870706)
σ_{s_t}	...	0.3478219
k	4.0874000 (-1.949800, 10.124600)	...
α	0.3338527 (-0.288500, 0.9562000)	...

Table 4.4: Comparison Table of Estimated Value. This table shows the posterior mean and 0.95 density interval of k, α, β, and OLS estimator with 95% confidence interval of μ, k, α, β.

4.3 Results

We set number of iteration in MCMC as 5000. In order to understand the Dirichlet Process mixture model in this algorithm, we can have a look at the properties of the concentration parameter α and number of cluster k. Table 4.3 shows that sample draws are clustered in about four groups on average. From the trace plot of Figure 4.6, the fluctuation of k and α are very similar in behavior, they have the same wave in the Markov Chain movement, when k goes higher and α becomes larger; when k moves down, α becomes smaller. There is some autocorrelation in the output, but the trace plot shows that the posterior density is effectively explored. Table 4.4 illustrates the estimator of β from DPM model is very similar with the OLS estimator.

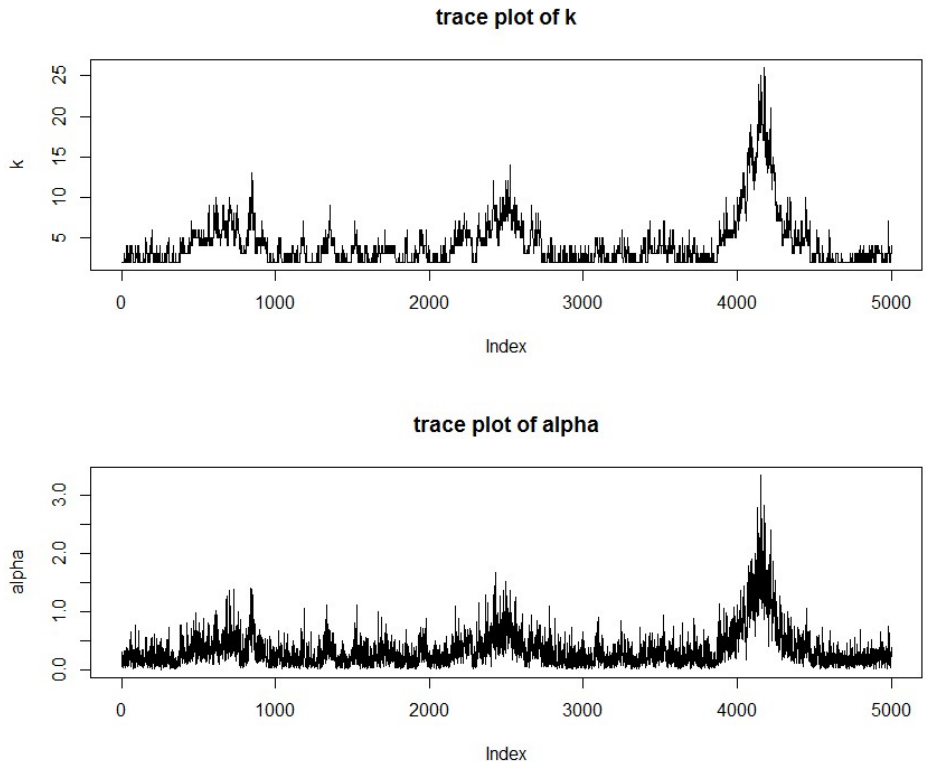

Figure 4.6: Trace Plot of α and k in the MCMC iteration

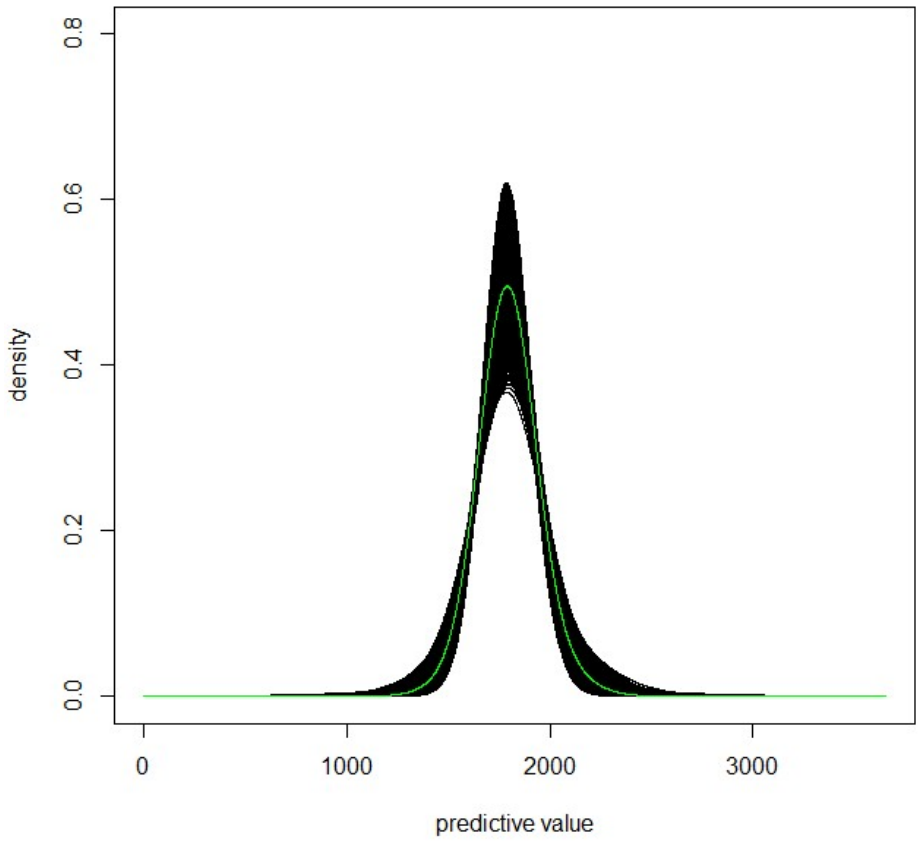

Figure 4.7: Trace Plot of pdfs of logRV

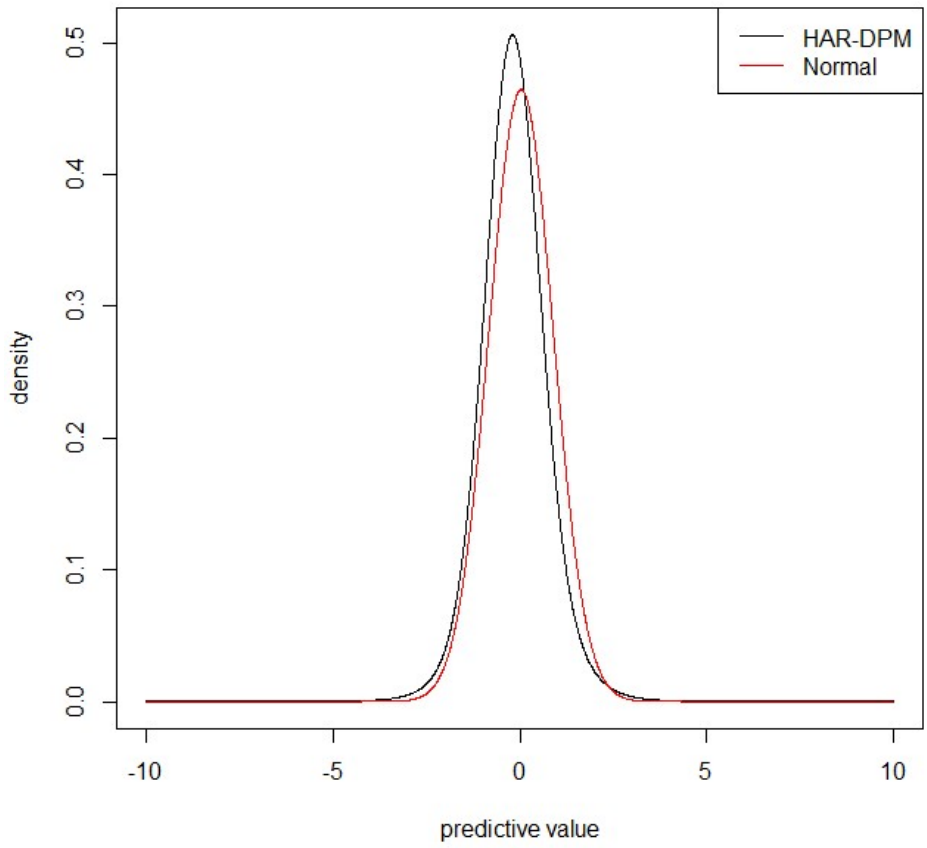

Figure 4.8: Comparison of normal (red) and HAR-DPM (black) pdf of logRV

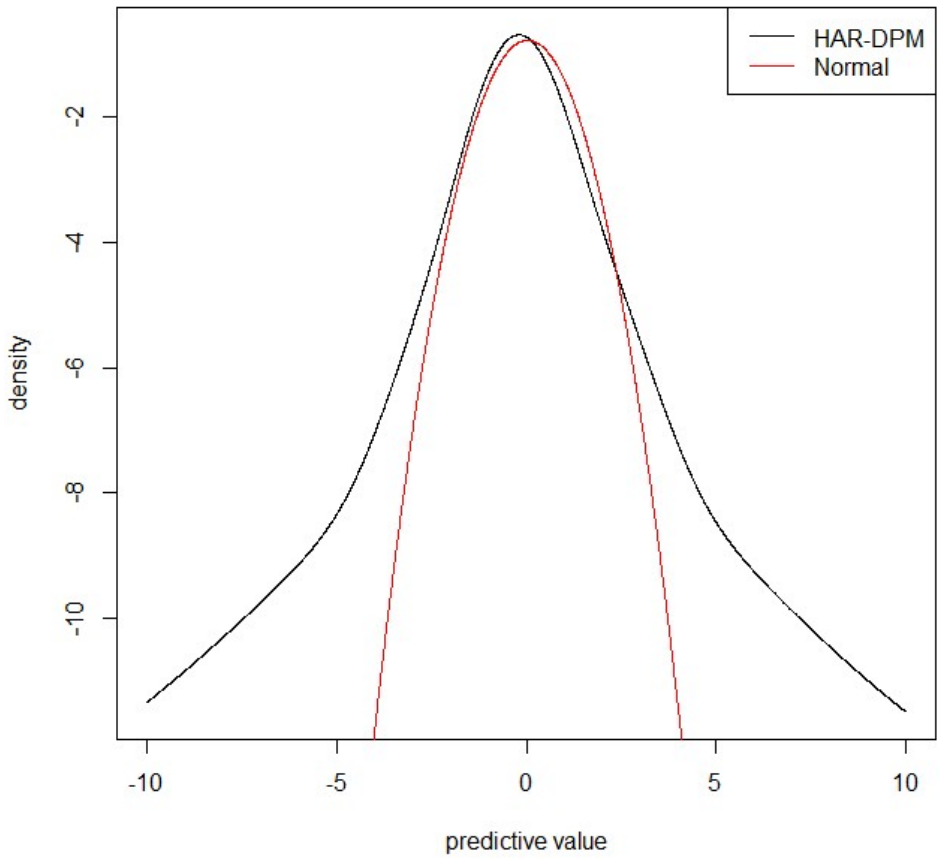

Figure 4.9: Comparison of normal (red) and HAR-DPM (black) log(pdfs) of logRV

Figure 4.8 and shows the predictive density of logarithm of realized volatility. The black curve is produced by the DPM model; and the red one is constructed by using a normal approximation with mean $\mu = \frac{1}{n}\sum_{i=1}^{n} y_i$ and variance $\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} E(y_i - \mu)^2$ from our daily log-realized volatility (with $q = 1$). Figure 4.7 gives an idea on the movement of predictive density during iterations, the predictive density are smooth and nicely displayed without much fluctuation within the iteration. Figure 4.9 show the logarithm of predictive densities, both are similar around 0 but the DPM covers much thicker tails for both positive and negative values of log-RV.

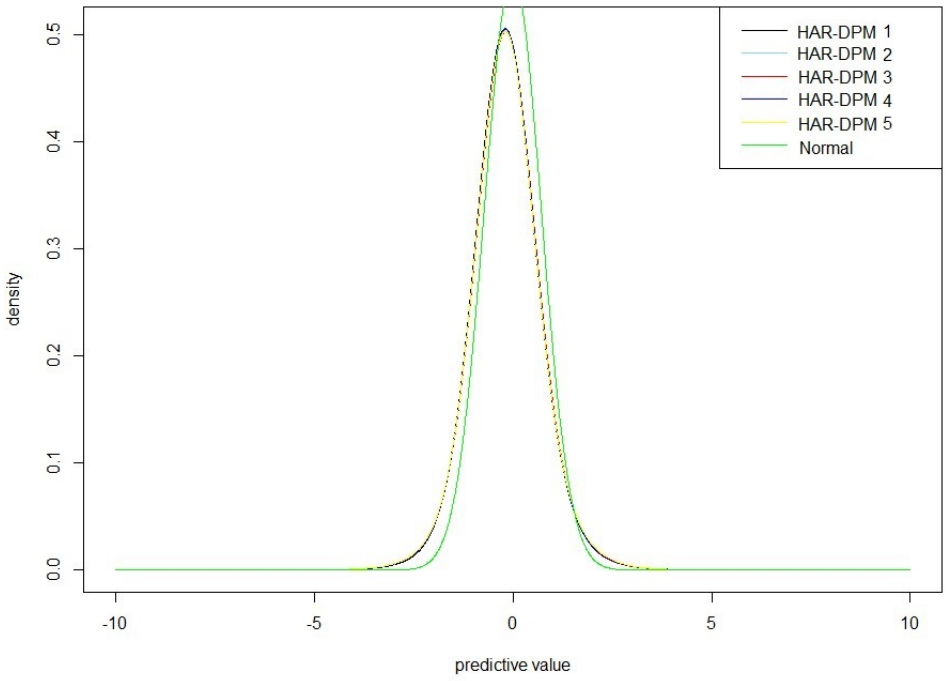

Figure 4.10: Comparison of normal (red) and HAR-DPM (black) pdf with different Monte Carlo seeds for the random number generator

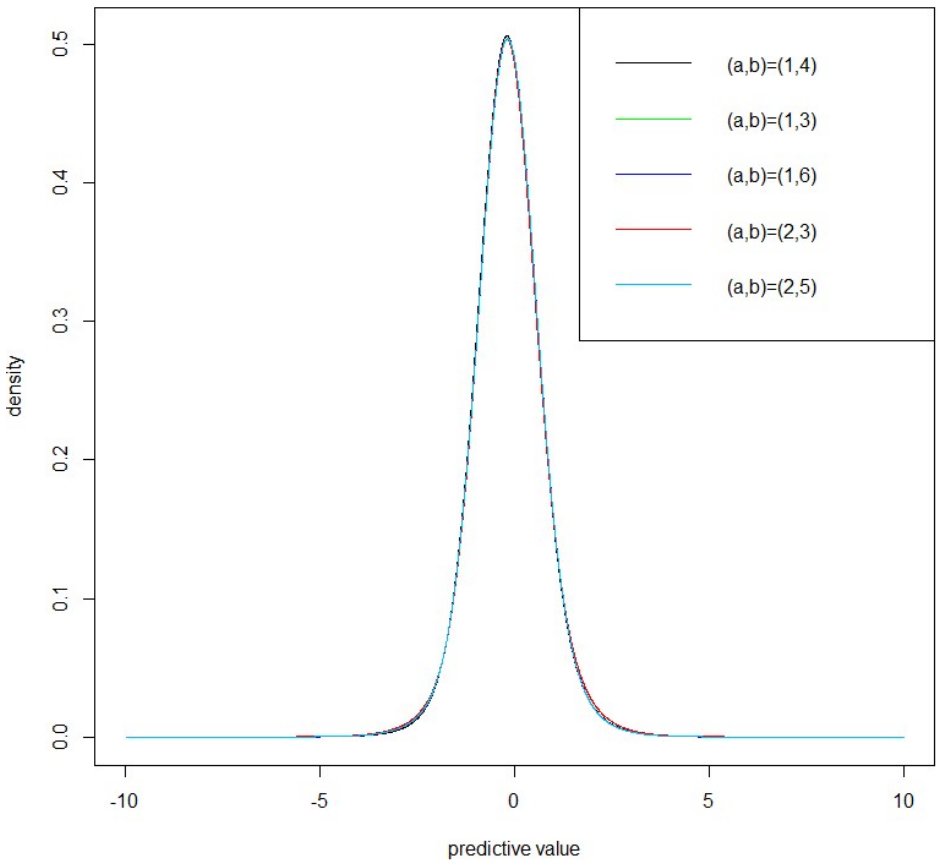

Figure 4.11: Comparison of normal (red) and HAR-DPM (black) log(pdfs) with different priors

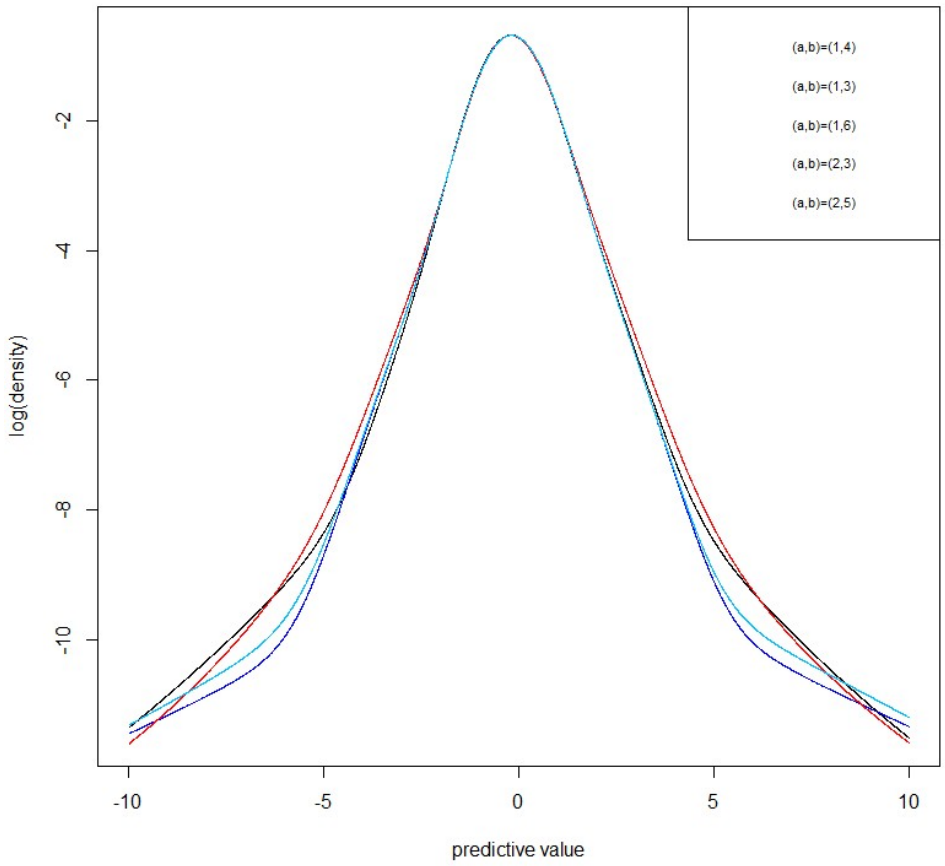

Figure 4.12: Comparison of normal (red) and HAR-DPM (black) log(pdfs) with different priors

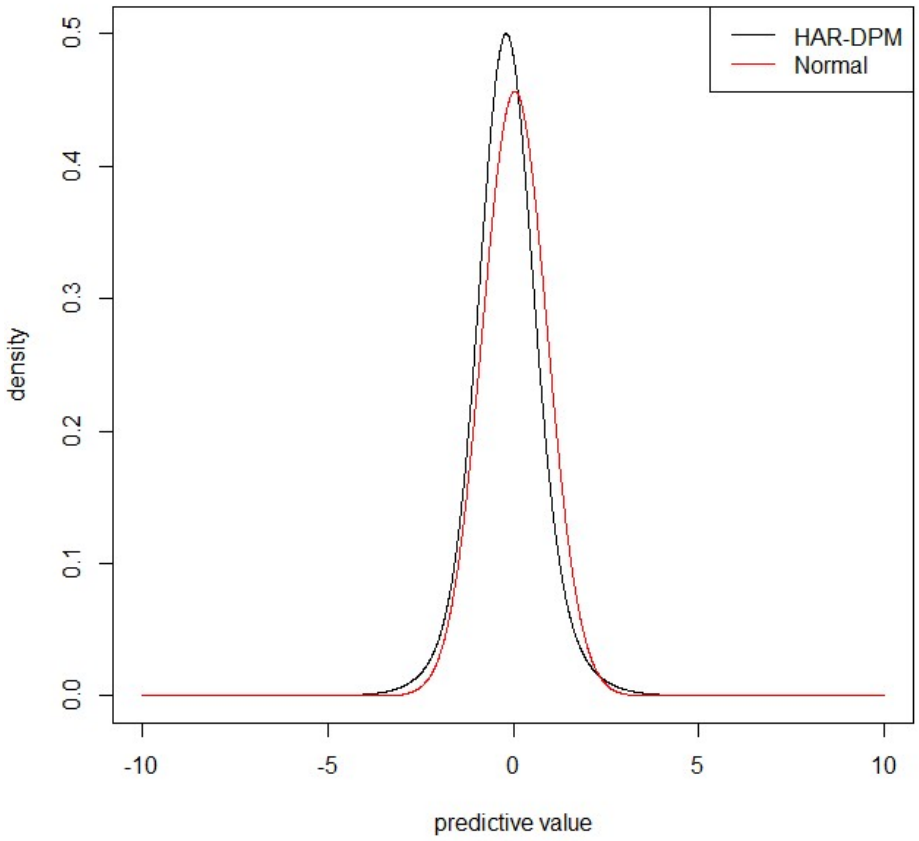

Figure 4.13: Comparison of normal (red) and HAR-DPM (black) pdfs for a larger MCMC iteration

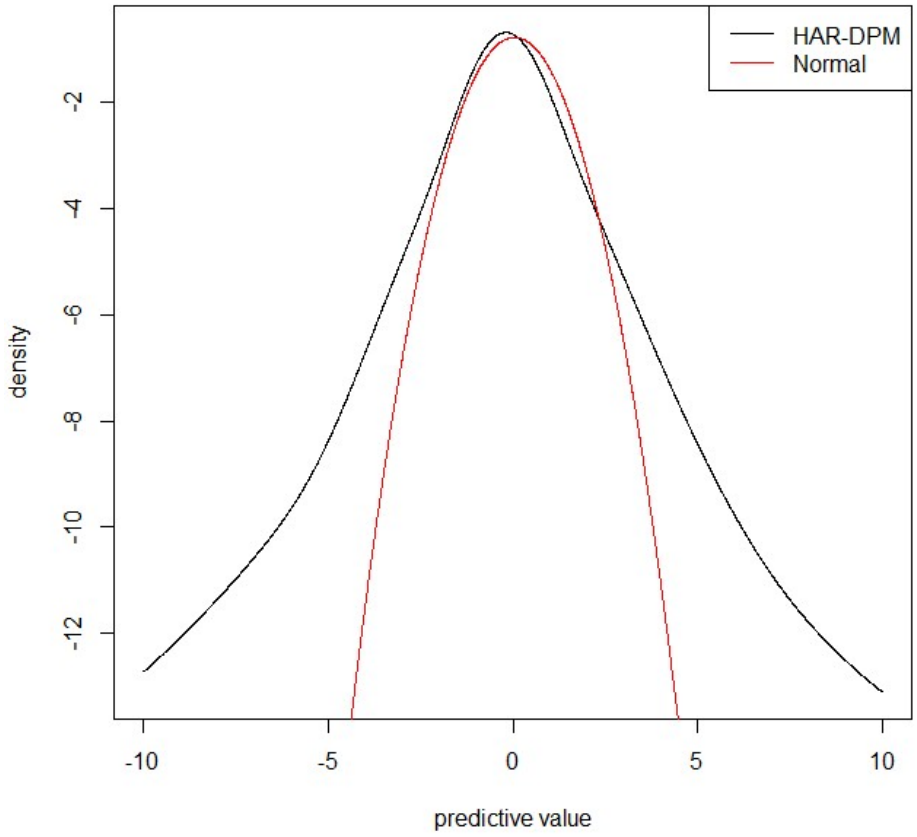

Figure 4.14: Comparison of normal (red) and HAR-DPM (black) log(pdfs) for a larger MCMC iteration

Figure 4.10 shows the predictive density of log-RV with respect to different Monte Carlo seeds for the random number generator. Each gives the same density. Figure 4.11 shows the predictive density of log-RV with respect to different priors. The densities are almost the same for each case. Figure 4.12 shows the associated log density plots. All of them indicate thick tails. Figure 4.13 displays predictive density estimation for 10000 MCMC iterations. The proposed method provides a robust prediction of realized volatility. I observe fat tails in log-RV, while the Gaussian model neglects the thick tails of log-RV. The existence of fat tails may be due to jumps in realized volatility. My results have important implications. First, in contrast to the literature, the innovation density for log RV has thick tails and is slightly asymmetric. Second, to the extent that log-RV is a measure of risk, then conventional models significantly understates risk from larger values of log-RV. Risk measurement that ignores this may lead to significantly lower capital controls.

Chapter 5

Appendix

5.1 R code for DMP

```
#libraries
{
  library(Matrix)
  library(MCMCpack)
  library(mvtnorm)
  library(stats)
}

# Import Data
{
  RV <- read.table("C:/Users/Tina/Dropbox/Thesis/coding/data.txt")[,5]
  #descripetion: date, return (5min) close-to-close 4pm,
  #return from daily data, RV, RV1, RV2, RV3
```

```
  logRV <- log(RV)
  n <- length(RV)
}

# function to return a matrix with row = n, and column = 3
#(used for estimation of beta)
find_OLSX<-function(logRV){

  OLSX <- matrix(rep(0,(n)*3),nrow=(n),ncol=3,byrow = TRUE)

  for (i in 1:4){ OLSX[i,] <- cbind(logRV[i],0,0) }

  for (i in 5:21){
    weeklogRV  <- mean(logRV[(i-4):i])
    OLSX[i,] <- cbind(logRV[i],weeklogRV,0)
  }

  for (i in 22:(n)){
    weeklogRV  <- mean(logRV[(i-4):i])
    monthlogRV <- mean(logRV[(i-21):i])
    OLSX[i,] <- c(logRV[i],weeklogRV,monthlogRV)
  }

  OLSX
```

```
}


# Initial guesses
{
  u <- c(rep(0.007,n/2),rep(0.009,n/2))
  V <- c(rep(0.5,n/2),rep(1.5,n/2))
  s <- c(rep(1,n/2),rep(2,n/2))
  meanBeta0 <- c(0,0,0)
  varBeta0  <- 10*diag(3)
  Beta0  <- meanBeta0 + t(chol(varBeta0)) %*% rnorm(3)
  OLSX <- find_OLSX(logRV)
  y    <- c(logRV[1],logRV[2:n]- OLSX %*% Beta0)
  mean(y)
  var(y)
  phi   <- cbind(u,V)
  theta <- unique(phi)
  n_u <-theta[,1]
  n_V <-theta[,2]
  n_theta <- c(n/2,n/2)
  k <- length(unique(u))
}


##initial parameters
{
```

```
  ss    <- 4
  S     <- 4
  m     <- 0   ##mu_mean
  tau   <- 10
  S     <- S +(y-m)^2/(1+tau)
  X     <- tau/(1+tau)
  x     <- (m+tau*y)/(1+tau)
  M     <- (1+tau)*S/ss
  c_s   <- gamma((1+ss)/2)*(gamma(ss/2)^(-1))*ss^(-0.5)
  alpha <- 0.5
  v_h <- 1
  s_h <- 1
  tau <- 10

}



# gives s,n_theta, find s
find_s <- function(a,b){
  n   <- length(a)
  n_a <- a
  s   <- rep(1:n)
  count<-rep(1,n)
  for(i in 2:n){
```

```
    if (length(unique(a[i] == a[-i]))==2){
      p <- (which(a[i] == a[-i]))[1]
      if (b[i]== b[p]){
        n_a[i] <- 0
        n_a[p] <- a[p]
        s[p]   <- nnzero(n_a[1:p[1]])
        s[i]   <- s[p]
        count[i]<-0
        count[p]<-1+count[p]}
    }
    if (length(unique(a[i]== a[-i]))==1){
      s[i]<-nnzero(n_a[1:i])}
  }
  cbind(a,b,s,count)
}
# find_s(u,V)

# function to update theta
update_theta <- function(u,V,y,n_u,n_V,s,ntheta){

  n <- length(y)
  phi <-cbind(u,V)

  k<-length(n_u)
```

```
p <- q <- rep(0,k)
#prob of being the j group, j = 1, ..., k
for (i in 1:n){

  #### intial treatments
  if (ntheta[s[i]]==1){
    if(s[i]!=k){
      ntheta[s[i]]<-ntheta[k]
      ## passing the values from k to s[i] th group
      n_u[s[i]]<-n_u[k]
      n_V[s[i]]<-n_V[k]
    }
    ntheta<-ntheta[1:(k-1)]
    ## deleting the last entries
    n_u<-n_u[1:(k-1)]
    n_V<-n_V[1:(k-1)]
    s[c(which(s==k))]=s[i]
    k<-k-1 }
  else {ntheta[s[i]]<- ntheta[s[i]]-1}
  ## the prob of having a new parameter,
  #sum_prob for summing the probabilities.
  prec<-sqrt(v_h/s_h*tau/(1+tau))
  sum_prob<- q0 <- p0 <- alpha*dt((y[i]-m)/prec,v_h)/prec
  #### here is the most problematic part:
```

```
for (j in 1:k){
  ### prob for being j th group
  q[j]<- ntheta[j]*dnorm(y[i],n_u[j],sqrt(n_V[j]))
  #q[j] <- ntheta[j]*(2*n_V[j])^(-0.5)*exp(-0.5*(y[i]-n_u[j])^2/n_V[j])
  sum_prob<-sum_prob+q[j]
} # for j

p0<-q0<- q0/sum_prob              # normalizing p0, q0
for (j in 1:k){
  q[j] <- q[j]/sum_prob          # normaling q[j]
  p[j] <- q0 + sum(q[1:j])       # cumulating p[j]
}



deci <- runif(1)
### setting a decision number

if (deci < p0){
  ### if it is a observation with new parameter:
  tau_bar<-tau+1
  mu_bar<-(tau*m+y[i])/tau_bar
  v_bar<-v_h+1
```

```
    s_bar<-s_h+(mu_bar-y[i])^2+(mu_bar-m)^2*tau
    V[i]<-1/rgamma(1,v_bar/2,s_bar/2)
    u[i]<-rnorm(1,mu_bar,sqrt(V[i]/tau_bar))
    n_u <- c(n_u, u[i])
    ### adding one more entries
    n_V <- c(n_V,V[i])
    ntheta <- c(ntheta,1)
    ### having one more group
    s[i]<-k<-k+1
    ### assign it to be the last group
  }
  else {
    if (deci < p[1] & deci >= p0){
      u[i] <- n_u[1]
      ### passing the value to individual u and V
      V[i] <- n_V[1]
      ntheta[1] <- ntheta[1]+1
      ### adding one to the number of the member in the first group
      s[i]<-1
      ###assign the i th element to the first group
    } ## if deci first group
    else{
      for (j in 2:k){
        #### for other groups
```

```
          if (deci < p[j] & deci >= p[j-1]){
            u[i] <- n_u[j]
            ### passing the new values to individual u and V
            V[i] <- n_V[j]
            ntheta[j] <-ntheta[j]+1
            ### adding one more member
            s[i]<-j
            ### assign it to be the last group
            break}  ## if deci the j th group (j>2)
        }  ## for j
      }  ## else 2
    } # else 1

  }# for i

  #### data frame output :

  data1<- data.frame(u,V,y,s)
  k<-length(n_u)
  data2<-data.frame(n_u,n_V,ntheta)
  c(data1,data2,data.frame(k))
}
# update_theta(u,V,y,theta[,1],theta[,2],s,n_theta)
```

```
# function to sample theta|s,k,y,
#second step to improve sample efficiency
sample_theta <- function (theta,s,k,y,ntheta){
  ##parameters

  n   <- length(y)
  n_u <- theta[,1]
  n_V <- theta[,2]
  for (j in 1:k){
    # i-th obs in each cluster j
    tau_bar <- tau + ntheta[j]
    y_bar   <- sum(y[which(s==j)])/ntheta[j] #mn1,  y_ bar_j
    u_bar   <- (tau*m + ntheta[j]*y_bar)/(tau+ntheta[j])  #mu bar
    v_h_bar <- v_h + ntheta[j]
    ss1<-sum(y[which(s==j)]^2)- y_bar^2*ntheta[j]
    ## ss = mn2 - number*pow( mn1 , 2 );
    s_h_bar <- s_h +ss1 +(u_bar-y_bar)^2*ntheta[j] +(u_bar-m)^2*tau
    n_V[j]  <- 1/(rgamma(1,v_h_bar/2,s_h_bar/2))
    n_u[j]  <- rnorm(1,u_bar,sqrt(n_V[j]/tau_bar))
  }
  cbind(n_u,n_V)
}
#sample_theta(theta,s,k,y,n_theta)
```

```
# function to update beta and y in the HAR-RV Model
update_Beta_y<-function(m_B,v_B,X,y,theta,s){
  u  <- v <- rep(0,n)
  for (i in 1:n){
    u[i] <-theta[s[i],1]
    v[i] <- theta[s[i],2]
  }

  XX <- cbind((X[,1]/v),(X[,2]/v),(X[,3]/v))[22:(n-1),]
  YY <- ((logRV-u)/v)[23:n]

  varB  <- solve( t(XX) %*% XX + solve(v_B))
  meanB <- varB %*% (t(XX) %*% YY + solve(v_B) %*% m_B)
  B <- meanB + t(chol(varB)) %*% rnorm(3)

  y <- c(logRV[1:22],logRV[23:n] - XX %*% B)

  data1 <- data.frame(y)
  data2 <- data.frame(B,meanB,varB)
  c(data1,data2)
}
update_Beta_y(meanBeta0,varBeta0,OLSX,y,theta,s)
```

```
# function to find the predictive density of y_n+1  (6)
predict_y <- function (theta,y,k,n_theta){

  n_u <- theta[,1]
  n_V <- theta[,2]
  sigma <- sqrt( (1/tau+1)* s_h /v_h )
  f1  <- 1/sigma*dt((y-m)/sigma,v_h)*alpha/(alpha+n)
  f2  <- 0
  for (i in 1:k){

    f2 <- f2 + dnorm(y,n_u[i],n_V[i]^(0.5))*n_theta[i]/(alpha+n)
  }
  p <- f1+f2
  p
}
#predict_y(theta,y,k,n_theta)
integrate(function(x){predict_y(theta,x,k,n_theta)},-Inf, Inf)

predict_logRV <- function (theta,k,n_theta,B,X,logRV,s,alpha){
  n <- length(s)
  n_u<-theta[,1]
  n_V<-theta[,2]
  sigma <- sqrt((1/tau+1)*s_h/v_h)
  f1  <- 1/sigma*dt((logRV -  X %*% B  - m)/sigma,v_h)*alpha/(alpha+n)
```

```
  f2  <- 0
  for (i in 1:k){
    f2 <- f2 +
      dnorm(logRV, X %*% B + n_u[i],n_V[i]^(0.5))*n_theta[i]/(alpha+n)
  }

  f1 + f2

}

#function to sample alpha
sample_alpha <- function(alpha, k){
  a   <- 1
  b   <- 4
  eta <- rbeta(1, alpha+1,n)
  pai_eta<-(a+k-1)/((n*(b-log(eta)))+a+k-1)
  p_eta<-runif(1)
  if (p_eta <= pai_eta) {
    alpha<-rgamma(1,shape=a+k,rate=b-log(eta))
  }
  if (p_eta >  pai_eta) {
    alpha<-rgamma(1,shape=a+k-1,rate=b-log(eta))
  }
```

```
  alpha
}
#sample_alpha(0.5,eta,k)

#main loop
{
  alpha <- 0.5
  pre_y <- seq(-10,10,20/(n-1))
  NN    <- 5000 #number of iteration

  list.s      <- vector("list",NN)
  array.k     <- matrix(rep(0,NN), ncol=NN)
  list.theta  <- vector("list",NN)
  list.y      <- vector("list",NN)
  list.logRV  <- vector("list",NN)
  array.alpha <- matrix(rep(0,n*NN), ncol=NN)
  list.beta   <- vector("list",NN)
  list.betaM  <- vector("list",NN)
  list.betaV  <- vector("list",NN)

  upd         <- update_Beta_y(meanBeta0,varBeta0,OLSX,y,theta,s)
  y           <- upd$y

  list.betaM[[1]]  <- upd$meanB
```

```
list.betaV[[1]]  <- cbind(upd$X1, upd$X2, upd$X3)

phi              <- unique(cbind(u,V))
first.data       <- update_theta(u,V,y,phi[,1],phi[,2],s,n_theta)  #first iterati

alpha            <- sample_alpha(alpha,first.data[[8]])
list.s[[1]]      <- first.data$s
array.k[1]      <- first.data[[8]]
list.theta[[1]] <- matrix(c(first.data$n_u,first.data$n_V),
                          ncol=2,nrow=first.data[[8]])
list.beta [[1]] <- upd$B
list.logRV[[1]] <- predict_logRV(list.theta[[1]],first.data[[8]],
                                 first.data$ntheta,upd$B,OLSX[n,],
                                 pre_y,first.data$s,alpha)
array.alpha[,1] <- alpha

for (i in 2:NN){

  list.betaM[[i]] <- upd$meanB
  list.betaV[[i]] <- cbind(upd$X1, upd$X2, upd$X3)
  upd             <- update_Beta_y(list.betaM[[i]],
                                   list.betaV[[i]],OLSX,y,list.theta[[i-1]],
                                   first.data$s)
  y               <- upd$y
```

```
    alpha           <- sample_alpha(alpha,first.data[[8]])
    array.alpha[,i] <- alpha
    first.data      <- update_theta(first.data$u,
                                    first.data$V,y,first.data$n_u,
                                    first.data$n_V,first.data$s,
                                    first.data$ntheta)
    list.s[[i]]     <- first.data$s
    array.k[i]     <- first.data[[8]]
    list.theta[[i]] <- matrix(c(first.data$n_u,
                                first.data$n_V),ncol=2,nrow=first.data[[8]])
    list.theta[[i]] <- sample_theta(list.theta[[i]],first.data$s,
                                    first.data[[8]],y,first.data$ntheta)
    list.logRV[[i]] <- predict_logRV(list.theta[[i]],first.data[[8]],
                                     first.data$ntheta,upd$B,OLSX[n,],pre_y,
                                     first.data$s,alpha)
    list.beta[[i]]  <- upd$B
    #print (first.data[[8]]) # value of k
    #print (alpha)
  }

  #list.theta
  #list.s
}
```

```
# OLS estimation of beta
B1<-B2<-B3<- rep(0,NN-1)

for (i in 1:NN){
  B1[i] <- list.beta[[i]][1]
  B2[i] <- list.beta[[i]][2]
  B3[i] <- list.beta[[i]][3]
}

mean(B1)
mean(B2)
mean(B3)

Beta<-cbind(B1,B2,B3)

var(Beta)
#check beta
#betahat=(t(X)x)^(-1)* t(X)*y
{
  uu<-list.theta[[NN]][,1]
  vv<-list.theta[[NN]][,2]
  s<-list.s[[NN]]
  u<-v<-rep(0,n)
```

```
  for (i in 1:n){u[i]<-uu[s[i]];v[i]<-vv[s[i]]}
  OLSXx <- cbind(1/v,OLSX[,1]/v,OLSX[,2]/v,OLSX[,3]/v)[22:(n-1),]
  A <- solve(t(OLSXx) %*% OLSXx) %*% t(OLSXx)
  OLSbeta <- solve(t(OLSXx) %*% OLSXx) %*% t(OLSXx)  %*% (((logRV)/v)[23:n])
  Y <- var(((logRV)/v)[23:n])
  variance <- (A * Y) %*% t(A)

  OLSbeta[1]-1.96* sqrt(variance[1,][1])
  OLSbeta[1]+1.96* sqrt(variance[1,][1])
  OLSbeta[2]-1.96* sqrt(variance[2,][2])
  OLSbeta[2]+1.96* sqrt(variance[2,][2])
  OLSbeta[3]-1.96* sqrt(variance[3,][3])
  OLSbeta[3]+1.96* sqrt(variance[3,][3])
  OLSbeta[4]-1.96* sqrt(variance[4,][4])
  OLSbeta[4]+1.96* sqrt(variance[4,][4])
}
```

5.2 R code for Graph

```
# Time Series plot
  Data <- read.table("C:/Users/Tina/Dropbox/Thesis/coding/data.txt")
  #descripetion: date, return (5min) close-to-close 4pm, return from daily data, R
  r <- RETURN<-100*Data[,2]
```

```
  r2 <-r^2
  v <- RV<- Data[,5]
  v0 <- RV2<-RV^{2} #RV square
  logRV <- log(RV)
  v1<-Data[,5] #RV1
  v2<-Data[,6] #RV2
  v3<-Data[,7] #RV3

{
  par(mfrow = c(4, 1))
  plot(RETURN,type="l")
  title("Time Series plot of Daily Realized Volatility")
  plot(r2,type="l",ylim=c(0,60))
  title("Time Series plot of Daily Returns")
  plot(RV,type="l",ylim=c(0,60))
  title("Time Series plot of squared Daily Returns")
  plot(logRV,type="l")
  title("Time Series plot of log Daily Realized Volatility")
}

# trace plot for k and alpha
{
  par(mfrow = c(2, 1))
  plot(array.k[1,],type="l",ylab="k")
```

```
  title("trace plot of k")

  plot(array.alpha[1,],type="l",ylab="alpha")
  title("trace plot of alpha")
}

# trace plot for log RV
{
  plot(list.logRV[[200]],type="l",col="red",ylim=c(0, 0.8),
       xlab="predictive value",ylab="density" )
  for (i in 201:4999){
    lines(list.logRV[[i]],type="l")
  }
  lines(list.logRV[[5000]],type="l",col="green")
}

# acf plot
{
  par(mfrow = c(3, 1))

  acf(r)
  title(sub="ACF of 5-minute return data")

  acf(v1)
```

```
  title(sub="ACF of 5-minute RV1")

  acf(log(v1))
  title(sub="ACF of 5-minute RV1")
}

# plot for 3-dim Dirichlet distribtuion
{
  AA <-rdirichlet(1000, c(100,100,100))
  plot3d(AA,xlab=NULL,ylab=NULL,zlab=NULL,box=0)

  BB <-rdirichlet(1000, c(1,1,1))
  plot3d(BB,xlab=NULL,ylab=NULL,zlab=NULL,box=0)

  CC <-rdirichlet(1000, c(0.05,0.05,0.05))
  plot3d(CC,xlab=NULL,ylab=NULL,zlab=NULL,box=0)

  DD <-rdirichlet(1000, c(0.05,1,100))
  plot3d(DD,xlab=NULL,ylab=NULL,zlab=NULL,box=0)

  EE <-rdirichlet(1000, c(1,100,100))
  plot3d(EE,xlab=NULL,ylab=NULL,zlab=NULL,box=0)

  FF <-rdirichlet(1000, c(1,1,0.05))
```

```
  plot3d(FF,xlab=NULL,ylab=NULL,zlab=NULL,box=0)
}

# plot of  kernel density
{
  x <- density(logRV,kernel="gaussian",na.rm = FALSE)
  plot(x,type="l",xlim=c(-10,10))
  title("classical kernel density estimation")
}

# graph of predictive value
{
  nn<-length(pre_y)
  pred_y<-rep(0,nn)
  burn_in<-trunc(NN*0.2)

  for (i in burn_in:NN){
    pred_y  <- pred_y + list.logRV[[i]]
    print(head(pred_y))
  }


  pred_y <- pred_y/(NN - burn_in +1)
```

```
  pre_y  <- pre_y[2:nn]
  pred_y <- pred_y[2:nn]

  plot(pre_y,pred_y,type = "l",col="black",
       xlab="predictive value",ylab="density")
  lines(pre_y,dnorm(pre_y,mean(y),sqrt(var(y))),col = "red")
  title("predictive density with 5000 iterations")
  legend("topright",legend=c("predictive","historical"),
         col=c("black","red"),lty=1,lwd=1,cex=1)

  plot(pre_y,log(pred_y),type = "l",col="black",
       xlab="predictive value",ylab="density")
  lines(pre_y,log(dnorm(pre_y,mean(y),sqrt(var(y)))),col = "red")
  title("log of predictive density with 5000 iterations")
  legend("topright",legend=c("predictive","historical"),
         col=c("black","red"),lty=1,lwd=1,cex=1)
}

#graph for break sticking process
{
alpha <- 0.1
pi<-p<-rbeta(10,1,alpha)
for (i in 2:9){
```

```
  pi[i]<-p[i]*pi[i-1]*(1-p[i-1])/p[i-1]
}

pi[10]<-1-sum(pi[1:9])

par(mfrow=c(3,1))
plot(pi)

alpha <- 1
pi<-p<-rbeta(10,1,alpha)
for (i in 2:9){
  pi[i]<-p[i]*pi[i-1]*(1-p[i-1])/p[i-1]
}

pi[10]<-1-sum(pi[1:9])
plot(pi)

alpha <- 10
pi<-p<-rbeta(10,1,alpha)
for (i in 2:9){
  pi[i]<-p[i]*pi[i-1]*(1-p[i-1])/p[i-1]
}

pi[10]<-1-sum(pi[1:9])
```

```
plot(pi)


}
```

Bibliography

Andersen, T. G. and Benzoni, L. (2008). Realized volatility. Working Paper Series WP-08-14, Federal Reserve Bank of Chicago.

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Labys, P. (2003). Modeling and forecasting realized volatility. *Econometrica*, **71**, 579625.

Andersen, T. G., Bollerslev, T., and Diebold, F. X. (2007). Roughing it up: Including jump components in the measurement, modeling and forecasting of return volatility. *The Review of Economics and Statistics*, **89**(4), 701–720.

Back, K. (1991). Asset pricing for general processes. *Journal of Mathematical Economics*, **20**, 371–395.

Barndorff-Nielsen, O. E. and Shephard, N. (2002). Econometric analysis of realised volatility and its use in estimating stochastic volatility models. *Journal of the Royal Statistical Society*, **63**, 253280.

Barndorff-Nielsen, O. E. and Shephard, N. (2004). Econometric analysis of realized covariation: High frequency based covariance, regression, and correlation in financial economics. *Econometrica*, **72**(3), 885–925.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, **31**(3), 307 – 327.

Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, **7**(2), 174–196.

De Finetti, B. (1931). Funzione caratteristica di un fenomeno aleatorio.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, **50**(4), 987–1007.

Escobar, M. D. (1988). *Estimating the means of several normal populations by nonparametric estimation of the distribution of the means*. Ph.D. thesis, Department of Statistics, Yale University, New Haven.

Escobar, M. D. and West, M. (1994). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, **90**, 577–588.

Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, **Vol. 1,**, 209–230.

Ferguson, T. S. (1983). Bayesian density estimation by mixtures of normal distributions. *Recent advances in statistics*, **24**, 287–302.

Hansen, P. R. and Lunde, A. (2006). Realized variance and market microstructure noise. *Journal of Business & Economic Statistics*, **24**(2), 127–161.

Ishwaran, H. and James, L. F. (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, **96**, 453.

Maheu, J. M. and McCurdy, T. H. (2011). Do high-frequency measures of volatility improve forecasts of return distributions? *Journal of Econometrics*, **160**(1), 69–76.

Mike West, P. M. and Escobar, M. D. (1994). *Hierarchical Priors and Mixture Models, with Applications in Regression and Density Estimation*, pages 363–386. New York: John Wiley.

Ole E. Barndorff-Nielsen, Peter Reinhard Hansen, A. L. N. S. (2008). Designing realised kernels to measure the ex-post variation of equity prices in the presence of noise. *Econometrica*, **76 No. 6**, 14811536.

Peter Whittle, H. W. (1951). *Hypothesis Testing in Time Series Analysis.* Ph.D. thesis, University of Uppsala.

Protter, P. E. (1990). *Stochastic Integration and Differential Equation: a new approach*. New York: springer.

Russell, J. R. and Bandi, F. M. (2004). Microstructure noise, realized volatility, and optimal sampling. Econometric Society 2004 Latin American Meetings 220, Econometric Society.

Sethuraman, J. (1991). A constructive definition of dirichlet priors. Technical report, DTIC Document.

Zhang, L., Mykland, P. A., and Ait-Sahalia, Y. (2003). A tale of two time scales: Determining integrated volatility with noisy high frequency data. Working Paper 10111, National Bureau of Economic Research.