Bayesian Sampling for Smoothing Parameter Estimation

A thesis submitted for the degree of

Doctor of Philosophy

by

Shuowen Hu

M. Applied Econometrics (Monash), B. Comm. & B. IT. (ANU)

Department of Econometrics and Business Statistics

Monash University

Australia

October 2015

Contents

Copyright notice

Abstract

Kernel density estimation is one of the most important techniques for understanding the distributional properties of data. It is understood that the effectiveness of such approach depends on the choice of a kernel function and the choice of a smoothing parameter (bandwidth). This thesis has undertaken some important topics in bandwidth selection for kernel density estimation for data that behave in various nature. The first issue evolves around selecting appropriate bandwidth given the characteristics of the local data in multivariate setting. In Chapter 3, the study proposes a kernel density estimator with tail-adaptive bandwidths. The study derives posterior of bandwidth parameters based on the Kullback-Leibler information and presented an MCMC sampling algorithm to estimate bandwidths. The Monte Carlo simulation study shows that the kernel density estimator with tail-adaptive bandwidths estimated through the proposed sampling algorithm outperforms its competitor. The tail-adaptive kernel density estimator is applied to the estimation of bivariate density of the paired daily returns of the Australian Ordinary index and S&P 500 index during the period of global financial crisis. The results show that this estimator could capture richer dynamics in the tail area than the density estimator with a global bandwidth estimated through the normal reference rule and a Bayesian sampling algorithm.

The second research project investigates bandwidth selection for multimodal distributions or data that exhibits clustering behaviours. Chapter 4 proposes a cluster-adaptive bandwidth kernel density estimator for data with multimodality. This method employs a clustering algorithm to assign a different bandwidth to each cluster identified in the data set. The study derives a posterior of bandwidth parameters based on the Kullback-Leibler information and presented an MCMC sampling algorithm to estimate bandwidths. The Monte Carlo simulation study shows that when the underlying density is a mixture of normals, the kernel density estimator with cluster-adaptive bandwidths estimated through the proposed sampling algorithm outperforms its competitor. When the underlying densities are fat-tailed, the combined approach of tail- and cluster-adaptive density estimator performs the best. In an empirical study, bandwidth matrices are estimated for the cluster-adaptive kernel density estimator of eruption duration and waiting time to the next eruption collected from Old Faithful greyer, which is often analysed due to its clustering nature. The results again shows clear advantage of the proposed cluster-adaptive kernel density estimator over traditional approaches.

The third topic extends the Bayesian bandwidth selection method to volatility models of financial asset return series. The study is motivated by the fact that only limited attention in the literature has been invested on the estimation of nonparametric nonlinear type of volatility models through a Bayesian approach. Chapter 5 presents a new volatility model called the semiparametic nonlinear volatility (SNV) model. Based on financial return series of major stock indices in the world, the performance of the proposed volatility model against the competing models are examined in both in-sample

and out-of-sample periods. The proposed model and the Bayesian estimation method show strong and convincing performance results. The study also evaluates the empirical value-at-risk (VaR) performance of the competing models. The proposed volatility model shows the best performance in most cases.

Acknowledgement

I would like to extend my sincere gratitude to my PhD supervisors Professor Don Poskitt and Associate Professor Xibin Zhang. This research degree has truly been a journey to the unknown, and the experience has changed me to be more independent, brave and complete. Don has been the light house of this journey and I would like to thank you for your strong leadership along the way. Xibin has always been my role model during these years. I am very grateful to you for your patience and guidance during the good times and difficult times. I deeply appreciate your understanding and support to many hard decisions I have made.

I would also like to thank my family. To my parents, I feel deeply indebted to you for the sacrifices you have made to support my decision to undertake this degree. To my wife Yuanyuan, I thank you for your love and support during this journey.

Declaration

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma in any university or equivalent institution, and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Shuowen Hu

Preface

The material in Chapter 3 has been published in *Computational Statistics and Data Analysis*, Volume 56, Issue 3, 2012, page 732-740.

The contribution in Chapter 3 of this thesis was presented in the Eighth ICSA International Conference held in Guangzhou, China in 2010.

The material in Chapter 4 and 5 is being prepared for submission to peer reviewed academic journals.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and aim of the thesis

Kernel density estimation is one of the most important techniques for understanding the distributional properties of data. It is understood that the effectiveness of such an approach depends on the choice of a kernel function and the choice of a smoothing parameter (as known as bandwidth).[1] Although these two issues cannot be treated independently, it is widely noted that the performance of kernel density estimation is mainly affected by the choice of bandwidth, and only in an minor way, by the choice of kernel (for example, Scott 1992, Wand & Jones 1995).[2]

The traditional kernel density estimation method aims to obtain a bandwidth to minimise a pre-specified distance function between the true density function and its density estimator. The most well-known method in this category is the least squared

[1] See Izenman (1991) for a discussion

[2] However, Marron & Nolan (1988), Vieu (1999), Horová et al. (2002) among others, have examined the kernel function choice.

cross-validation to choose the bandwidth by minimising the integrated squared error (Rudemo 1982, Bowman 1984). However, such an approach is difficult at best because the true density function is unknown.

An alternative method is the likelihood cross-validation which minimises the Kullback-Leibler information of two densities $f(x)$ and $\hat{f}$ the density estimator introduced by Duin (1976). As shown by Zhang, King & Hyndman (2006), the optimisation of the Kullback-Leibler function does not depend on the true density function.

While conventional studies have focused on estimating a fixed bandwidth (or global-bandwidth) for all data of a sample (see Jones et al. 1996, for a survey), other investigations such as Terrell & Scott (1992) and Sain & Scott (1996) have promoted the idea of data-driven adaptive bandwidths in density estimation. It allows the bandwidth to vary at different data points. The problem has been extensively investigated in univariate settings. However, less attention appears to have been paid to data-driven methods for adaptive bandwidth estimation for multivariate data. One of the main difficulty is the curse of dimensionality, in which the number of bandwidth parameters increases dramatically with the dimension of the data. In addition, the existing literature does not provide an efficient computation algorithm that is flexible enough to choose bandwidths when dealing with multivariate data.

The first two research topics are motivated by such problems and aim to provide a multivariate data-driven bandwidth estimation method that allows for certain degree of flexibility for the bandwidth to be dependant on the observations in the sample.

The first research topic aims to address the necessity for appropriate smoothness in low density regions of the underlying density function by proposing the tail-adaptive kernel density estimator. Empirically, most financial analysts believe U.S. stock market has a leading effect on most other stock markets world wide. Using a kernel density estimator of bivariate stock-index returns, it is possible to derive the conditional distribution of the return in one market for a given return in the U.S. market, and thereby better understand how the two stock markets are correlated. As the marginal densities of individual stock-index returns often exhibit leptokurtosis and heavy tails, the kernel density estimation of the bivariate density of stock-index returns may require different bandwidths to be assigned to the observed returns in different density regions. To estimate such bandwidths, we adopt the MCMC algorithm proposed by Zhang et al. (2006), where bandwidths are treated as parameters.

In the second topic, the tail-adaptive kernel density estimator is extended and used to estimate multi-modal densities, where each mode may have different dispersion. The concept of multi-modality is closely related to that of "cluster" in clustering analysis. As defined by Hartigan (1975, p205), "Clusters may be thought of as regions of high density separated from other such regions by regions of low density". The basic idea of density-based clustering is to identify the association of observations and the corresponding empirical modes by finding the connected components in the level set. This study proposes a cluster-adaptive kernel density estimator that address the multi-modality issue. In order to find the modes in the observed data, we employ one of the density-based clustering algorithms called CRA proposed by Cuevas et al. (2000, 2001) to automatically search the clusters in the data set, where multi-modality exists.

This means we are able to leverage the clustering information obtained from the CRA algorithm to automatically allocate different bandwidth matrices to each cluster in the data set, where the bandwidths parameters are estimated through a proposed MCMC algorithm. In an empirical application, we applied the cluster-adaptive kernel density estimator to the Old Faithful greyer data. The results showed a clear advantage of the proposed cluster-adaptive kernel density estimator over traditional approaches.

Further extension of the Bayesian kernel density bandwidth estimation technique to the estimation of financial volatility is carried out in the third topic. Empirical evidence has shown that the volatility of financial asset returns is often highly persistent and asymmetrically distributed. The existing literature has focused on parametric estimation of ARCH (Engle 1982) and GARCH (Bollerslev 1986) models, as well as their extensions. A nonparametric estimation method for volatility models usually aims at addressing the strong parametric assumption of ARCH and GARCH models, such as the linearity assumption in the volatility equation (e.g. Pagan & Schwert 1990) and the distribution function of error term (e.g. Engle & Gonzalez-Rivera 1991).

Recently, a nonlinear nonstationary heteroscedastic (NNH) model was proposed (Park 2002) as an alternative class of volatility models. The NNH model assumes the conditional variance as a known parametric nonlinear function of a persistent explanatory variable. Han & Park (2008) extended the NNH model by allowing the ARCH(1) component in the model, while Han & Zhang (2012) proposed a nonparametric version of the NNH model called the nonstationary nonparametric volatility (NNV) model.

The NNV model assumes that the nonlinear function of the regressor the model is unknown and employs Nadaraya-Watson estimator. It is well known that the Nadaraya-Watson estimator's performance depends on its bandwidth. However, the bandwidth selection issue was not discussed in details by Han & Zhang (2012).

To fill in this gap, we propose a new volatility model, which combines the ARCH(1) model and the NNV model, and aim to develop a Bayesian sampling algorithm to estimate bandwidth. In addition, our Bayesian technique allows the error term to follow an unknown distribution, which we estimate through the kernel method.

The empirical performance of the proposed SNV model and the Bayesian estimation method are evaluated and compared against the NNV models and ARCH-NNH models through alternative bandwidth selection methods. Based on financial return data of eight major global stock markets, both in-sample and out-of-sample performance are examined. Through the calculation of the loss functions given by Patton (2011), the Bayesian method shows strong performance results in in-sample period and even stronger result in out-of-sample period.

In addition, the empirical VaR performance of the competing models are examined. The proposed SNV model with standard Gaussian density showed the best performance in most cases. The empirical performance of the proposed SNV model is highly competitively comparing to the existing NNV and ARCH-NNH models.

1.2 Outline of the thesis

The structure of the rest of the thesis is as follows. In Chapter 2, an extensive literature review is provided. It identifies the motivations for the topics presented in this thesis. In Chapter 3, we propose the tail-adaptive kernel density estimator and examine its performance via Monte Carlo simulation studies. We investigates the dependence of daily returns of the Australia stock market on the U.S. market. In Chapter 4, we propose to extend the tail-adaptive kernel density estimator by allowing it to be adaptive across clusters. The performance of this density estimator is investigated through Monte Carlo and empirical studies. In Chapter 5, we propose an alternative semi-parametric volatility model, which is estimated through Bayesian sampling. We show that the proposed model offers more flexibility and is highly competitive against its competitors. The conclusion of this thesis is presented in Chapter 6.

Chapter 2

Literature Review

2.1 Introduction

This chapter provides a brief introduction of the Markov chain Monte Carlo (MCMC) simulation technique, which is employed in the subsequent studies of the thesis. We present a survey of the literature on kernel density estimation method and its related applications.

Through an extensive survey of the previous studies in kernel density estimation and related applications, several obvious gaps have been identified and require further investigation. It is found that the bandwidth plays an important role in determining the overall performance of the kernel density estimator. Studies have suggested that the bandwidth parameter be dependent on the sample points. For example, a large bandwidth is required in the area that need high degree of smoothing (and vice versa). This problem has been addressed in the univariate case. However, extension to higher-dimension is not straightforward mainly due to computational difficulties

when dealing with a large number of parameters. We therefore wish to remedy this issue via a Bayesian sampling approach.

In addition, it is found that application of a kernel-based semiparametric and nonparametric methods on financial volatility estimation is becoming more popular with promising empirical performance over traditional parametric methods. It motivates us to propose an alternative Bayesian model to estimate financial volatility models with an unknown error distribution.

The following of this chapter is organised as follows. The next section provides review of literature on multivariate kernel density estimation and discusses the motivation of an adaptive density estimator. Section 2.3 discusses the possibility of combining density-based clustering technique and adaptive density estimation. In Section 2.4, we present a review of the literature on semiparametric and nonparametric financial volatility models and discuss the motivation for a new and general model.

2.2 Markov Chain Monte Carlo simulation

Let $f(x)$ denote a density function and $\hat{f}$ its estimator. The MCMC is a well known technique for solving problems involving high dimensional integrations. As described by Robert & Casella (1999) that the "MCMC method for the simulation of a distribution f is any method producing an ergodic Markov chain θ_t whose stationary distribution is f".

Let θ be the parameter vector to be estimated. According to Bayes theorem, the posterior density of θ is

$$\pi(\theta|y) = \frac{p(\theta)p(y|\theta)}{\int p(\theta)p(y|\theta)}, \tag{2.2.1}$$

where $\pi(\theta)$ is the prior of θ, $p(y|\theta)$ is the likelihood of y given θ, and $\int p(\theta)p(y|\theta)$ is a normalising constant.

Bayesian inference focuses on the features of the posterior density, such as moments and quantiles, which can be expressed in terms of expected value of functions θ under the posterior density of θ. For example, the expectation of $k(\theta)$ is

$$E_{\pi(\theta|y)}[k(\theta|y)] = \int k(\theta)\pi(\theta|y)d\theta. \tag{2.2.2}$$

However, the analytical expression of $E_{\pi(\theta|y)}[k(\theta|y)]$ is often intractable because the normalising constant is often unknown.

To fix this problem, we could simulate from the posterior and approximate (2.2.2) through Monte Carlo simulation. This involves drawing a sequence of independent random numbers, θ^i, for $i = 1, 2, \cdots, N$, from $\pi(\theta|y)$ and estimating (2.2.2) by

$$E_{\pi(\theta|y)}[k(\theta|y)] \approx \frac{1}{N}\sum_{i=1}^{N} k\left(\theta^i\right). \tag{2.2.3}$$

Based on the law of large numbers, the above approximation is able to produce an accurate result by increasing the sample size N. However, the posterior is often non-standard. This makes it difficult to draw independent random numbers.

Tierney (1994) presented that a reversible Markov chain has a unique stationary density $\pi(x)$ with the following properties that form the theoretical fundamentals of the ergodic theorem for MCMC simulations:

1) Irreducibility, which states that x can reach any point in $x^{(i)}$, for $i = 1, 2, \ldots, M$, with a positive probability in finite time, regardless of the starting value.

2) Aperiodicity, which prevents the Markov chain going through different set of states.

Gilks et al. (1996) showed that it is not necessary for the posterior sample to be independent to produce reliable approximation of (2.2.2) if $\pi(\theta|y)$ is the stationary transition density for the Markov chain sampling process.

2.2.1 Metropolis-Hastings algorithm

In the application of MCMC, an important question is how to draw random samples from the posterior density, which is often complex and non-standard. In order to solve this problem, an algorithm was introduced by Metropolis, Rosenbluth, Rosenbluth, Teller, Teller et al. (1953) and generalized by Hastings (1970). This algorithm is called the Metropolis-Hasting algorithm and aims to integrate complicated functions by generating random numbers.

Let $q(\theta|\theta^{(i)})$ denote a proposal density and $\theta^{(i)}$ as current state. To draw a sample of random variables θ, the proposal density must be specified as a suitable density, e.g. the Normal density. A candidate $\tilde{\theta}$ is sampled from $q(\theta|\theta^{(i)})$ and accepted as the new

state $\theta^{(i+1)}$ with an acceptance probability $\alpha(\theta^{(i)}, \tilde{\theta})$ given by

$$\begin{aligned} \alpha(\theta^{(i)}, \tilde{\theta}) &= \min\left\{\frac{\pi(\tilde{\theta}|y)q(\theta^{(i)}|\tilde{\theta})}{\pi(\theta^{(i)}|y)q(\tilde{\theta}|\theta^{(i)})}, 1\right\}, \qquad \text{if } \pi(\theta^{(i)}|y)q(\tilde{\theta}|\theta^{(i)}) > 0, \\ &= 1, \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise,} \end{aligned} \tag{2.2.4}$$

where $\pi(\theta|y)$ is the posterior of θ given y.

It is shown that $\alpha(\theta^{(i)}, \tilde{\theta})$ does not depend on the normalising constant of $\pi(\cdot)$ because it appears in both numerator and denominator. The acceptance probability is calculated and compared to a random number u drawn from a uniform density $U(0,1)$. If $\alpha(\theta^{(i)}, \tilde{\theta}) > u$, $\tilde{\theta}$ is accepted as the new state $\theta^{(i+1)} = \tilde{\theta}$; otherwise, the new state is set to the current state $\theta^{(i+1)} = \theta^{(i)}$. If the proposal density is symmetric, it leads to $q(\theta_1|\theta_2) = q(\theta_2|\theta_1)$, and the acceptance probability reduces to $\pi(\tilde{\theta}|y)/\pi(\theta^{(i)}|y)$.

During each iteration, a candidate is drawn through a random-walk process:

$$\tilde{\theta} = \theta^{(i)} + \tau\epsilon,$$

where ϵ is the standard normal and independent of $\theta^{(i)}$, and τ is a tuning parameter. This algorithm is the random-walk Metropolis algorithm because the proposal density is symmetric. The acceptance probability is

$$\alpha(\theta^{(i)}, \tilde{\theta}) = \min\left\{\frac{\pi(\tilde{\theta}|y)}{\pi(\theta^{(i)}|y)}, 1\right\}.$$

The random-walk Metropolis algorithm is often used due to its simplicity.

In addition to the random-walk Metropolis algorithm, the independent Metropolis-Hastings algorithm requires the candidate to be independent of the current state. It

implies that the proposal density meets the condition that $q(\tilde{\theta}|\theta^{(i)}) = q(\tilde{\theta})$. The acceptance probability of the independent Metropolis-Hastings algorithm is

$$\alpha(\theta^{(i)}, \tilde{\theta}) = \min\left\{\frac{\pi(\tilde{\theta}|y)q(\theta^{(i)})}{\pi(\theta^{(i)}|y)q(\tilde{\theta})}, 1\right\}.$$

Another specification of the Metropolis-Hastings algorithm is the Gibbs sampler, which makes use of the full conditional posterior of each variable in MCMC simulations. If the conditional posterior has a closed form of a known distribution, the candidate can be directly sampled from the conditional posterior and is always accepted as a new state. However, in some cases, the full conditional posterior of latent variables is non-standard and makes the Gibbs sampler infeasible. The Metropolis-Hastings algorithm can be applied to draw samples in this circumstance. More generally, one can use both algorithms to sample the variables of interest. A specific example is the Bayesian estimation of stochastic volatility models whose sampling procedure is a hybrid procedure and involves sampling latent volatilities by using a Metropolis-Hastings algorithm and sampling parameters by using a Gibbs sampler (see for example, Kim et al. 1998).

2.2.2 Convergence of MCMC algorithm

In practice, it is important to ensure the MCMC algorithm achieves reasonable convergence because a simulated chain should converge geometrically to the stationary posterior density $\pi(\cdot)$ according to the ergodic theorem. Let $\{\theta^{(i)}\}$, for $i = 1, 2, \ldots, N$, denote a Markov chain which is sampled from the conditional posterior $\pi(\theta|y)$. This

chain can be summarized in terms of ergodic averages:

$$\bar{f}_N(\hat{\theta}) = \frac{1}{N}\sum_{i=1}^{N} f(\theta^{(i)}),$$

where $f(\theta)$ is a real function of θ. Roberts (1996) showed that the central limit theorem of the ergodic average should hold under the condition that

$$\sqrt{N}\left\{\bar{f}_N(\hat{\theta}) - E_\pi\left(f(\theta^{(i)})\right)\right\} \longrightarrow N(0,\sigma_f^2),$$

where convergence is in distribution as $N \longrightarrow \infty$ and σ_f is a positive value.

In order to check the mixing performance of simulated chains, Geyer (1992) and Roberts (1996) suggested estimating σ_f^2 by using a batch mean method, which divide the sequence $\left\{\theta^{(i)}\right\}$, for $i = 1,2,\ldots,N$ into m, batches and each batch contains n iterations in the sense that $N = m \times n$. The mean of kth batch $\bar{\theta}_k$ is computed as

$$\bar{\theta}_k = \frac{1}{n}\sum_{j=(k-1)n+1}^{kn} f(\theta^{(i)}),$$

for $j = 1,2,\ldots,n$, where n should be sufficiently large so that $\left\{\bar{\theta}_k\right\}$, for $k = 1,2,\ldots,m$, are independent and identically distributed as $N(E_\pi[f(\theta)],\sigma_f^2)/n$. The batch mean estimate of σ_f^2 is computed as

$$\hat{\sigma}_f^2 = \frac{n}{m-1}\sum_{k=1}^{m}\left(\bar{\theta}_k - \bar{f}_N(\hat{\theta})\right)^2.$$

Therefore, we can estimate the standard error of $\bar{f}_N$ by $\sqrt{\hat{\sigma}_f^2/N}$ which is commonly used to check for the convergence of a simulated Markov chain. Based on $\hat{\sigma}_f^2$, Kim,

Shepherd & Chib (1998) presented a simulation inefficiency factor (SIF) to measure the convergence performance. The SIF is a ratio of the variance of the sampled mean to the variance of the sample mean from a hypothetical sampler that draws independent random observations from the posterior distribution:

$$\text{SIF} = \frac{\hat{\sigma}_f^2}{\tilde{\sigma}_f^2}, \tag{2.2.5}$$

where $\bar{\sigma}_f^2$ is computed as

$$\tilde{\sigma}_f^2 = \left\{ \frac{1}{N-1} \sum_{i=1}^{N} [f(\theta^{(i)}) - \bar{f}_N(\hat{\theta})]^2 \right\}.$$

Theoretically, the smaller the SIF is, the better convergence the sampler achieves. Note that none of the available methods can guarantee that a sampler achieves convergence (Tsay 2005). Therefore, one need to plot the sampled path, its autocorrelation function (ACF) and histogram to visually confirm that the simulated chain has achieved reasonable convergence. In general, the convergence of a sampler is independent of the starting value. However, it may take a long time for the sampler to achieve the convergence if the starting value is too far away from the true value. This requires us to use a burn-in period which is discarded in order to reduce the effect of starting value on the simulated chains. In the current literature, there exists no method to determine the length of burn-in period because the convergence may vary in different algorithms and also is subject to different data. As suggested by Brooks (1998), a practical suggestion is to keep on track of the convergence diagnostics of the sampler and then decide how long the iterations should be used in simulations.

2.3 Multivariate kernel density estimation

Let $\boldsymbol{x} = (x_1, x_2, \cdots, x_d)'$ denote a random vector with density $f(\boldsymbol{x})$ defined in $\mathbb{R}^d$, and let $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$ represent a sample of independent random vectors drawn from $f(\boldsymbol{x})$. The general multivariate kernel density estimator is given by

$$\hat{f}(\boldsymbol{x}) = \frac{1}{n|H|^{1/2}} \sum_{i=1}^{n} K\left(H^{-1/2}(\boldsymbol{x} - \boldsymbol{x}_i)\right) = \frac{1}{n} \sum_{i=1}^{n} K_H(\boldsymbol{x} - \boldsymbol{x}_i) \tag{2.3.1}$$

where $K_H(\boldsymbol{x}) = |H|^{-1/2} K(H^{-1/2}\boldsymbol{x})$ with $K(\cdot)$ is a multivariate kernel function and H is a symmetric and positive definite matrix which is called the bandwidth matrix.

Three important choices have to be made when using a multivariate density estimator (Wand & Jones 1995, p94). They are, the form of a kernel function $K(\cdot)$, the choice of smoothing parameterisation and the choice of the bandwidth matrix H.

The kernel function is often chosen to be a symmetric multivariate density function, and a very popular choice is the multivariate standard d-variate normal density

$$K(\boldsymbol{x}) = (2\pi)^{-d/2} \exp\left\{-\frac{1}{2}\boldsymbol{x}'\boldsymbol{x}\right\}, \tag{2.3.2}$$

and it is chosen to be the kernel function in our case.

The choice of smoothing parameterisation generally refers to the type of bandwidth matrix H to be used for a multivariate kernel density estimation. That is whether one should choose a diagonal or full bandwidth matrix. A full bandwidth matrix provides more flexibility as it allows for different degrees of smoothing in each dimension and

the possible correlation between each pair of dimensions. A diagonal bandwidth matrix is less flexible as it does not allow for correlation between dimensions. However, the full bandwidth matrix introduces more parameters to be chosen. In fact, a full symmetric matrix has $d(d+1)/2$ different parameters, while a diagonal matrix has only d parameters. It is very obvious that the additional number of parameters to be chosen increases quickly as the dimension increases. As a consequence, more data is needed and the computation becomes more intensive. Therefore, the advantage of using a full bandwidth matrix can quickly disappear as the number of dimensions increases.

The last choice is to select the bandwidth matrix with respect to certain criterion. This is of crucial importance because the bandwidth matrix controls the amount and direction of multivariate smoothing. It is generally accepted that the performance of a kernel density estimator depends mainly on the choice of bandwidth, and only a minor way of by the choice of the kernel function (Izenman 1991, Scott 1992, Wand & Jones 1995, Simonoff 1996). Therefore, several criterions are employed in the literature to evaluate the performance of the bandwidth selection.

2.3.1 Least square cross-validation

The basic idea of least square cross-validation (LSCV) is to choose a bandwidth by minimising the integrated squared error (ISE) (Rudemo 1982, Bowman 1984). The ISE of a kernel density estimator is given as (Scott 1992, Wand & Jones 1995)

$$\mathrm{ISE}\{\hat{f}_H(\boldsymbol{x})\} = \int \left[\hat{f}_H(\boldsymbol{x}) - f(\boldsymbol{x})\right]^2 d\boldsymbol{x}. \tag{2.3.3}$$

The ISE is a random variable that depends on the true unknown density function, the type of estimator and the sample size (Scott 1992). In most cases, it is sufficient to examine the average of the ISE, which is called the mean integrated squared error (MISE)

$$\text{MISE}\{\hat{f}_H(\boldsymbol{x})\} = E\left\{\int \left[\hat{f}_H(\boldsymbol{x}) - f(\boldsymbol{x})\right]^2 d\boldsymbol{x}\right\}. \tag{2.3.4}$$

In univariate case, Wand & Jones (1995) presented the specification for the asymptotic MISE as

$$\text{AMISE}\left\{\hat{f}_H(\boldsymbol{x})\right\} = \frac{1}{nh}R(K) + \frac{1}{4}h^4\mu_2(K)^2R(f''), \tag{2.3.5}$$

where $\mu_2 = \int x^2 f(x)dx$ and $R(f'') = \int f''(x)^2 dx$. Under certain smoothness conditions on the density f (Wand & Jones 1995), the AMISE of a multivariate kernel density estimator is given as

$$\text{AMISE}\{\hat{f}_H(\boldsymbol{x})\} = \frac{1}{n}|H|^{-1/2}\int K(\boldsymbol{x})^2 d\boldsymbol{x} + \frac{1}{4}\mu_2(K)^2\int \text{tr}^2\left\{H\nabla^2 f(\boldsymbol{x})\right\} d\boldsymbol{x}, \tag{2.3.6}$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix and $\nabla^2 f(\boldsymbol{x})$ denotes the Hessian matrix of $f(\boldsymbol{x})$. The optimal bandwidth with respect to the AMISE criterion is defined as

$$\widehat{H}_{\text{AMISE}} = \arg\min_H \text{AMISE}\left\{\hat{f}_H(\boldsymbol{x})\right\}. \tag{2.3.7}$$

When data are observed from multivariate normal density with no correlation and the diagonal bandwidth matrix $H = \text{diagonal}(h_1, h_2, \cdots, h_d)$ is used, Scott (1992) showed

the bandwidth selector that minimises AMISE as

$$h_i = \sigma_i \left\{ \frac{4}{(d+2)n} \right\}^{1/(d+4)}, \tag{2.3.8}$$

for $i = 1, 2, \cdots, d$, and σ_i is the standard deviation of dimension i and can be replaced by the sample standard deviation in practice. This bandwidth selector is called the Normal Reference Rule by Scott (1992). Although in most interesting cases the data are non-normal and variables are correlated, the Normal Reference Rule is widely used in the literature due to its practicality.

The least squares cross-validation (LSCV) is first used by Rudemo (1982) and Bowman (1984) in univariate cases, but can be extended to multivariate cases (Sain, Baggerly & Scott 1994). It is based on the idea of expanding the MISE of $\hat{f}_H(\boldsymbol{x})$ (Wand & Jones 1995) to be further expressed as

$$\text{MISE}\left\{\hat{f}_H(\boldsymbol{x})\right\} = E\int \hat{f}_H(\boldsymbol{x})^2 d\boldsymbol{x} - 2E\int \hat{f}_{\boldsymbol{H}}(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} + \int f(\boldsymbol{x})^2 d\boldsymbol{x}. \tag{2.3.9}$$

As the last term does not depend on H, minimising the $\text{MISE}\left\{\hat{f}_H(\boldsymbol{x})\right\}$ is the same as minimising

$$\text{MISE}\left\{\hat{f}_H(\boldsymbol{x})\right\} - \int f(\boldsymbol{x})^2 d\boldsymbol{x} = E\left[\int \hat{f}_H(\boldsymbol{x})^2 d\boldsymbol{x} - 2\int \hat{f}_H(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}\right], \tag{2.3.10}$$

and an unbiased estimator of it is

$$\text{LSCV(H)} = \int \hat{f}_H(\boldsymbol{x})^2 d\boldsymbol{x} - \frac{2}{n}\sum_{i=1}^{n} \hat{f}_{H,i}(\boldsymbol{x}_i), \tag{2.3.11}$$

where

$$\hat{f}_{H,i}(\boldsymbol{x}) = \frac{1}{n-1}\sum_{j\neq i}^{n} K_H(\boldsymbol{x}_i - \boldsymbol{x}_j), \tag{2.3.12}$$

is called the leave-one-out density estimator based on the sample without $\boldsymbol{x}_i$. The technique to choose H that minimises LSCV(H) leads to the LSCV bandwidth selector

$$\widehat{H}_{\text{LSCV}} = \arg\min_{H} \text{LSCV}(H)$$

, which is also called the unbiased cross-validation (UCV). However, several studies have shown that this bandwidth selector has a slow rate of convergence and highly variable (Nolan & Polland 1987, Park & Marron 1990).

To improve the performance of $\hat{H}_{\text{LSCV}}$, Scott & Terrell (1987) proposed the biased cross-validation (BCV) method. The objective function of BCV is obtained by using an estimator to approximate the unknown $R(f'')$ in the univariate AMISE. However, both UCV and BCV have been known that sometimes they have more than one local minimum (Hall & Marron 1982, Scott 1992). Therefore, one must be careful when applying these methods in practice. The UCV and BCV estimators were extended to multivariate settings using a diagonal bandwidth matrix in Sain, Baggerly & Scott (1994). Their simulation study showed that the BCV method generally performs better than the UCV method.

The UCV estimator is also being referred to as a type of plug-in estimator because it employs an estimator to approximate the unknown quantity in the AMISE criterion.

Duong & Hazelton (2003) showed a bivariate example of the plug-in bandwidth selector as an alternative to (2.3.6)

$$\text{AMISE}\left\{\hat{f}_H(\boldsymbol{x})\right\} = \frac{1}{n}|H|^{-1/2}\int_{\mathbb{R}^2} K^2(\boldsymbol{x})d\boldsymbol{x} + \frac{1}{4}\int_{\mathbb{R}^2} \boldsymbol{x}\boldsymbol{x}'K^2(\boldsymbol{x})d\boldsymbol{x}(\text{vech}'H)\boldsymbol{\Psi}_4(\text{vech}H), \tag{2.3.13}$$

where vech is the vector half operator, and $\boldsymbol{\Psi}_4$ is a 3×3 matrix given by

$$\boldsymbol{\Psi}_4 = \int_{\mathbb{R}^2} \text{vech}\left\{2\nabla^2 f(\boldsymbol{x}) - \text{diag}(\nabla^2 f(\boldsymbol{x}))\right\}\text{vech}'\left\{2\nabla^2 f(\boldsymbol{x}) - \text{diag}(\nabla^2 f(\boldsymbol{x}))\right\}d\boldsymbol{x}, \tag{2.3.14}$$

and diag(A) is the matrix A with its non-diagonal elements being zero. The method requires an estimate of $\boldsymbol{\Psi}_4$ to be plugged-in in order to produce $\hat{H}_{\text{AMISE}}$ which aims to minimise the AMISE criterion function.

The plug-in bandwidth selector proposed by Wand & Jones (1994) sometime fails to produce finite bandwidths for full bandwidth matrix, and the technique is considered to be immature, as noted by Duong & Hazelton (2003) and Sain, Baggerly & Scott (1994). Duong & Hazelton (2003) overcame this disadvantage by providing an alternative that always produces a finite bandwidth matrix in bivariate cases. However, further extension of the plug-in bandwidth selection algorithm to higher dimension settings is not very well documented.

2.3.2 Likelihood cross-validation

Likelihood cross-validation is another procedure that aims to select an optimal bandwidth. Let $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$ represent a sample of independent random vectors drawn

from the true multivariate density $f(\boldsymbol{x})$ in $\mathbb{R}^d$. The log likelihood is defined as

$$\log L = \sum_{i=1}^{n} \log f(\boldsymbol{x}_i), \tag{2.3.15}$$

and the estimated log likelihood is written as

$$\log L(H) = \sum_{i=1}^{n} \log \hat{f}_H(\boldsymbol{x}_i). \tag{2.3.16}$$

Kullback-Leibler information is a divergence measure between $f(\boldsymbol{x})$ and $\hat{f}_H(\boldsymbol{x})$. In this case, Kullback-Leibler information is defined as

$$d_{\mathrm{KL}}\left(f, \hat{f}_H\right) = \int_{\mathbb{R}^d} \log\left[\frac{f(\boldsymbol{x})}{\hat{f}_H(\boldsymbol{x})}\right] f(\boldsymbol{x}) d\boldsymbol{x} \tag{2.3.17}$$

$$= \int_{\mathbb{R}^d} \log[f(\boldsymbol{x})] f(\boldsymbol{x}) d\boldsymbol{x} - \int_{\mathbb{R}^d} \log[\hat{f}_H(\boldsymbol{x})] f(\boldsymbol{x}) d\boldsymbol{x}, \tag{2.3.18}$$

which is non-negative. The procedure to minimise d_{KL} is introduced by Duin (1976). Its statistical properties are discussed in Hall (1982) and Bowman (1984), where its use in density estimation is discussed in Hall (1987*a*,*b*).

The recent study of Zhang et al. (2006) employed this procedure to choose a bandwidth matrix that directly maximises the second term in (2.3.18), which can be approximated by

$$\widehat{\mathrm{E}} \log\left[\hat{f}_H(\boldsymbol{x})\right] = \frac{1}{n}\sum_{i=1}^{n} \log \hat{f}_H(\boldsymbol{x}_i) = \frac{1}{n}\sum_{i=1}^{n} \log\left[\frac{1}{n}\sum_{j=1}^{n} K_H(\boldsymbol{x}_i - \boldsymbol{x}_j)\right]. \tag{2.3.19}$$

However, directly maximising (2.3.19) with respect to H results in a matrix of zeros. As noted by Härdle (1991) and Pagan & Ullah (1999), one way out of this situation is

to limit the estimation of $\hat{f}_H$ on the subset $\{\boldsymbol{x}_j : j \neq i\}$ to approximate the likelihood cross-validation based on Kullback-Leibler information

$$\mathrm{CV}_{\mathrm{KL}}(H) = \frac{1}{n}\sum_{i=1}^{n} \log \hat{f}_{H,i}(\boldsymbol{x}_i), \tag{2.3.20}$$

where $\hat{\hat{f}}_{H,i}(\boldsymbol{x})$ is the leave-one-out estimator given in (2.3.12) and can be further expressed as

$$\hat{\hat{f}}_{H,i}(\boldsymbol{x}) = \frac{1}{n-1}\sum_{\substack{j=1 \\ j\neq i}}^{n} |H|^{-1/2} K\left(H^{-1/2}(\boldsymbol{x}_i - \boldsymbol{x}_j)\right). \tag{2.3.21}$$

Therefore, the bandwidth $\hat{H}_{\mathrm{KL}}$ is the optimal bandwidth when it satisfies

$$\widehat{H}_{\mathrm{KL}} = \arg\max_{H} \mathrm{CV}_{\mathrm{KL}}(H). \tag{2.3.22}$$

Hence, the likelihood cross-validation is equivalent to a maximisation problem and requires a numerical optimisation method in practice. However, the implementation is difficult as the dimension of the data increases.

One should be aware of the shortcomings when using the likelihood cross-validation in density estimation. As discussed in Hall (1987*a*,*b*), the support of the kernel function must not be compact, and if the true underlying density has thicker tails, so must the kernel. This problem is also called the tail effects.

It can be understood by considering a simple situation where we are estimating a density with long tails. An observation $\boldsymbol{x}$ may locate in the tail of the distribution where the density is low and data are sparse. If the kernel function is compactly supported or has thin tails, the estimated density for $\boldsymbol{x}$ will be lower than the true density. In an

extreme case, the estimated density is very close to zero and $\log \hat{f}_H(\boldsymbol{x})$ will approach $-\infty$, which will dominate the Kullback-Leibler information. Although the normal kernel is not compactly supported, it does have thin tails. Therefore, Kullback-Leibler information may be misleading in this situation.

As discussed in Härdle (1991), maximising $\mathrm{CV}_{\mathrm{KL}}(H)$ will force us to choose large bandwidth to prevent such cases from happening. This may lead to slight over-smoothing for the higher density regions. In order to overcome such a disadvantage, it is sensible to use adaptive bandwidths and allow for different bandwidths for high and low density regions.

2.3.3 Adaptive bandwidth kernel density estimation

The least square and likelihood cross-validation rules described so far assumes that the bandwidth matrix to be fixed for all data points. This approach is called global bandwidth estimator because it applies the same degree of smoothing for all data points. It works well for many densities, especially for densities that are unimodal and short-tailed. However, the problem of using a global bandwidth is that the kernel methods often produce unsatisfactory results for complex or irregular densities.

Sain & Scott (1996) showed a good bimodal example in univariate setting. Consider a bimodal normal mixture distribution $f(x) = 3/4\phi(x + 3/2) + 1/4\phi_{1/3}(x - 3/2)$, where $\phi_\sigma(x - \mu)$ is a normal density with mean μ and variance σ^2. This density has two modes of the same height but different spread. As discussed in Sain (2002), the global bandwidth selection approach resulted in bandwidth $h = 0.248$ for sample size 200. If apply bandwidth selection separately to each mode the smoothing parameters were

$h = 0.403$ and 0.175, respectively. This suggested that a global bandwidth selection method has to make compromise to each mode and resulted in under-smoothing of the first mode and over-smoothing of the second mode. Jones (1990) presented another graphical illustration of how a global bandwidth could mismanaged a long-tailed density.

The solution is to have a broader kernel to give more smoothing in regions of low density where data are sparse and use a narrower kernel for the high density where less smoothing is necessary near the mode (Silverman 1986, Sain 2002). The response to the suggestion is to let the bandwidth vary with the data and also with the characteristics of the density of interest, which is referred to as adaptive bandwidth estimator.

Several versions of adaptive bandwidth estimator has been studied. For example, Mielniczuk et al. (1989) proposed to use a weighting function on data point $\boldsymbol{x}$ in a global bandwidth estimator. A more common approach is to make the bandwidth as a function of local data point.[1] Two forms of adaptive bandwidth estimator are common in the literature (see, for example Scott 1992). The first form makes the smoothing matrix to depends on the estimation point $\boldsymbol{x}$ and is called the balloon estimator by Terrell & Scott (1992). The general form of such estimator is given as

$$\hat{f}_{\mathrm{B}}(\boldsymbol{x}) = \frac{1}{n|H(\boldsymbol{x})|^{1/2}} \sum_{i=1}^{n} K\left(H(\boldsymbol{x})^{-1/2}(\boldsymbol{x}-\boldsymbol{x}_i)\right) = \frac{1}{n}\sum_{i=1}^{n} K_{H(\boldsymbol{x})}(\boldsymbol{x}-\boldsymbol{x}_i), \tag{2.3.23}$$

where $H(\boldsymbol{x})$ is the bandwidth matrix at estimation point $\boldsymbol{x}$. At each estimation point, the same kernel used to compute the density estimate, which is pointwise equivalent

[1] Nolan & Marron (1989) discussed this issue from the more general Delta-sequence estimator perspective.

to a global bandwidth estimator given in (2.3.1). This estimator was first introduced by Loftsgaarden & Quesenberry (1965) as the kth nearest-neighbour (k-NN) estimator. However, the balloon estimator does not integrate to one and is therefore not a good choice for estimating the density function f (Terrell & Scott 1992, Izenman 1991).

The second form is called the sample-point estimator and is given by

$$\hat{f}_S(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|H(\boldsymbol{x}_i)|^{1/2}} K\left(H(\boldsymbol{x}_i)^{-1/2}(\boldsymbol{x}-\boldsymbol{x}_i)\right) = \frac{1}{n}\sum_{i=1}^{n} K_{H(\boldsymbol{x}_i)}(\boldsymbol{x}-\boldsymbol{x}_i), \tag{2.3.24}$$

where $H(\boldsymbol{x}_i)$ is the bandwidth matrix for the sample data point $\boldsymbol{x}_i$. Hence, the bandwidth changes with the sample data points, and regardless where the density is going to be estimated. This type of estimator is introduced by Breiman et al. (1977), who suggested choosing the bandwidth proportional to $f(\boldsymbol{x}_i)^{-1/d}$. Abramson (1982*a*,*b*) provided the square root law of choosing the bandwidth proportional to $f(\boldsymbol{x}_i)^{-1/2}$, where a pilot estimate of f is used in practice.

The sample-point estimator is a complete-adaptive estimator because it assigns different bandwidths to different sample points, but it is this very flexibility that makes it difficult to estimate or choose bandwidths for. With multivariate data the complete-adaptive density estimator assigns n different bandwidth matrices to n observations, and if a diagonal bandwidth matrix is employed, the number of bandwidths required will be $n \times d$ for d-dimensional data. One way to reduce the level of difficulty involved is to apply the sample-point estimator to grouped or binned data (see for example,

Sain & Scott 1996, Sain 2002). Such a density estimator is given as

$$\hat{f}_S(\boldsymbol{x}) = \frac{1}{n}\sum_{j=1}^{m}\frac{n_j}{|H(\boldsymbol{t}_j)|^{1/2}}K\left(H(\boldsymbol{t}_j)^{-1/2}(\boldsymbol{x}-\boldsymbol{t}_j)\right) = \frac{1}{n}\sum_{j=1}^{m} n_j K_{H(\boldsymbol{t}_j)}(\boldsymbol{x}-\boldsymbol{t}_j), \tag{2.3.25}$$

where m is the number of bins, n_j is the number of data points in the jth bin, $\boldsymbol{t}_j$ is the centre of the jth bin, and $H_{\boldsymbol{t}_j}$ is the bandwidth matrix associated with the jth bin. Therefore, the each bin has a fixed smoothing bandwidth matrix associated to it. By using LSCV criterion, the bivariate simulation results of Sain (2002) showed that the binned sample point estimator outperforms the balloon estimator.

2.3.4 Motivation for new adaptive density estimator

By allocating a different bandwidth matrix to each bin, the binned sample-point estimator obviously reduces the number of bandwidths that need to be assigned. However, the number of bandwidths for the binned sample-point density estimator still grows exponentially with the dimension. For example, when there are m bins in each dimension, the number of bandwidths to be estimated is m^2 for bivariate data, m^3 for trivariate data, and so on. Hence a key issue of adaptive density estimator is how to put data into a small number of groups while still preserving the intuition of adaptive density estimator.

In this thesis, we propose to divide the observations into two regions or groups, namely a low-density region (LDR) and a high-density region (HDR), and assigning two different bandwidth matrices to the two regions. (More generally, of course, we could allocate m percentiles and assign $m+1$ different bandwidth matrices.) In this way, the number of bandwidths to be estimated is obviously reduced. When the true

distribution is unimodal, the low-density region corresponds to the tails, and intuitively, the low-density region should be assigned larger bandwidths than the high-density region. We call this type of kernel density estimator the *tail-adaptive density estimator*.

The idea of dividing observations into low- and high-density regions is not new. Hartigan (1975, 1987) defined clusters of observations as regions associated with different density values and Hyndman (1996) presented an algorithm for computing and graphing data in different density regions. A comprehensive review of applications relating to the issue of low-and high-density regions is given in Mason & Polonik (2009). Samworth & Wand (2010) considered a univariate bandwidth selection method for high-density region estimation, but we are not aware of any previous work that has adopted the concept of grouping data into low- and high-density regions as a mechanism for assigning bandwidth matrices for multivariate kernel density estimation, as we do here.

A major difficulty faced by the practitioner when implementing the (binned) sample-point estimator is how to assign values to the various bandwidth parameters. Likelihood cross-validation could be used to estimate the bandwidths, but this method is likely to encounter severe computational difficulties due to the large number of bandwidths involved.

To battle the computation difficulties of estimating large number of bandwidths in multivariate data, we consider using an adaptation of the Bayesian sampling approach that has recently been investigated. Brewer (2000) presented a Bayesian sampling procedure for estimating variable bandwidths in a univariate setting and showed that the

Bayesian kernel density estimation method produced better performance than the so-called binning method proposed by Sain & Scott (1996). Kulasekera & Padgett (2006) discussed Bayes estimation of a global bandwidth for kernel density estimation based on univariate censored data using an asymmetric kernel. de Lima & Atuncar (2010) derived a closed form Bayes estimate of a global bandwidth matrix for multivariate kernel density estimation using an extension of the Bayesian bandwidth selector proposed by Gangopadhyay & Cheung (2002) for univariate density estimation.

Zhang et al. (2006) proposed a Bayesian approach as the numerical solution to this issue. The Bayesian approach treats the nonzero elements in H as parameters and obtain the posterior density from likelihood cross-validation. Their study employs the Markov Chain Monte Carlo (MCMC) algorithm in order to estimate H and is applicable to any dimension and with no increased difficulty as the dimension in the data increases. The simulation results show better performance of the Bayesian approach comparing to methods proposed by Duong & Hazelton (2003) and the Normal Reference Rule. Given the advantages of the Bayesian approach, we will apply this method to the problems in this project.

2.3.5 Concluding remark

As discussed in the Section 2.2.2 and Section 2.2.3, the kernel density estimator with a global bandwidth is often inadequate for data with complex and irregular densities. Recent investigations have promoted the idea of data-driven adaptive bandwidth density estimation. Although the problem has been intensely investigated in univariate settings, less attention appears to have been paid to data-driven methods for adaptive

bandwidth estimation for multivariate data. To remedy this problem, we propose the tail-adaptive density estimator, where data are grouped into low- and high-density regions and different bandwidth matrices are assigned to observations in different regions. To estimate the bandwidths, we adopt the MCMC algorithm proposed by Zhang et al. (2006), where the bandwidths are treated as parameters. The tail-adaptive density estimator shall also attempt to fix the "tail effect", from which the global bandwidth selection under Kullback-Liebler information usually suffers.

2.4 Density-based clustering

The concept of multi-modality is closely related to the concept of "cluster" in clustering analysis. As defined by Hartigan (1975, p205), "Clusters may be thought of as regions of high density separated from other such regions by regions of low density". Therefore, it is intuitively easy to understand that the true cluster in population is associated with a mode in f, which can be represented by the empirical modes in $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$ draw from f. First, a kernel density estimate $\hat{f}(\boldsymbol{x}_i)$ can be obtained. For an given density level τ chosen by the user, observations with $\hat{f}(\boldsymbol{x}_i) > \tau$ are treated as a level set $L(\hat{f} > \tau)$ and observations with $\hat{f}(\boldsymbol{x}_i) \leq \tau$ are considered as noise.

The basic idea of density-based clustering is to identify the association of the observations and the empirical modes by finding the connected components in the level set.[2] There are mainly two aspects that need to be addressed. First, how to find the

[2] Density-based clustering is one branch of clustering algorithms. Please see Berkhin (2006) and Xu et al. (2005) for a survey of other clustering algorithms.

connected components in the level set. Second, how to estimated the number of connected components in the level set. We wish to combine these two issues into adaptive kernel density estimation.

2.4.1 Finding connected components in the level set

Many studies have been focused on estimating the level set. a comprehensive review of general applications of level set estimators is presented by Mason & Polonik (2009). In the context of clustering, a classical algorithm to locate clusters is the k-means introduced by MacQueen (1967). It partitions n observations in to k clusters ($k \leq n$) by minimizing the total Euclidean distance between cluster members and the cluster centres. The algorithm is iterative of two steps. First, k seeds are assigned as cluster means by the user and each observation is given a cluster membership depending on which cluster mean is the closest to its location. Next, the cluster means are re-calculated based on the observations in each cluster. The algorithm iterates until assignment of cluster membership is not changing. This algorithm is very efficient when the number of cluster is correctly chosen. However, choosing the wrong cluster number can produce poor results.

Ester et al. (1996) proposed an algorithm called DBSCAN for spatial data. The algorithm consists of four steps, (a) calculate a kernel density estimate for each observation; (b) choose a density level τ and obtain the level set with $\hat{f}(\boldsymbol{x}_i) > \tau$; (c) construct a graph connecting each observation to other observations in the level set within a distance r; (d) define the connected components in the graph as clusters. Walther (1997)

presented a similar method for estimating level set by constructing the union of balls around the observations in the level set but does not contain noise points.

Another well known clustering method is the single linkage method (Aderberg 1973). The algorithm starts with n clusters and at each stage it merges two closet points to form a new cluster. The distance between two clusters are calculated based on the smallest distance between any members of the two clusters. However, Hartigan (1981, 1985) suggested that asymptotically single linkage is not consistent when dimension is higher than one. The single linkage method belongs to the hierarchical clustering approach. Other studies in this direction includes Wong & Lane (1983), who used a kth nearest neighbours density estimate, and Stuetzle (2003), who presented a method using the minimum spanning tree of a sample. Other studies by Klemelä (2004, 2006) aim to plot the tree structure of multivariate density estimates, and visually identify the clusters. Hence, hierarchical method requires some form of supervision from the user to decide when to stop the merging or deciding how many clusters exists.

A graph based clustering method is proposed by Azzalini & Torelli (2007). The algorithm proposed by the study suggested to use a Delaunay triangulation of the observations in the level set to form the cluster cores. However, the method to search for the connected components in the graph is not very well documented specially high dimensional data.

To our purpose, in order to make the kernel density estimator be adaptive on the modes or clusters, a clustering algorithm is needed to identify the different modes in the density function and also the number of modes there exists.

2.4.2 Finding the number of clusters

Automatic detection of the number of clusters in a data set is one of the most challenging problems in cluster analysis. This problem has been addressed in the literature by many studies. Please see Gordon (1999) for some classical techniques, and Jain (2010) for some references of the parametric approach. Tibshirani et al. (2001) proposed the gap statistic to estimate the number of clusters. The study formalized the elbow phenomenon often seen in the plot of number of clusters T against within-cluster dispersion W_T. As T increases, W_T decreases and the decrease in W_T becomes noticeably flat after certain value of T. A similar approach presented by Sugar & James (2003) uses a different measure of within-cluster dispersion. However, both studies require other clustering algorithm such as k-means to find the connected components.

Burman & Polonik (2009) proposed an algorithm via a kth nearest neighbours approach. Their method is briefly explained as follows. Let $k_1, k_2 < n$, and $\hat{d}(\boldsymbol{x})$ be the distance between $\boldsymbol{x}$ to its k_1th nearest neighbours. The algorithm has iterative steps:

1. Calculate $\hat{d}(\boldsymbol{x}_i)$, for $i = 1, \ldots, n$, and find the first modal as $M_1 = \arg\min_{\boldsymbol{x}_i} \hat{d}(\boldsymbol{x}_j)$.

2. Eliminate all observations points that are k_2 nearest neighbours of M_1. Denote the remaining data by S_1.

3. Find the second modal by $M_2 = \arg\min_{\boldsymbol{x}_i \in S_1} \hat{d}(\boldsymbol{x}_j)$.

4. Eliminate all data points that are k_2 nearest neighbours of either M_1 or M_2 to find $S_1 \subset S_2$.

5. Find the third modal via $M_3 = \arg\min_{\boldsymbol{x}_i \in S_2} \hat{d}(\boldsymbol{x}_j)$.

6. Repeat until no candidate mode is found.

Burman & Polonik (2009) suggested that the optimal choice of k which minimizes MISE can be shown as depends on some quantity of the unknown true density function. However, the data driven algorithm for the selection of k is not well documented.

A density-based estimation method was proposed by Cuevas et al. (2000, 2001). The studies presented a level set estimator by forming union of balls around all the observations in the level set, while the number of clusters is estimated simultaneously. The algorithm is called CRA by the authors. The basic idea of CRA algorithm is to approximate the level set $L(\hat{f} > \tau)$ (via kernel density estimation) by a set estimator of a union of balls (Devroye & Wise 1980) defined by

$$\widehat{L} = \bigcup_{i=1}^{k} B(\boldsymbol{x}_i, \epsilon), \tag{2.4.1}$$

where $\boldsymbol{x}_i, i = 1, \cdots, k$ denote the sampling observations that $\boldsymbol{x}_i \in \{\hat{f} > c\}$ (k is random), and $B(\boldsymbol{x}_i, \epsilon)$ is closed sphere or ball centered at $\boldsymbol{x}_i$ with radius $r \geq 0$ given by

$$B(\boldsymbol{x}_i, \epsilon) = \{\boldsymbol{z}_i : \boldsymbol{z}_i \in \mathbb{R}^d \text{ and } \|\boldsymbol{x}_i - \boldsymbol{z}_i\| \leq \epsilon\}. \tag{2.4.2}$$

Therefore, the estimator $\widehat{T}$ as a number of connected component of $\widehat{L}$ is defined as

$$\widehat{T} = T(\widehat{L}). \tag{2.4.3}$$

A connected component of $\widehat{L}$ is associated with a spanning tree with vertices $\boldsymbol{x}_i$ and edges smaller than 2ϵ. In other words, within a connected component of $\widehat{L}$, we can join

every pair of $\boldsymbol{x}_i$ with a path consisting a finite number of edges with length smaller than 2ϵ. One of the easy way in many ways to calculate ϵ is

$$\epsilon = \max_i \min_{j \neq i} \|\boldsymbol{z}_i - \boldsymbol{z}_j\|/2. \tag{2.4.4}$$

As discussed in Cuevas et al. (2000) the value obtained from (2.4.4) could be too conservative, which tends to overestimate the number of connected components. Therefore, a clustering criterion must be assumed in advance. For a connected component to be recognized as a cluster, it must have at least 5% of the data. This criterion is also suggested by Cuevas et al. (2000).

2.4.3 Concluding remarks

The motivation for using adaptive bandwidths mainly comes from two aspects. First, observations in low density region require higher level of smoothing compared to observations in high density region. Second, the underlying density can be multi-modal, where each mode may have different direction and spread. We discussed the first aspect and proposes the tail-adaptive kernel density estimator, in which the observations are divided into two regions or groups and two different bandwidth matrices are assigned to the two regions.

We wish to propose a cluster-adaptive kernel density estimator that address the multi-modality issue. The idea has been briefly discussed in Sain (2002) as a way to reduce the dimensionality problem raised from the binned adaptive kernel density estimator. In this project, we wish to extend the tail-adaptive kernel density estimator to be adaptive on the modes of the density function.

In order to find the modes in the data, we wish to employ one of the density based clustering algorithms to automatically search the clusters in the data set where multimodality exists. The CRA algorithm proposed by Cuevas et al. (2000, 2001) is closely linked the kernel density estimation as it requires the kernel density estimation of the high density region (HDR). Moreover, the calculation of ϵ in the CRA algorithm is straight forward.

Therefore, we aim to combine the tail-adaptive kernel density estimator and CRA algorithm to make cluster-adaptive density estimation. This means we are able to use the clustering information obtained from CRA algorithm to automatically allocate different bandwidth matrices to each cluster in the data set, where the bandwidths parameters are estimated through a proposed MCMC algorithm.

2.5 Volatility models for financial returns

Empirical evidence has shown that the volatility of financial asset returns is often highly persistent and asymmetrically distributed. Early studies have focused on the parametric approach of ARCH (Engle 1982) and GARCH (Bollerslev 1986) type models to capture the stylized facts. The nonparametric branch of research on volatility models has aimed at addressing the strong parametric assumption of ARCH and GARCH type models, such as the linearity assumption in the volatility equation (e.g. Pagan & Schwert 1990) and the distribution function of error term (e.g. Engle & Gonzalez-Rivera 1991).

More recently, the nonlinear nonstationary heteroscedastic (NNH) model was proposed (Park 2002) as an alternative class of volatility models. The NNH model assumes the conditional variance as a known parametric nonlinear function of a persistent explanatory variable x_t. Han & Park (2008) extended the NNH model by allowing the ARCH(1) component in the model, while Han & Zhang (2012) proposed a nonparametric version of the NNH model called the nonstationary nonparametric volatility (NNV) model.

2.5.1 ARCH and GARCH models

The autoregressive conditional heteroscedasticity (ARCH) model proposed by Engle (1982) is the pioneering model that aims to capture the time-varying volatility. Let y_t be the return series, and $u_t = y_t - \mu$ be the innovation at time t, the ARCH model is defined by

$$\begin{aligned} u_t &= \sigma_t \varepsilon_t, \qquad \varepsilon \sim i.i.d.(0,1) \\ \sigma_t^2 &= \alpha_0 + \sum_{i=1}^{q} \alpha_i u_{t-i}^2 \end{aligned} \tag{2.5.1}$$

where σ_t^2 is the variance of u_t conditional on the information available at time t. This model is referred to as the ARCH(q) model. The parameters in the volatility equation must satisfy $\alpha_0 > 0$, and $\alpha_i \geq 0$, for $i = 1, 2, \cdots, q$, with at least one $\alpha_i > 0$ to guarantee the conditional variance of u_t to be strictly positive. The error term ε_t is assumed to have zero mean and variance one, and the common choice is the standard normal distribution. However, there are several problems when fitting a ARCH(q) model to

financial return series. The main issues is that the order q needs to be large to accommodate the high persistence in volatility, which makes the model less parsimonious.

Bollerslev (1986) proposed an extended ARCH model to be the generalized ARCH (GARCH) model. The GARCH(p,q) model is given by

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i u_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2, \tag{2.5.2}$$

where constraints such as $\alpha_0 > 0$, $\alpha_i \geq 0$ and $\beta_j \geq 0$ are imposed to the conditional variance is positive. In practice, p and q are chosen to be 1. In fact, GARCH(1,1) is a very popular volatility model because it is parsimonious while still mimics a ARCH(∞) process. However, the GARCH model could not capture the leverage effects because it assumes symmetric effect of positive and negative shocks from (u_{t-i}) on σ_t. In addition, the GARCH model produces exponential decay in the autocorrelation function of squared return and therefore could not take into account of the long memory in volatility.

To be able to capture asymmetric effect in volatility, Nelson (1991) proposed the exponential GARCH (EGARCH) model

$$ln(\sigma_t^2) = \alpha_0 + \sum_{i=1}^{q} \alpha_i \frac{|u_{t-i}| + \gamma_i u_{t-i}}{\sigma_{t-i}} + \sum_{j=1}^{p} \beta_j ln(\sigma_{t-j}^2). \tag{2.5.3}$$

Since $\varepsilon_{t-i} = u_{t-i}/\sigma_{t-i}$ and σ_{t-i} is positive, a positive u_{t-i} will cause the log volatility to increase by $\alpha_i(1+\gamma_i)|\varepsilon_{t-i}|$, and a negative u_{t-i} will cause the log volatility to decrease by $\alpha_i(1-\gamma_i)|\varepsilon_{t-i}|$.

An alternative model that aims to capture the leverage effect is presented by Glosten et al. (1993) called the threshold GARCH (TGARCH) model. A TGARCH(p,q) has the form

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} (\alpha_i + \gamma_i I_{t-i}) u_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2, \tag{2.5.4}$$

where I_{t-i} is an indicator function such that

$$I_j = \begin{cases} 1 & \text{if } u_{t-i} < 0 \\ 0 & \text{if } u_{t-i} \geq 0 \end{cases},$$

and γ_i is nonnegative,α_i and β_j should satisfy similar conditions to those of (2.5.2). Since $\gamma_i > 0$, a negative shock from u_{t-i} would have a larger impact on σ_t^2 than a positive shock of u_{t-i}.

If $\sum_{i=1}^{q} \alpha_i + \sum_{j=1}^{p} \beta_j = 1$, the model is called integrated GARCH model (IGARCH) (Engle & Bollerslev 1986). Comparing with the GARCH model, the impact of past squared innovations u_{t-1}^2 in IGARCH model is persistent. A special case of IGARCH(1,1) model is the RiskMetrics defined by

$$\sigma_t^2 = (1 - \beta_1) u_{t-1}^2 + \beta_1 \sigma_{t-1}^2, \tag{2.5.5}$$

which is well-known for calculating value at risk (VaR).

Empirical studies have found the existence of long-memory in autocorrelations of squared or absolute returns in financial asset returns (see for example Ding et al.

1993). This finding has motivated Baillie et al. (1996) to develop the fractionally integrated GARCH (FIGARCH) model, defined by

$$[1-\beta(L)]^d \sigma_t^2 = \alpha + [1-\beta(L)-\phi(L)(1-L)^d]u_t^2, \tag{2.5.6}$$

where $0 < d < 1$, L denotes the lag operator, $\phi(L) = [1-\alpha(L)-\beta(L)](1-L)^{-1}$, $\alpha(L) = \alpha_1 L + \alpha_2 L^2 + \alpha_q L^q$, $\beta(L) = \beta_1 L + \beta_2 L^2 + \beta_p L^p$ and $(1-L)^d$ is the fractional differencing operator.

The (G)ARCH type models assumes that the conditional variance of stock returns is generated from an autoregressive moving average fashioned process. It is well documented that (G)ARCH type models are capable of capturing volatility clustering. See Bollerslev et al. (1992) and Bollerslev et al. (1994) for some reviews of ARCH type models.

2.5.2 Volatility models with covariates

However, the (G)ARCH models assumes that volatility of a financial asset returns only relates to information from its own history. It is commonly believed that the return of one financial market could be affected by other factors such as volatilities from other markets. Hence, studies has shown that external variables may contain significant information for the volatility of a financial series. For example, Lamoureux & Lastrapes (1990) used trading volumes as an independent variable in their volatility model. Engle, Ng & Rothschild (1990) and Engle & Patton (2001) investigated the relationship

between U.S. treasury bill returns and stock market returns. Other studies has examined the impact of news from different countries and bid-ask spreads in foreign exchange markets (Engle, Ito & Lin 1990, Bollerslev & Melvin 1994).

Alternative to the ARCH type models, Park (2002) introduced a class of volatility model which assumes the conditional variance as a function of some explanatory variable x_t. This model is called the nonlinear nonstationary heteroscedastic (NNH) model and defined by

$$\begin{aligned} \sigma_t^2 &= f(x_{t-1}), \\ x_t &= \rho x_{t-1} + \nu_t \end{aligned} \tag{2.5.7}$$

and the mean equation in (2.5.1) can be re-written as

$$\begin{aligned} u_t^2 &= f(x_{t-1}) + \eta_t \\ \eta_t &= f(x_{t-1})(\varepsilon_{t-1}^2 - 1) \end{aligned} \tag{2.5.8}$$

where x_t has a unit root with $\rho = 1$. $f(\cdot)$ is a known nonnegative and nonlinear function and should belong to the integrable and asymptotically homogeneous function class (as discussed in Park & Phillips 1999, 2001). Han & Park (2008) combined the NNH model with the ARCH(1) model. Such model is defined as

$$\begin{aligned} \sigma_t^2 &= \alpha u_{t-1}^2 + f(x_{t-1}), \\ x_t &= \left(1 - \frac{c}{n}\right) x_{t-1} + \nu_t \end{aligned} \tag{2.5.9}$$

where $c \geq 0$ and such model is called the ARCH-NNH model. In addition to the NNH model, the ARCH-NNH model allows for exact unit root in x_t as well as near unit root. The study suggested using the QMLE to estimate the parameters of the model as long as the parameters are linear in the volatility equation. Hence, the paper used $f(x) = a|x|^b$ with $b = 1$ as the parametric functional form for f. The forecasting results show that the ARCH-NNH model out performs GARCH(1,1) and FIGARCH(1,1) in lower frequency data such as weekly and monthly.

The limitations are that the study only considered $f(x) = a|x|^b$ or $f(x) = a|x|$ as the volatility function and use QMLE to estimate the parameters of the model if the parameters are linear in the volatility equation.

On the other hand, Han & Zhang (2012) presented a nonparametric version of the NNH model called the nonstationary nonparametric volatility (NNV) model

$$\sigma_t^2 = m(x_{t-1}), \tag{2.5.10}$$

where $m(\cdot)$ is smooth but unknown function and $m(x_t) > 0$ for all t. x_t is able to follow the unit root or near unit root process defined in (2.5.9). This model is an application of the nonparametric cointegration model discussed by Wang & Phillips (2009*a*,*b*), in which $m(x)$ is estimated via the Nadaraya-Watson estimator. It was shown that the model can generate long memory property in volatility as long as the explanatory variable contains a unit root. Han & Zhang (2012) suggested that by choosing the bandwidth based on QLIKE loss criterion, the NNV model outperforms GARCH model in both in-sample and out-of-sample evaluation. However, this model assumes

the volatility is determined only by external variable. We believe it is reasonable to assume that the volatility depends on the historical information contained in the return series itself.

2.5.3 Nonparametric and semiparametric volatility models

A number studies has promoted the use of a nonlinear function in the volatility equation in ARCH models. Pagan & Schwert (1990) proposed an nonparametric ARCH model where

$$\sigma_t^2 = m(u_{t-1}^2), \tag{2.5.11}$$

where $m(\cdot)$ is estimated via the Nadaraya-Watson estimator defined by

$$\hat{m}(\boldsymbol{x}) = \frac{\sum_{t=1}^{n} u_t^2 K_h(\boldsymbol{x}_t - \boldsymbol{x})}{\sum_{t=1}^{n} K_h(\boldsymbol{x}_t - \boldsymbol{x})}, \tag{2.5.12}$$

where $K_h(\boldsymbol{x}_t - \boldsymbol{x}) = h^{-1}K(\boldsymbol{x}/h)$ is the kernel density function and h is the bandwidth. A study by Masry & Tjøstheim (1995) also employed the Nadaraya-Watson estimator. Härdle & Tsybakov (1997) considered using local polynomial functions to estimate the ARCH volatility function with the number of lag equal to 1. The study identified the asymmetric relationship between return and volatility in exchange rate data. Yang et al. (1999) later proposed another method, which could include the number of lag up to q. Franke et al. (2004) suggested a bootstrap method can be used to estimate the nonparametric ARCH(1) model.

However, the ARCH specification is often difficult to capture the dynamic nature of financial series, since it often requires a large number of lags. Hence, a GARCH setting

is more appropriate. Engle & Ng (1993) proposed a partially nonparametric GARCH model defined as

$$\sigma_t^2 = \alpha_0 + \beta\sigma_{t-1}^2 + m(u_{t-1}), \tag{2.5.13}$$

where $m(\cdot)$ is a linear spline function which guarantees continuous. Such model allows for asymmetric leverage effect. A study by Linton & Mammen (2005) studied a class of semiparametric ARCH(∞) models, that includes (2.5.13) as a special case (see Linton 2009, for a review).

The performance of the Nadaraya-Watson estimator depends mostly on the choice of the bandwidth (Härdle 1990), and the bandiwidth selection process is closely related to bandwidth selection of kernel density estimation. Such Bayesian technique has been shown to outperform traditional cross-validation methods such as bootstrap and rule-of-thumb estimators (see Jones et al. 1996, Zhang et al. 2006, for a survey).

Other nonparametric studies includes, Härdle & Tsybakov (1997) considered using local polynomial functions to estimate the ARCH volatility function with the number of lag equal to 1. Yang et al. (1999) later proposed another method, which could include the number of lag up to q. Franke et al. (2004) suggested a bootstrap method can be used to estimate the nonparametric ARCH(1) model. Since the ARCH specification is often difficult to capture the dynamic nature of financial series, a GARCH setting is preferred. For instance, Engle & Ng (1993) proposed a partially nonparametric GARCH model that allows for asymmetric leverage effect. A study by Linton & Mammen (2005) studied a class of semiparametric ARCH(∞) models, that includes the partially nonparametric GARCH model of Engle & Ng (1993) as a special case

(see Linton 2009, for a review). Other nonparametric GARCH(1,1) studies include Bülmann & McNeil (2002) and Yang (2006), where both suggested improved performance over the traditional GARCH(1,1) model.

Another branch of studies has focused on adopting an unknown distribution function for the error term. The early studies on (G)ARCH models has assumed the distribution of ε_t in (2.5.1) is conditional normal and parameters are estimated via maximum likelihood estimator (MLE). Bollerslev & Wooldridge (1992) suggested if the first two moments of the underlying GARCH process are correctly specified, the quasi maximum likelihood estimatior (QMLE) can still produce consistent estimates under the assumption of conditional normality, even if the error term is not normally distributed. However, QMLE suffers from efficiency loss in such situation. In fact, many studies has suggested the normality assumption can be too strong (see French et al. 1987, Badrinath & Chatterjee 1988, Hall & Yao 2003, for example).

Given such constraints under parametric setting, Engle & Gonzalez-Rivera (1991) showed the efficiency loss under QMLE and introduced the semiparametric GARCH model, where the density of ε_t is of an unknown functional form. The paper employed the discrete maximum penalized likelihood estimation technique and suggested that semiparametric method is more efficient over QMLE. Drost & Klaassen (1997) developed a kernel based estimator for the unknown density function of ε_t and showed the efficiency bounds of the estimation of the parameters.

A recent study by Zhang & King (2011) proposed a Bayesian method to simultaneously estimate the GARCH parameters and conditional error density without specifying the error distribution function. The paper assume the density of ε_t is unknown and is

approximated by a kernel density function:

$$f(\varepsilon_t; h) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K\left(\frac{\varepsilon_t - \varepsilon_i}{h}\right) \tag{2.5.14}$$

where $K(\cdot)$ is the standard Gaussian density function. Zhang & King (2011) presented a Bayesian sampling technique which can estimate the GARCH parameters and the error density simultaneously. The empirical results indicate strong evidence on better performance achieved by the Bayesian method comparing to traditional parametric GARCH(1,1) model.

2.5.4 Concluding Remarks

Empirical evidence has shown that volatility of financial asset returns is often highly persistent and asymmetrically distributed. Early studies have been focused on the parametric approach of ARCH and GARCH type models to capture the stylized facts. The nonparametric branch of research on volatility models has aimed at addressing the parametric assumption of ARCH and GARCH type models, such as the linearity assumption in the volatility equation and the distribution function of error term. In addition, the NNH model assumes the conditional variance as a known parametric nonlinear function of a persistent explanatory variable x_t. The NNV model assumes the nonlinear function of x_t in the model to be unknown.

We propose a more general version of the NNV model. We believe that information from previous asset returns could contribute to today volatility, hence it is reasonable to allow an linear ARCH(1) component in the volatility equation. The relationship

between the volatility and the covariate is nonlinear, and is estimated by the Nadaraya-Watson estimator.

In order to estimate the bandwidth parameters, we adopt the Bayesian sampling method shown in Zhang et al. (2009) and Zhang & King (2011) to simultaneously estimate bandwidth parameter and the linear coefficient through an MCMC algorithm. Furthermore, instead of imposing any particular parametric assumption on the error term distribution, we allow the error term distribution to follow an unknown distribution.

Chapter 3

Bayesian Adaptive Kernel Density Estimation of Irregular Distributions

3.1 Introduction

As we discussed in Section 2.2, the performance of global bandwidth kernel density estimator is limited in some situations, and adaptive bandwidth selection for multivariate distributions has received limited attention due to the difficulty of estimating a large number of bandwidths. In this chapter, we propose to remedy this problem with an alternative method called the tail-adaptive kernel density estimator, which assigns two different bandwidth to sample data in low- and high-density regions. In this way, it will improve the performance of the resulting kernel density estimator while still restricting the number of bandwidth at a manageable level.

This chapter is organized as follows. In Section 3.2, we derive the posterior of bandwidth parameters and present an MCMC sampling algorithm for estimating these bandwidths. Sections 3.3 and 3.4 present the results of Monte Carlo simulation studies designed to examine the performance of the tail-adaptive density estimator. In these experiments we consider the issue of bandwidth estimation for univariate, bivariate and 5-dimensional multivariate density estimation using several density functions designed to have irregular shapes, such as multi–modality, skewness and heavy tails. To demonstrate the efficaciousness of our proposed technique, we compare the performance of the tail-adaptive density estimator with the Bayesian global bandwidth estimator and the NRR bandwidth procedure. The results indicate that assigning different bandwidths to LDR and HDR regions can often result in substantial improvements. To illustrate the potential use of our methods for inferential purposes, in Section 3.5 we apply the tail-adaptive density estimator to the estimation of the bivariate density of two asset returns, the continuously compounded daily returns of the All Ordinaries and S&P 500 indices. Employing the estimated density we can compute, for example, the value at risk (VaR) for the ALL Ordinaries conditional on the S&P 500 index taking particular values overnight, say, or similar quantities that might be of interest to market analysts. Section 3.6 concludes the chapter.

3.2 Tail-adaptive kernel density estimator

The concept of grouping observations into low- and high-density regions has been discussed in many statistical problems. Hartigan (1975, p205) defined a cluster as a high-density region that is separated from other high-density regions by low-density

regions. In this project, we are particularly interested in grouping observations into the low-density region, inside which every observation has a density value less than or equal to the density of every observation outside the region. In a different situation, Hyndman (1996) presented a definition for highest density region, and we follow his definition to define the LDR as follows.

Let α be a threshold value that determines the proportion of the low-density region relative to the whole sample space. Let $L(f_\alpha)$ denote a subset of the sample space, so that the $(100 \times \alpha)$% low-density region is shown as

$$L(f_\alpha) = \{\boldsymbol{x} : f(\boldsymbol{x}) \leq f_\alpha\},$$

where f_α is the largest constant such that $\Pr\{\boldsymbol{x} \in L(f_\alpha)\} \leq \alpha$.

Let

$$I_j = \begin{cases} 1 & \text{if } \boldsymbol{x}_j \in L(f_\alpha) \\ 0 & \text{otherwise} \end{cases},$$

for $j = 1, 2, \cdots, n$. Let $\boldsymbol{h}^{(1)}$ denote the bandwidth vector assigned to observations inside $L(f_\alpha)$, and $\boldsymbol{h}^{(0)}$ the bandwidth vector assigned to observations outside $L(f_\alpha)$. The kernel density estimator is

$$\begin{aligned} \hat{f}_{\boldsymbol{h}^{(1)},\boldsymbol{h}^{(0)}}(\boldsymbol{x}) = & \frac{1}{n} \sum_{j=1}^{n} \Big\{ I_j K\left((\boldsymbol{x} - \boldsymbol{x}_j)./\boldsymbol{h}^{(1)}\right)./\boldsymbol{h}^{(1)} \\ & + (1 - I_j) K\left((\boldsymbol{x} - \boldsymbol{x}_j)./\boldsymbol{h}^{(0)}\right)./\boldsymbol{h}^{(0)} \Big\}, \end{aligned} \tag{3.2.1}$$

and its leave-one-out estimator is denoted as $\hat{f}_{\boldsymbol{h}^{(1)},\boldsymbol{h}^{(0)},i}(\boldsymbol{x_i})$

$$\hat{f}_{\boldsymbol{h}^{(1)},\boldsymbol{h}^{(0)},i}(\boldsymbol{x_i}) = \frac{1}{n-1}\sum_{\substack{j=1\\ j\neq i}}^{n}\Big\{I_j K\left((\boldsymbol{x}_i-\boldsymbol{x}_j)./\boldsymbol{h}^{(1)}\right)./\boldsymbol{h}^{(1)} + (1-I_j)K\left((\boldsymbol{x}_i-\boldsymbol{x}_j)./\boldsymbol{h}^{(0)}\right)./\boldsymbol{h}^{(0)}\Big\},$$

for $i = 1, 2, \cdots, n$. As the low-density region becomes the tail area when the underlying density is unimodal, we also call (3.2.1) the tail-adaptive estimator for simplicity. The tail-adaptive density estimator allows for assigning two different bandwidth matrices to observations inside the low- and high-density regions. Note that the value of α can be chosen as either 5% or 10%. Then $L(f_\alpha)$ can be interpreted as the subset that contains the data in the tails of the density. Even though $f(\boldsymbol{x})$ is unknown, f_α can be approximated through the kernel density estimator of $f(\boldsymbol{x})$ using a global bandwidth.

3.2.1 Posterior of bandwidth parameters

Given $\boldsymbol{h}^{(1)}$ and $\boldsymbol{h}^{(0)}$, the approximate likelihood is

$$\ell(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n | \boldsymbol{h}^{(1)}, \boldsymbol{h}^{(0)}) = \prod_{i=1}^{n} \hat{f}_{\boldsymbol{h}^{(1)},\boldsymbol{h}^{(0)},i}(\boldsymbol{x_i}). \tag{3.2.2}$$

as suggested by Zhang et al. (2006), we assume that the prior of each bandwidth to be the Cauchy density $p(h_k^{(l)}) \propto \frac{1}{1+h_k^{(l)}\times h_k^{(l)}}$, for $k = 1, 2, \cdots, d$, and $l = 0$ and 1. The posterior of $\boldsymbol{h}^{(1)}$ and $\boldsymbol{h}^{(0)}$ for given $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ is

$$\pi(\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(0)} | \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n) \propto \left\{\prod_{i=1}^{n} \hat{f}_{\boldsymbol{h}^{(1)},\boldsymbol{h}^{(0)},i}(\boldsymbol{x_i})\right\} \times \left\{\prod_{k=1}^{d} p(h_k^{(1)}) \times p(h_k^{(0)})\right\}. \tag{3.2.3}$$

The posterior given by (3.2.3) is of non-standard form, and we cannot derive an analytical expression as the estimate of $\{\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(0)}\}$. However, we can use the random-walk Metropolis-Hastings algorithm to sample $\{\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(0)}\}$ from (3.2.3).

Random-walk Metropolis-Hastings algorithm uses a symmetric density, e.g. normal density, to generate candidates. As long as the Random-walk Metropolis-Hastings algorithm works, it is generally not necessary to explore other proposal densities.

The sampling procedure is as follows.

1) Obtain an initial kernel density estimator with bandwidths chosen via NRR; and derive the low- and high-density regions for given α.

2) Assign initial values to $\boldsymbol{h}^{(1)}$ and $\boldsymbol{h}^{(0)}$, which are respectively, the bandwidth matrices given to observations within the low- and high-density regions specified in Step 1).

3) Let $\tilde{\boldsymbol{h}}$ denote the vector of all elements of $\{\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(0)}\}$. Use the random-walk Metropolis-Hastings algorithm to update $\tilde{\boldsymbol{h}}$ with the acceptance probability computed through the posterior given by (3.2.3).

4) Derive the low- and high-density regions according the density estimator with the bandwidth matrices updated in Step 3).

5) Repeat Steps 3) and 4) until the simulated chain of $\tilde{\boldsymbol{h}}$ achieves reasonable mixing performance.

During the above iterations, we usually discard the draws during the burn-in period, and record the draws of $\tilde{\boldsymbol{h}}$ thereafter. Let $\{\tilde{\boldsymbol{h}}_{(1)}, \tilde{\boldsymbol{h}}_{(2)}, \cdots, \tilde{\boldsymbol{h}}_{(M)}\}$ denote the recorded draws.

The posterior mean (or ergodic average) denoted as $\sum_{i=1}^{M} \tilde{\boldsymbol{h}}_{(i)}/M$, is an estimate of $\tilde{\boldsymbol{h}}$. Once the bandwidth matrices are estimated, the analytical form of the kernel density estimator is obtained.

3.3 A Monte Carlo simulation study

To investigate the performance of the proposed tail-adaptive kernel density estimator, we approximate Kullback-Leibler information between the density estimator and its corresponding true density via Monte Carlo simulation. Kullback-Leibler information defined in (2.3.18) is a measure of discrepancy between the true density and its estimator. To approximate Kullback-Leibler information, we draw a large number of random vectors $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ from true density $f(\boldsymbol{x})$ and compute

$$\hat{d}_{KL}\Big(f(x), \hat{f}(x)\Big) = \frac{1}{N}\sum_{i=1}^{N} \log\Big(f(\boldsymbol{x}_i)/\hat{f}(\boldsymbol{x}_i)\Big), \tag{3.3.1}$$

where $\hat{f}(\cdot)$ denote a density estimator of $f(\cdot)$. The performance of a bandwidth estimate is examined through the performance of the resulting kernel density estimator. A bandwidth estimation method is better than its competitor if Kullback-Leibler information resulted from the former is less than that resulted from the latter.

3.3.1 True densities

We conduct Monte Carlo simulation by simulating samples from six target densities labeled A, B, C, D, E and F, which are denoted as A_1 to F_1 for univariate densities, and A_2 to F_2 for bivariate densities. Figure 3.1 provides the density plot for univariate

densities and Figure 3.2 shows the contour plot for bivariate densities. These densities are of irregular shapes. Density A and B are normal densities with bimodality. Density E and F are Student t densities with heavy-tail features. Density C and G are skew-normal and skew-t densities, respectively. Their specifications are explained as follows.

Density A is a mixture of two equally weighted normal densities with bimodality:

$$f_A(\boldsymbol{x}|\mu_1,\Sigma_1,\mu_2,\Sigma_2) = \frac{1}{2}\phi(\boldsymbol{x}|\mu_1,\Sigma_1) + \frac{1}{2}\phi(\boldsymbol{x}|\mu_2,\Sigma_2),$$

where $\phi(\boldsymbol{x}|\mu,\Sigma)$ is a multivariate normal density with mean μ and variance-covariance matrix Σ. The univariate true density is $f_{A_1}(x) = 1/2\phi(x|2,1) + 1/2\phi(x|-1.5,1)$, while the bivariate bivariate true density has the following mean vectors and variance-covariance matrices.

$$\mu_1 = \begin{pmatrix} -1.5 \\ -1.5 \end{pmatrix},\ \Sigma_1 = \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix},\ \mu_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix},\ \Sigma_2 = \begin{pmatrix} 1 & -0.9 \\ -0.9 & 1 \end{pmatrix}.$$

Note that this bivariate density was used by Zhang et al. (2006).

Density B is a mixture of two normal densities with different weights but an equal height at the modes:

$$f_B(\boldsymbol{x}|\mu_1,\Sigma_1,\mu_2,\Sigma_2) = \frac{3}{4}\phi(\boldsymbol{x}|\mu_1,\Sigma_1) + \frac{1}{4}\phi(\boldsymbol{x}|\mu_2,\Sigma_2).$$

The univariate density is $f_{B_1}(x) = 3/4\phi(x|-1.5,1) + 1/4\phi(x|-1.5,1/9)$, which was discussed by Sain & Scott (1996). The bivariate density is the same mixture with mean

Figure 3.1: *Density graphs of target univariate densities.*

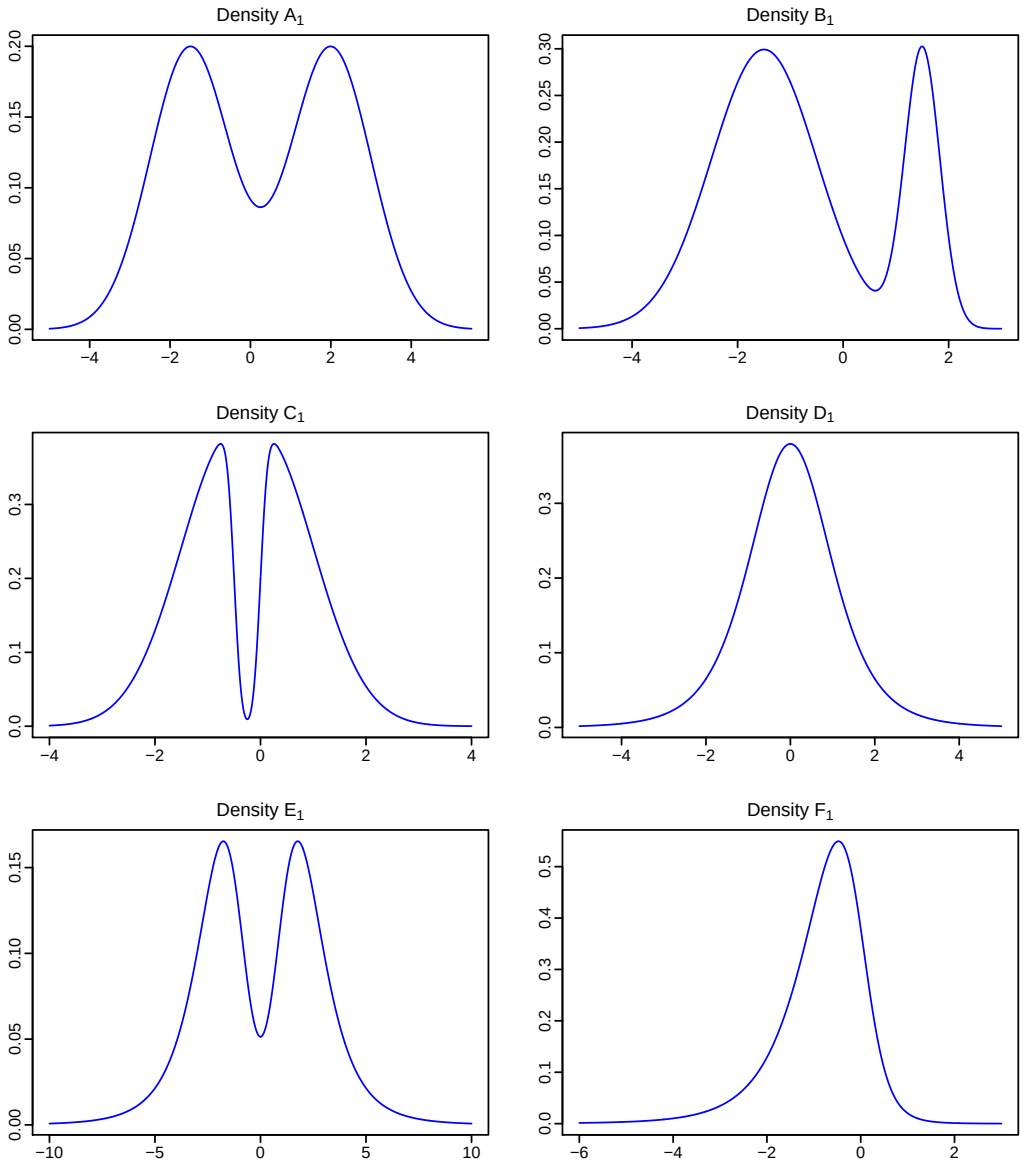
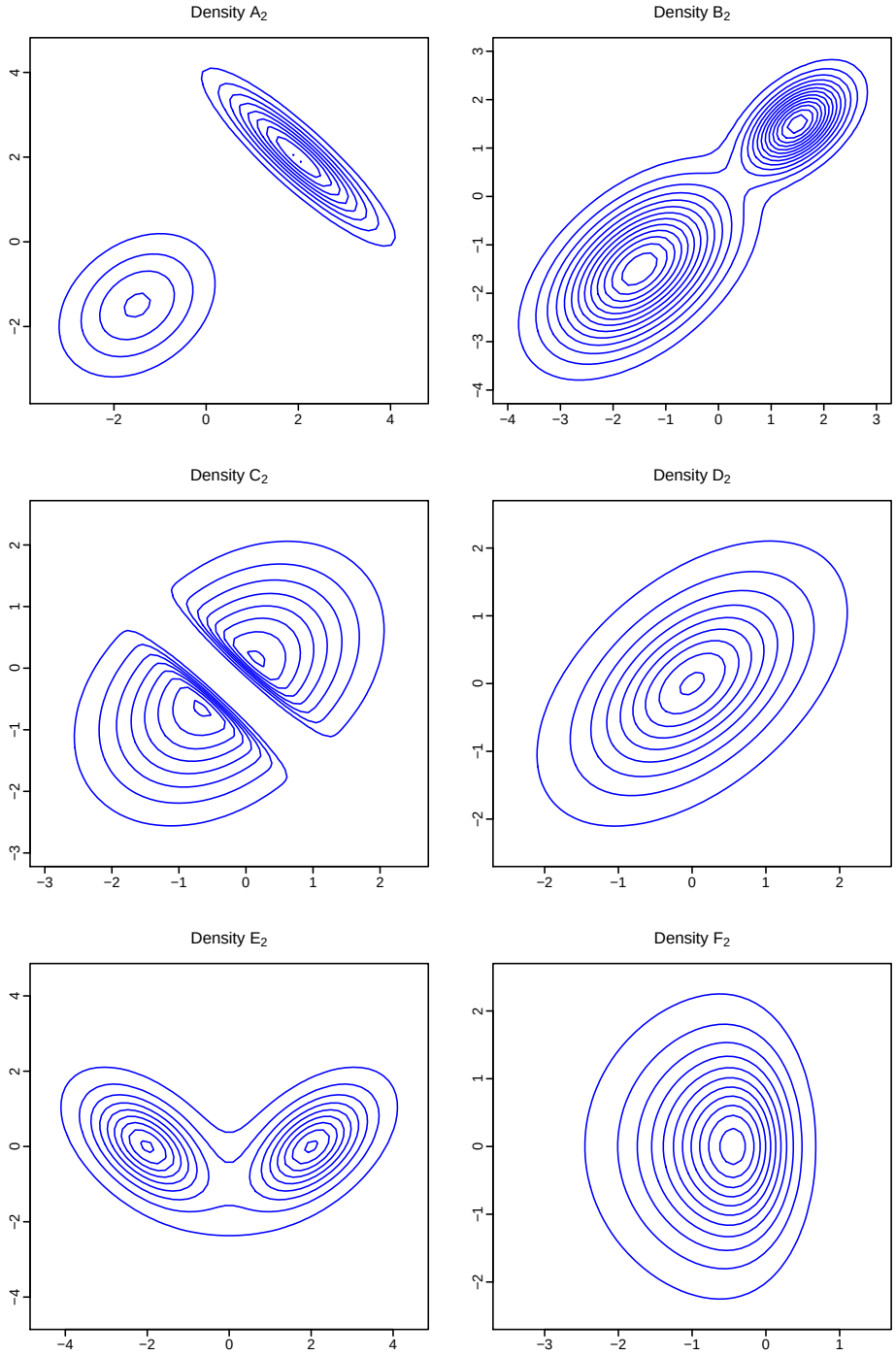


Figure 3.2: *Contour graphs of target bivariate densities.*



vectors and variance-covariance matrices given as follows.

$$\mu_1 = \begin{pmatrix} -1.5 \\ -1.5 \end{pmatrix}, \Sigma_1 = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix}, \mu_2 = \begin{pmatrix} 1.5 \\ 1.5 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 1/3 & 1/6 \\ 1/6 & 1/3 \end{pmatrix}.$$

Density C is a mixture of two skew-normal densities:

$$\begin{aligned} f_C(\boldsymbol{x}|\mu_1, \gamma_1, \mu_2, \gamma_2, \Sigma) =& 0.5 \times 2\phi(\boldsymbol{x}|\mu_1, \Sigma)\Phi(\gamma_1'(\boldsymbol{x} - \mu_1)) \\ &+ 0.5 \times 2\phi(\boldsymbol{x}|\mu_2, \Sigma)\Phi(\gamma_2'(\boldsymbol{x} - \mu_2)) \end{aligned}$$

where $\Phi(\cdot)$ is the cumulative density function of a multivariate standard normal distribution, and $\gamma_1, \gamma_2 \in \boldsymbol{R}^d$ are the shape parameters determining the skewness. This distribution was proposed by Azzalini & Valle (1996) and the conventional normal density can be obtained when $\gamma_1 = \gamma_2 = 0$. The univariate density f_{C_1} has the following parameter values: $\mu_1 = -0.5$, $\mu_2 = 0$, $\alpha_1 = -9$ and $\alpha_2 = 9$. The bivariate density has the following parameters values:

$$\mu_1 = \begin{pmatrix} -0.5 \\ -0.5 \end{pmatrix}, \alpha_1 = \begin{pmatrix} -9 \\ -9 \end{pmatrix}, \mu_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \alpha_2 = \begin{pmatrix} 9 \\ 9 \end{pmatrix}, \Sigma = \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}.$$

Density D is a Student t distribution denoted as $t_d(\boldsymbol{x}|\mu, \Sigma, \nu)$:

$$f_D(\boldsymbol{x}|\mu, \Sigma, \nu) = \frac{\Gamma((\nu + d)/2)}{(\nu\pi)^{d/2}\Gamma(\nu/2)|\Sigma|^{1/2}} \left[1 + \frac{1}{\nu}(\boldsymbol{x} - \mu)'\Sigma^{-1}(\boldsymbol{x} - \mu)\right]^{-(d+\nu)/2},$$

which has the location parameter μ, dispersion matrix Σ and degrees-of-freedom $\nu = 5$. The parameter vector of the univariate density $f_{D_1}(x)$ is $(0, 1, 5)'$, while bivariate

density $f_{D_2}(\boldsymbol{x})$ has the following parameters:

$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \ \Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}.$$

Density E is a mixture of two Student t densities with degrees of freedom $\nu = 5$:

$$f_E(\boldsymbol{x}|\mu_1, \mu_2, \Sigma, \nu) = 0.5\ t_d(\boldsymbol{x}|\mu_1, \Sigma_1, \nu) + 0.5\ t_d(\boldsymbol{x}|\mu_2, \Sigma_2, \nu).$$

The univariate density $f_{E_1}(x) = 0.5\ t_1(x|-2, 1, 5) + 0.5\ t_1(2, 1, 5)$, and the bivariate density $f_{E_2}(\boldsymbol{x}) = 0.5\ t_2(x|\mu_1, \Sigma_1, 5) + 0.5\ t_2(\mu_2, \Sigma_2, 5)$, where

$$\mu_1 = \begin{pmatrix} -2 \\ 0 \end{pmatrix}, \ \Sigma_1 = \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix}, \ \mu_2 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \ \Sigma_2 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}.$$

Density F is a skew-t density proposed by Azzalini & Capitanio (2003):

$$f_F(\boldsymbol{x}|\mu, \Sigma, \alpha, \nu) = 2\ t_d(\boldsymbol{x}|\mu, \Sigma, \nu) T_d(\tilde{\boldsymbol{x}}|\nu + d), \tag{3.3.2}$$

where

$$\tilde{\boldsymbol{x}} = \gamma' \omega^{-1} (\boldsymbol{x} - \mu) \left(\frac{\nu + d}{(\boldsymbol{x} - \mu)' \Sigma^{-1} (\boldsymbol{x} - \mu) + \nu} \right)^{1/2},$$

ω is a diagonal matrix with diagonal elements the same as those of Σ, and $T_d(\cdot|\nu + d)$ is the cumulative density of the Student t distribution with $\nu + d$ degrees of freedom. The density given by (3.3.2) is able to capture heavy tailed property with $\nu = 5$ and moderately skewness. The univariate density $f_{F_1}(x)$ has parameters $\mu = 0$, $\alpha = -2$ and

$\Sigma = 1$. The bivariate density $f_{F_2}(\boldsymbol{x})$ has the following parameters:

$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \ \gamma = \begin{pmatrix} -2 \\ 0 \end{pmatrix}, \ \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

The density graph of each of the six univariate densities is presented in Figure 3.1, while the contour plot of each of the six bivariate densities is given in Figure 3.2. We can find that these densities exhibit a variety of different distributional properties.

3.3.2 Accuracy of our Bayesian bandwidth estimation

We generated samples of sizes $n = 200, 500, 1000$ from each of the six univariate densities, as well as samples of sizes $n = 500, 1000, 2000$ from each of the six bivariate densities. The kernel function for estimating univariate densities was chosen to be the univariate standard Gaussian density known as the Gaussian kernel, and the product of univariate Gaussian kernels was used as the kernel function for estimating multivariate densities. The bandwidth matrix in estimating multivariate densities was chosen to be a diagonal matrix.

First, we estimated the diagonal bandwidth matrices for our proposed tail-adaptive kernel density estimator with $\alpha = 0.05$ and 0.1. Second, we consider the kernel density estimator with a global bandwidth (matrix), which was estimated through two existing selection or estimation methods, namely the NRR discussed by Scott (1992) and the Bayesian sampling technique presented by Zhang et al. (2006).

In terms of our proposed tail-adaptive density estimator used for each generated sample, we applied the random-walk Metropolis-Hastings algorithm to the update of all

Table 3.1: *MCMC results obtained based on a sample generated from density F*

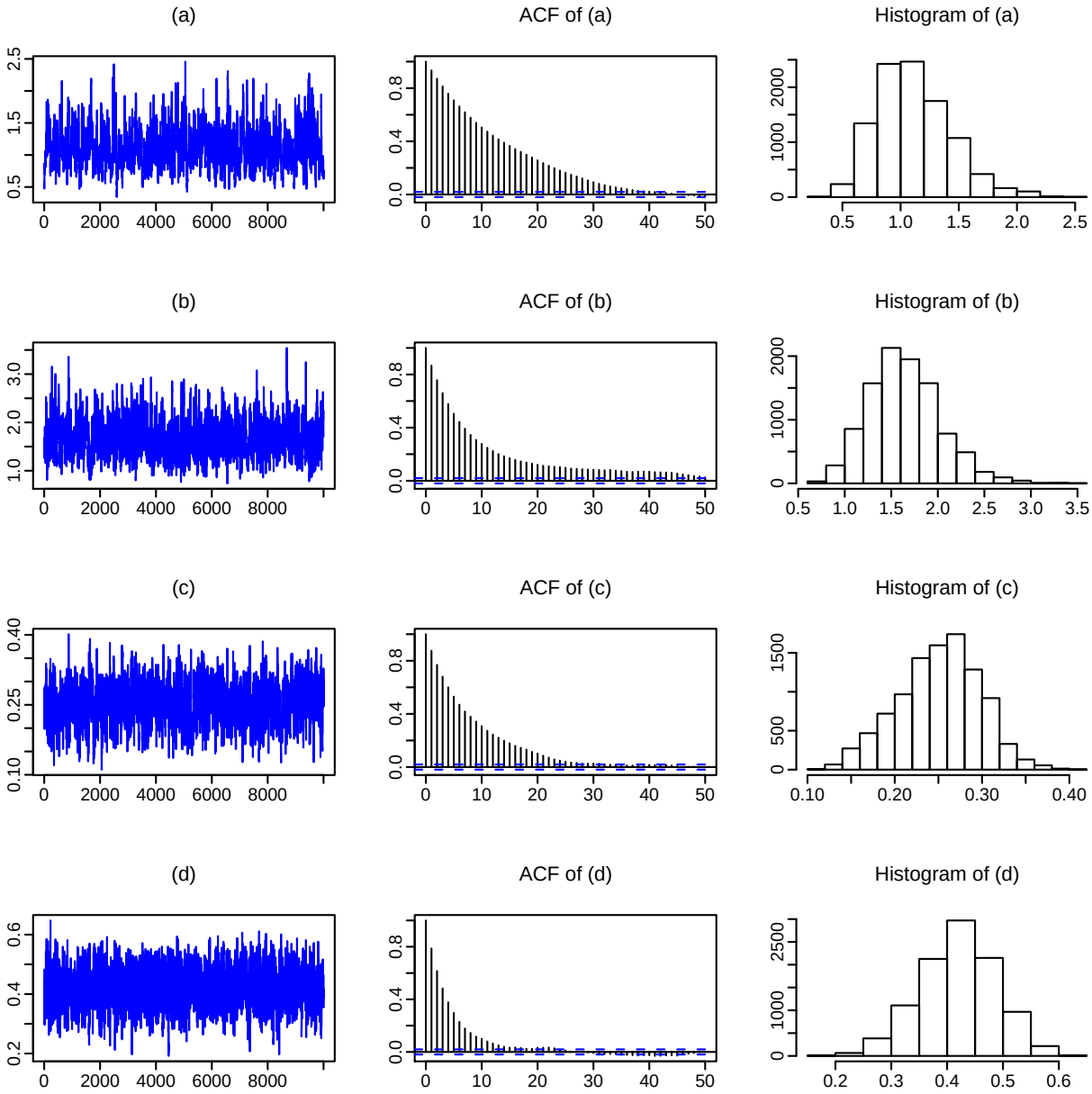
Bandwidths	Mean	Standard deviation	Batch-mean standard deviation	SIF	Acceptance rate
$h_1^{(1)}$	1.1121	0.3184	0.0157	24.32	0.28
$h_2^{(1)}$	1.6432	0.3816	0.0164	18.57	
$h_1^{(0)}$	0.2505	0.0469	0.0019	17.13	
$h_2^{(0)}$	0.4196	0.0675	0.0018	7.35	

bandwidths in the univariate situation (or all components of the bandwidth matrices in the bivariate situation) with the acceptance probability calculated through (3.2.3).

There are 3,000 iterations during the burn-in period, and the recorded period contains 10,000 iterations. We computed the batch-mean standard deviation discussed by Roberts (1996) and the simulation inefficient factor (SIF) discussed by Kim et al. (1998) to monitor the mixing performance (or loosely speaking, the convergence performance). Both indicators are explained in details in Zhang et al. (2006). As the simulated chain is a Markov chain, the SIF value can be roughly interpreted as the number of draws needed so as to produce independent draws. Therefore, a small SIF value usually indicate good mixing performance. In addition, a plot of the sample path of each parameter, together with its autocorrelation function (ACF) and histogram graphs is also presented for visual inspection of the mixing performance.

Consider a sample generated from $f_{F_2}(\boldsymbol{x})$ with the probability of the low-density region $\alpha = 0.05$ and sample size $n = 1000$. Figure 3.3 presents graphs of the sample path, its ACF and histogram of each bandwidth. Table 3.1 presents a summary of the MCMC results, in which we found that the SIF values are very small, and the batch-mean standard deviations are respectively, much smaller than their counterparts of

Figure 3.3: *Plots of posterior draws obtained through our proposed sampling algorithm for tail-adaptive bandwidths in kernel density estimation with α=0.05: (a) $h_1^{(1)}$; (b) $h_2^{(1)}$; (c) $h_1^{(0)}$; and (d) $h_2^{(0)}$.*



overall standard deviations. These indicators show that the mixing performance of the proposed sampling algorithm applied to the tail-adaptive kernel density estimator is very good and acceptable.

The estimates of bandwidths are also sensible. Note that f_{F_2} is a fat-tailed density with left skewness in one dimension and a certain degree of symmetry in the other dimension (see Figure 3.2). We found that the tail-adaptive density estimator clearly captures the fat-tailed feature of the true density. For example, the estimates of both components of $\boldsymbol{h}^{(1)}$ for observations inside the low-density region are respectively, much larger than the estimates of both components of $\boldsymbol{h}^{(0)}$ for observations outside this region.

In order to examine the performance of the proposed tail-adaptive density estimator with different bandwidth matrices assigned to the low- and high-density regions, we also derived global bandwidths (or bandwidth matrices for the bivariate situation) through the NRR and the Bayesian sampling method. However, we do not report the estimated bandwidths, but the resulting Kullback-Leibler information.

We generated N=100,000 random numbers (or vectors for the bivariate situation) from the true density and calculated the estimated Kullback-Leibler information defined by (3.3.1). For the six univariate densities, Table 3.2 presents the estimated Kullback-Leibler information between the true density and each density estimator resulted from each bandwidth estimation method. Among all six densities considered, the tail-adaptive density estimator with bandwidths estimated through Bayesian sampling and low-density probability 0.05 clearly performs better than the global-bandwidth estimator with bandwidth selected through NRR; and the former

Table 3.2: *Estimated Kullback-Leibler information for univariate densities*

		Kullback-Leibler information			
		Global-bandwidth		Tail-adaptive bandwidths	
Density	n	NRR	Bayesian	$\alpha = 0.05$	$\alpha = 0.10$
f_{A_1}	200	0.0374	0.0238	0.0311	0.0388
	500	0.0127	0.0070	0.0070	0.0069
	1000	0.0091	0.0033	0.0031	0.0032
f_{B_1}	200	0.1137	0.0506	0.0399	0.0371
	500	0.0545	0.0134	0.0157	0.0181
	1000	0.0368	0.0136	0.0126	0.0105
f_{C_1}	200	0.2094	0.0837	0.0738	0.0781
	500	0.0688	0.0567	0.0332	0.0349
	1000	0.0478	0.0246	0.0161	0.0142
f_{D_1}	200	0.0322	0.0602	0.0280	0.0340
	500	0.0170	0.0457	0.0210	0.0230
	1000	0.0118	0.0285	0.0139	0.0152
f_{E_1}	200	0.0974	0.1019	0.0445	0.0377
	500	0.0491	0.0536	0.0336	0.0273
	1000	0.0283	0.0256	0.0117	0.0123
f_{F_1}	200	0.0670	0.0695	0.0364	0.0401
	500	0.0578	0.0798	0.0282	0.0355
	1000	0.0143	0.0153	0.0091	0.0102

clearly performs better than the global-bandwidth estimator with bandwidth estimated through Bayesian sampling except Density A_1. When the Bayesian estimation of a global bandwidth performs worse than the NRR of a global bandwidth for Densities D_1 to F_1, our proposed Bayesian estimation of tail-adaptive bandwidths outperforms the NRR. Table 3.2 also shows that there is no obvious difference between different choices of α, which is the probability of the low-density region.

Table 3.3: *Estimated Kullback-Leibler information for bivariate densities*

		Global-bandwidth		Tail-adaptive bandwidth	
Density	n	NRR	Bayesian	$\alpha = 0.05$	$\alpha = 0.10$
f_A	500	0.2878	0.0858	0.0772	0.0748
	1000	0.2382	0.0617	0.0498	0.0467
	2000	0.1981	0.0402	0.0339	0.0338
f_B	500	0.1201	0.0499	0.0444	0.0442
	1000	0.0826	0.0349	0.0332	0.0337
	2000	0.0653	0.0256	0.0219	0.0217
f_C	500	0.1126	0.0930	0.0783	0.0768
	1000	0.0924	0.0689	0.0559	0.0558
	2000	0.0900	0.0648	0.0497	0.0498
f_D	500	0.1171	0.0946	0.0464	0.0449
	1000	0.0809	0.0769	0.0286	0.0312
	2000	0.0590	0.0565	0.0242	0.0270
f_E	500	0.1436	0.1072	0.0623	0.0530
	1000	0.1038	0.1088	0.0328	0.0397
	2000	0.0782	0.0666	0.0262	0.0282
f_F	500	0.1169	0.1641	0.0520	0.0545
	1000	0.0781	0.0657	0.0261	0.0306
	2000	0.0708	0.0637	0.0237	0.0242

The estimated Kullback-Leibler information for bivariate densities is given in Table 3.3. Among all six densities considered, the tail-adaptive density estimator obviously performs better than global-bandwidth density estimator with bandwidth matrix estimated through either the NRR or Bayesian sampling. Note that Bayesian estimation of a global bandwidth matrix performs slightly worse than NRR in the case of f_{F_2} with sample size 500, our proposed Bayesian estimation of tail-adaptive bandwidth performs clearly better than the two competitors. The results also indicate that the performance of the tail-adaptive density estimator is not very sensitive to different values of the probability of low-density region.

The mean integrated squared error (MISE) was also used to examine the performance of tail-adaptive density estimator. We numerically approximate the MISE through 200 data sets for each the bivariate densities with sample size 500, 1000 and 2000.

Table 3.4: *Estimated MISE (×100) for bivariate densities*

		Global-bandwidth		Tail-adaptive bandwidth	
Density	n	NRR	Bayesian	$\alpha = 0.05$	$\alpha = 0.10$
f_A	500	1.8482	0.4945	0.4415	0.4401
	1000	1.5546	0.3354	0.2920	0.2901
	2000	1.2875	0.2230	0.1875	0.1854
f_B	500	0.6526	0.2718	0.2612	0.2606
	1000	0.4927	0.1828	0.1738	0.1727
	2000	0.3595	0.1164	0.1101	0.1089
f_C	500	0.8812	0.6022	0.5422	0.5422
	1000	0.7059	0.4208	0.3694	0.3804
	2000	0.5589	0.2868	0.2600	0.2692
f_D	500	0.3250	0.6573	0.2782	0.2638
	1000	0.2152	0.5121	0.1796	0.1737
	2000	0.1489	0.3915	0.1219	0.1245
f_E	500	0.4022	0.3722	0.1919	0.1822
	1000	0.3039	0.2980	0.1279	0.1242
	2000	0.2207	0.2283	0.0840	0.0844
f_F	500	0.3331	0.7560	0.3092	0.2966
	1000	0.2229	0.6150	0.2054	0.2022
	2000	0.1496	0.4856	0.1386	0.1437

Table 3.4 shows that the lower-adaptive estimator always outperforms the global-bandwidth estimator.

3.4 Tail-adaptive density estimation for high dimensions

Our proposed Bayesian sampling algorithm for estimating bandwidths (or bandwidth matrices in multivariate situations) in tail-adaptive kernel density estimation is applicable to data of any dimension. In this section, we aim to examine the performance of the tail-adaptive estimator with bandwidth matrices estimated through Bayesian sampling in comparison with its two competitors, namely the NRR and Bayesian estimation of a global bandwidth matrix proposed by Zhang et al. (2006).

3.4.1 True densities

We consider four target densities labeled G, H, I and J. Density G is a mixture of two multivariate normal densities:

$$f_G(\boldsymbol{x}|\mu_1,\mu_2,\Sigma_1,\Sigma_2) = \frac{1}{2}\phi(\boldsymbol{x}|\mu_1,\Sigma_1) + \frac{1}{2}\phi(\boldsymbol{x}|\mu_2,\Sigma_2),$$

with location parameter vectors specified as $\mu_1 = (-1.5,-1.5,-1.5,-1.5,-1.5)'$ and $\mu_2 = (2,2,2,2,2)'$ and both variance-covariance matrices of the form

$$\Sigma = \frac{1}{1-\rho^2}\begin{pmatrix} 1 & \rho & \rho^2 & \rho^3 & \rho^4 \\ \rho & 1 & \rho & \rho^2 & \rho^3 \\ \rho^2 & \rho & 1 & \rho & \rho^2 \\ \rho^3 & \rho^2 & \rho & 1 & \rho \\ \rho^4 & \rho^3 & \rho^2 & \rho & 1 \end{pmatrix}, \tag{3.4.1}$$

where $\rho = 0.3$ for Σ_1 and $\rho = -0.9$ for Σ_2.

Density H is a multivariate skew-normal densities:

$$f_H(\boldsymbol{x}|\mu,\Sigma,\alpha) = 2\phi(\boldsymbol{x}|\mu,\Sigma)\Phi(\gamma'(\boldsymbol{x}-\mu)),$$

where Σ is defined by (3.4.1) with $\rho = 0.9$, $\mu = (-0.5,-0.5,-0.5,-0.5,-0.5)'$, $\Phi(\cdot)$ is the standard normal cumulative density, and the skewness parameter vector $\gamma = (-9,-9,-9,-9,-9)'$.

Density I is a mixture of two multivariate Student t densities:

$$f_I(\boldsymbol{x}|\mu_1,\mu_2,\Sigma_1,\Sigma_2,\nu) = 0.5\ t_d(\boldsymbol{x}|\mu_1,\Sigma_1,\nu) + 0.5\ t_d(\boldsymbol{x}|\mu_2,\Sigma_2,\nu),$$

where $\mu_1 = (-2, 0, -2, 0, -2)'$, $\mu_2 = (2, 0, 2, 0, 2)'$, $\nu = 5$, and both Σ_1 and Σ_1 are defined by (3.4.1) with $\rho = -0.5$ and $\rho = 0.5$, respectively.

Density J is a multivariate skew-t densities:

$$f_J(\boldsymbol{x}|\mu, \Sigma, \alpha, \nu) = 2t_d(\boldsymbol{x}|\mu, \Sigma, \nu)\, T_d(\tilde{\boldsymbol{x}}|\nu + d),$$

where $\mu = \mathbf{0}$, $\nu = 5$, Σ is a $d \times d$ identity matrix, and $\tilde{\boldsymbol{x}}$ is defined by (3.3.2) with $\gamma = (2, 0, 2, 0, 2)'$.

3.4.2 Accuracy of our Bayesian bandwidth estimation

We generated samples of sizes $n = 500, 1000, 2000$ from each of the five-dimensional densities.

We calculated Kullback-Leibler information between the true density and its estimator resulted from each of the three bandwidth estimation methods. We note that MISE was not presented in this case. MISE is extremely time consuming to compute, and in the bivariate case shown above, the MISE results are very consistent with the Kullback-Leibler information. Hence we believe calculating MISE again in this case is not necessary.

Table 3.5 presents the estimated Kullback-Leibler information between the true density and its estimator resulted from each of the three bandwidth estimation methods. We found that our proposed Bayesian estimation of the tail-adaptive bandwidth matrix obviously outperforms the NRR for choosing a global bandwidth matrix in kernel density estimation. Moreover, we found that the former clearly performs better than

Table 3.5: *Estimated Kullback-Leibler information for 5-dimensional densities*

		Global-bandwidth		Tail-adaptive bandwidth	
Density	n	NRR	Bayesian	$\alpha = 0.05$	$\alpha = 0.10$
f_G	500	0.8923	0.4280	0.4026	0.4004
	1000	0.7705	0.3093	0.2848	0.2825
	2000	0.6933	0.2489	0.2343	0.2300
f_H	500	0.4559	0.3438	0.3212	0.3179
	1000	0.4041	0.2892	0.2613	0.2582
	2000	0.3355	0.2226	0.2033	0.1987
f_I	500	0.5943	0.5674	0.3446	0.3187
	1000	0.4994	0.4814	0.2891	0.2666
	2000	0.4395	0.4255	0.2274	0.2072
f_J	500	0.6107	0.5755	0.3226	0.3033
	1000	0.5969	0.4415	0.2538	0.2284
	2000	0.5050	0.3937	0.1971	0.1773

Bayesian estimation of a global bandwidth matrix. These findings are consistent with what we found in the bivariate situation.

For all sample sizes of each density considered, we found that the tail-adaptive kernel density estimator with $\alpha = 0.1$ slightly outperforms the same estimator with $\alpha = 0.05$. However, we would be reluctant to make a decision as to whether the former performs better than the latter because such a difference resulted from the two different probability values is marginal.

3.5 An application of the tail-adaptive density estimator

In this section, we apply the proposed tail-adaptive kernel density estimator to the estimation of bivariate density of stock-index returns. We obtained the daily closing index values of the S&P500 index in the U.S. stock market and the All Ordinaries

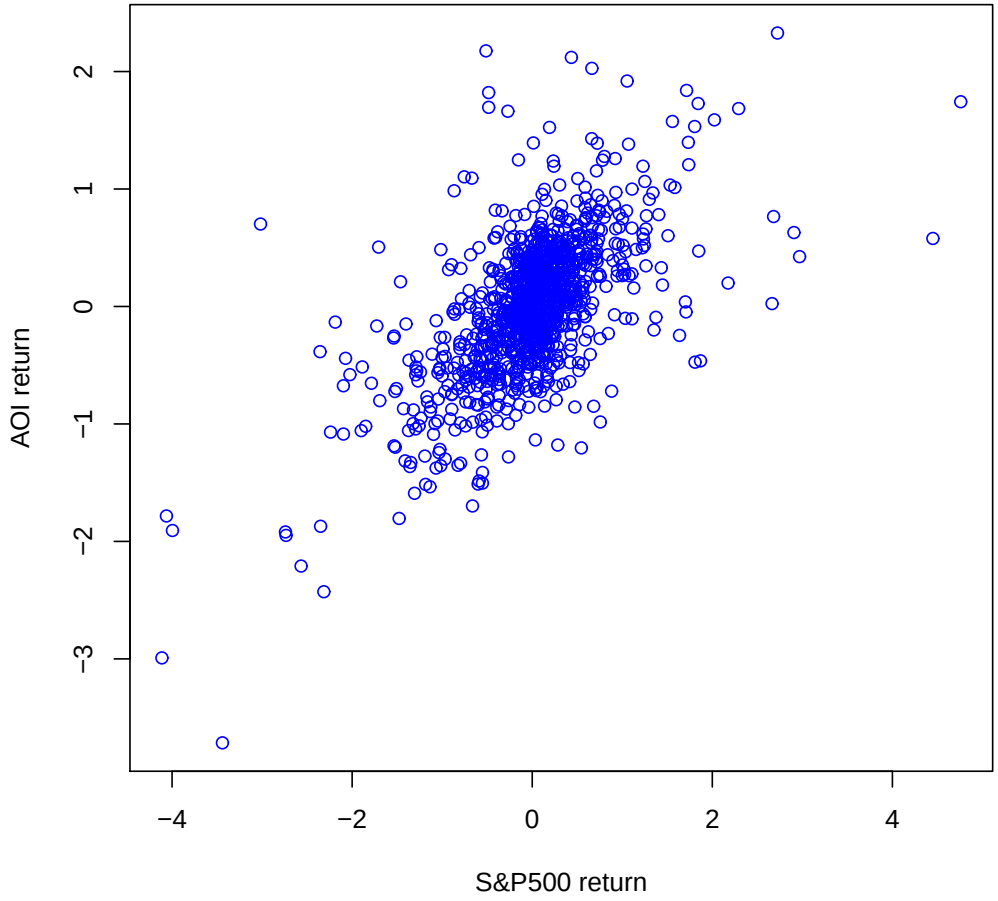
Table 3.6: *Descriptive statistics of the daily continuously compounded returns of the S&P500 index and All Ordinaries (AORD)*

Series	n	Mean	Standard deviation	Skewness	Kurtosis	Correlation
S&P500	1155	-0.0058	0.7034	-0.2197	11.1613	0.6171
AORD	1155	0.0015	0.5779	-0.3955	6.4593	

(AORD) in the Australian stock market, where the sample period is from the 2nd January 2006 to the 16th September 2010 excluding non-trading days. The AORD continuously compounded return was matched to the overnight S&P500 return. In financial economics, most researchers believe that the density of financial asset returns has a higher peak and heavier tails than the normal density. If a global bandwidth is used for kernel density estimation, the use of a global bandwidth is likely to over-smooth the density due to the existence of observations in the tail areas. The use of complete-adaptive bandwidths may not be attractive in applications due to the large number of bandwidth parameters. Therefore, we wish to apply the tail-adaptive kernel density estimator to the estimation of bivariate-return density.

Let x_t denote the closing index at date t. The daily continuously compounded returns in percentage form was computed as $(\ln x_t - \ln x_{t-1}) \times 100$. The sample size is $n = 1155$. The sample period covers the period of current global financial crisis, where there are some extreme observations. Table 3.6 presents some basic descriptive statistics. We found that both return series have mean values around zero, a certain degree of negative skewness and excessive kurtosis. As shown in the scatter plot of the bivariate observations given in Figure 3.4, the daily returns of both indices are correlated with the Pearson correlation coefficient 0.6171. We can visually identify many extreme

Figure 3.4: *A scatter plot of daily continuously compounded daily returns of S&P500 and AORD in percentage form during the period from the 2nd January 2006 to 16th September 2010*



return values in Figure 3.4, which indicates that the joint density of the bivariate index returns has very heavy tails during the sample period.

We used our Bayesian sampling algorithm to estimate bandwidths matrices for the tail-adaptive kernel density estimator of the bivariate index returns, where the probability of low-density region was chosen to be 5%. We also applied the Bayesian sampling algorithm proposed by Zhang et al. (2006) and NRR to the estimation of global bandwidth matrix for the kernel estimation of the bivariate return density.

There were 3,000 iterations in burn-in period and 10,000 iterations in the recorded period for both sampling algorithms. Table 3.7 presents a summary of the results, where the batch-mean standard deviation and SIF measures indicate very good mixing performance of both samplers. Moreover, we calculated the log marginal likelihood of Newton & Raftery (1994) for each of the two density estimators so as to decide which is favored against the other. The log marginal likelihood for our tail-adaptive kernel density estimator is -1657.14, which is obviously larger than -1719.64, the log marginal likelihood for the global-bandwidth kernel density estimator. Thus, we have found strong evidence supporting our tail-adaptive density estimator against the global-bandwidth density estimator.

With the estimated tail-adaptive bandwidth matrices given in the 3rd column of Table 3.7, we calculated the tail-adaptive density estimator of the bivariate index returns, whose density surface and contour graph presented in the 1st row of Figure 3.5. Moreover, the 2nd row of Figure 3.5 presents the same set of graphs produced by the global bandwidth matrix estimated via the Bayesian sampling algorithm of Zhang et al. (2006). The last row of Figure 3.5 presents the same set of graphs produced by

Figure 3.5: *Surface graphs and contour plots of the three density estimators produced by (a) tail-adaptive bandwidths with $\alpha = 5\%$; (b) Bayesian global bandwidth; and (c) NRR bandwidth. In each surface graph, the x-axis represents return in percentage, and the y-axis represents density. In each contours plot, both axises represent return in percentage.*

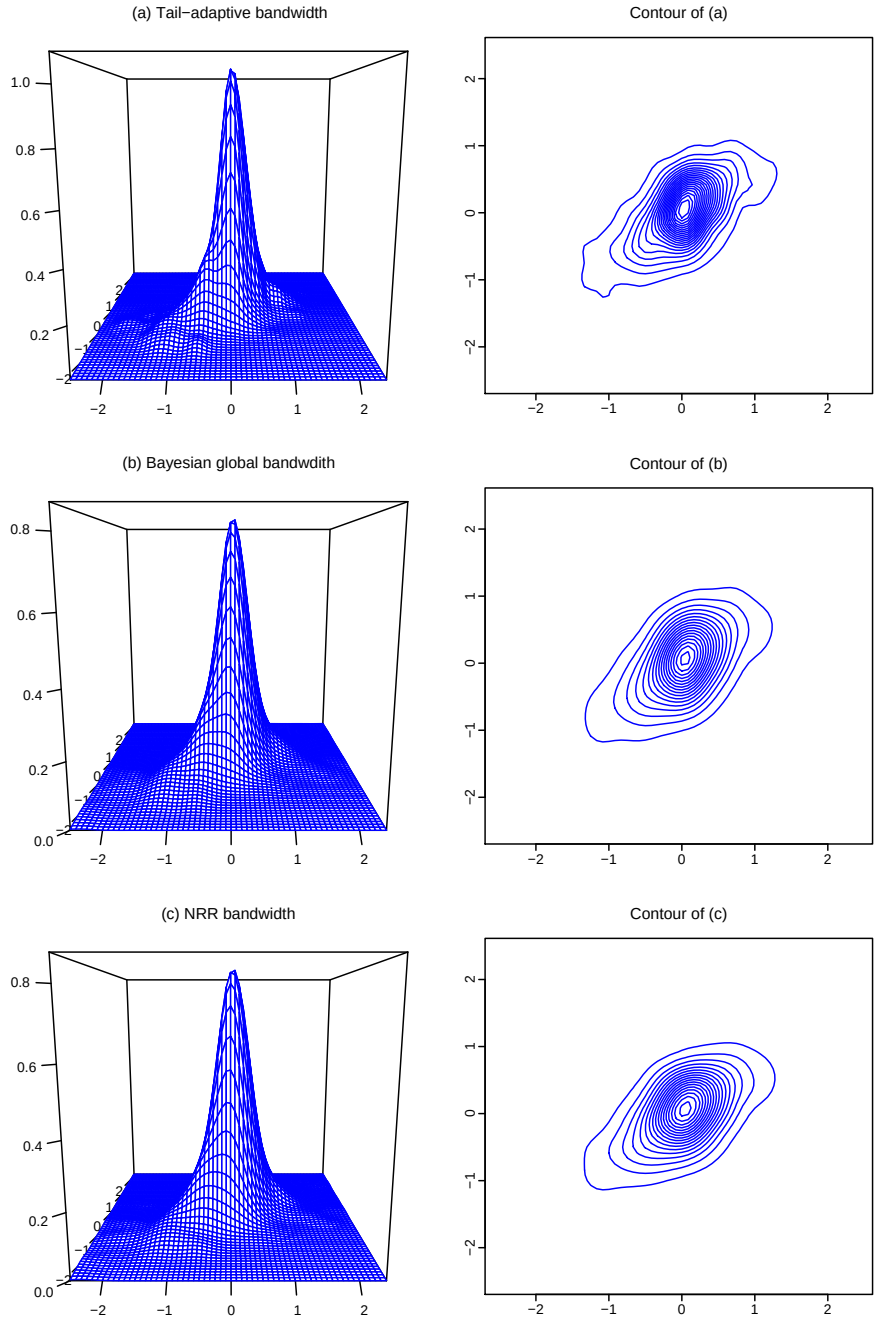


Table 3.7: *A summary of MCMC results obtained through our proposed Bayesian sampling algorithm to the tail-adaptive kernel density estimator of the S&P500 and AORD returns*

	Bandwidths	Mean	Standard deviation	SIF	Acceptance rate	log marginal likelihood
NRR	h_1	0.2171				
	h_2	0.1783				
Bayesian global bandwidth	h_1	0.1795	0.0113	5.63	0.21	-1719.64
	h_2	0.2485	0.0121	5.89		
Tail-adaptive bandwidth with $\alpha = 0.05$	$h_1^{(1)}$	0.5533	0.2217	39.30	0.27	-1657.14
	$h_2^{(1)}$	0.5552	0.1140	19.97		
	$h_1^{(0)}$	0.1221	0.0161	15.39		
	$h_2^{(0)}$	0.1547	0.0174	13.55		

the global bandwidth matrix estimated via NRR. Both the density surface and the contour produced via the tail-adaptive estimator is obviously different from those produced via each global-bandwidth density estimator. Both the density surface of contour plot of the tail-adaptive density estimator show that this estimator captures richer dynamics than the other two density estimators.

Let x_t denote the S&P500 return and y_t the AORD return. We used the bandwidth matrices estimated through our tail-adaptive density estimator to estimate the conditional density of AORD return given that the S&P500 return equals a certain value. Such a conditional density is expressed as

$$f(y|x_t = x) = \frac{f(y,x)}{f_x(x)},$$

where $f(y,x)$ is the joint density of (y_t, x_t), and $f_x(x)$ is the marginal density of x_t. According to Holmes et al. (2010) and Polak et al. (2010), bandwidths estimated through a joint density can also be used for the purpose to compute conditional density. As

market analysts are often concerned with the left tail of the density of stock-index returns, we computed the conditional density of AORD returns given that the S&P500 return is at each of the quantiles of 10%, 7.5% 5%, 2.5%, 1% and 0.5%, which are corresponding to percentage return values of -0.73, -0.89, -1.13, -1.52, -2.24 and -2.74, respectively. The graph of each conditional density is presented in the 1st columns of Figure 3.6 and Figure 3.7, from which we can visually understand the distributional properties of the AORD return given that the U.S. stock market finished daily trading with the S&P500 index return at a certain value.

With the tail-adaptive bandwidth matrices estimated via our Bayesian sampling algorithm, we are able to estimate the conditional probability of the form

$$\Pr\{y_t \leq y | x_t \leq x\} = \frac{\Pr\{y_t \leq y, x_t \leq x\}}{\Pr\{x_t \leq x\}}. \tag{3.5.1}$$

Such a calculation can be done simply by replacing the Gaussian kernel with its cumulative density function. The interpretation of (3.5.1) is also clear and meaningful to market analysts. Given that the U.S. stock market went down beyond x%, the probability that the Australian stock market would drop beyond y% is approximated through (3.5.1). We found that $\Pr\{y_t \leq 0 | x_t \leq 0\} = 0.67$. It means that when the U.S. stock market finished daily trading with a negative return, there was a 67% chance that the Australian stock market would also drop. Given that such a chance is more than 50%, we could say that the Australian stock market followed the U.S. stock market during the global financial crisis.

Figure 3.6: *Each graph in the left column represents the conditional density given that the S&P500 return is at the chosen value. Each graph in right column represents the conditional CDF computed through (3.5.2) at different y values for a given x value marked by the vertical line, while the horizontal line marks the y value that is the same as the chosen x value.*

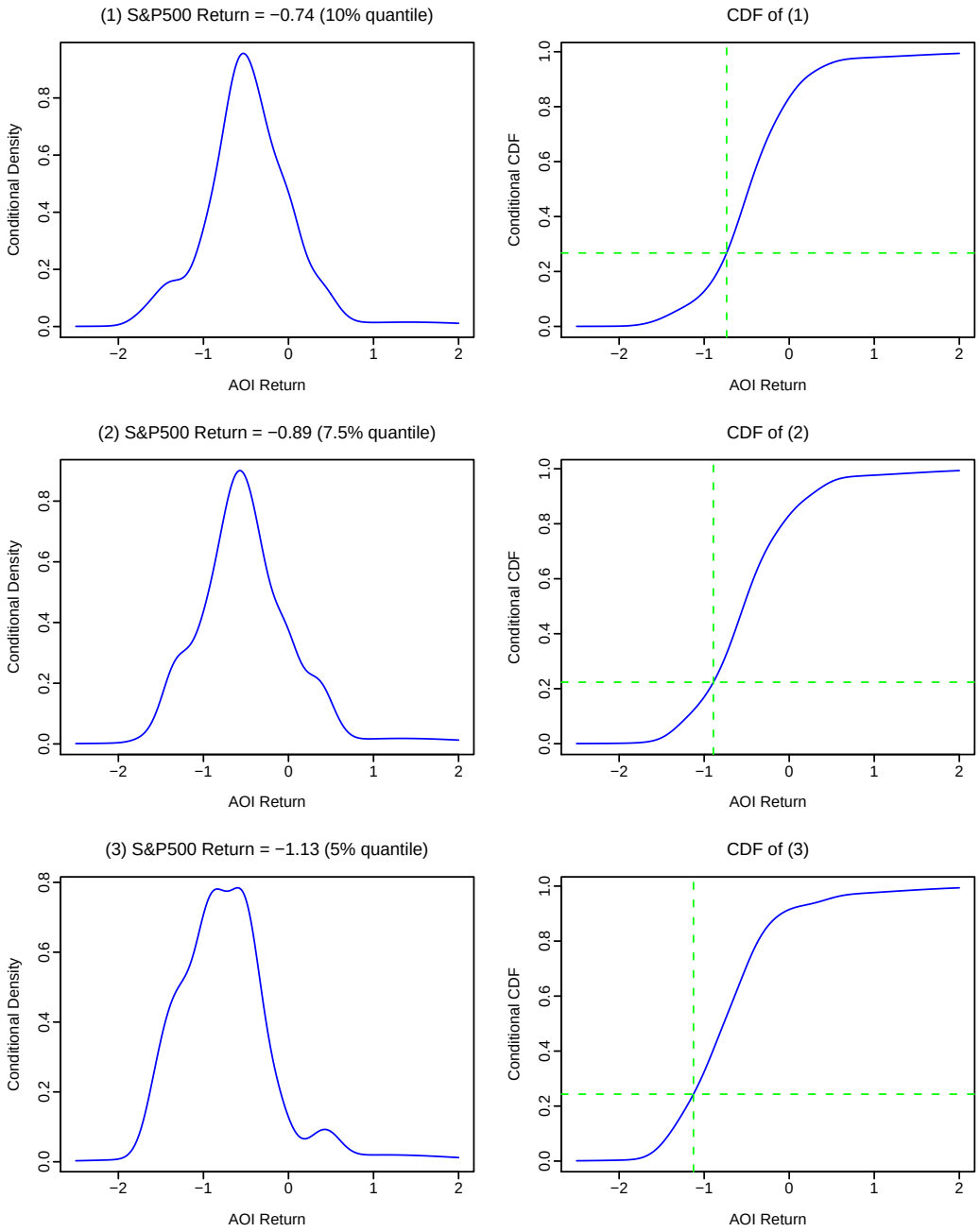
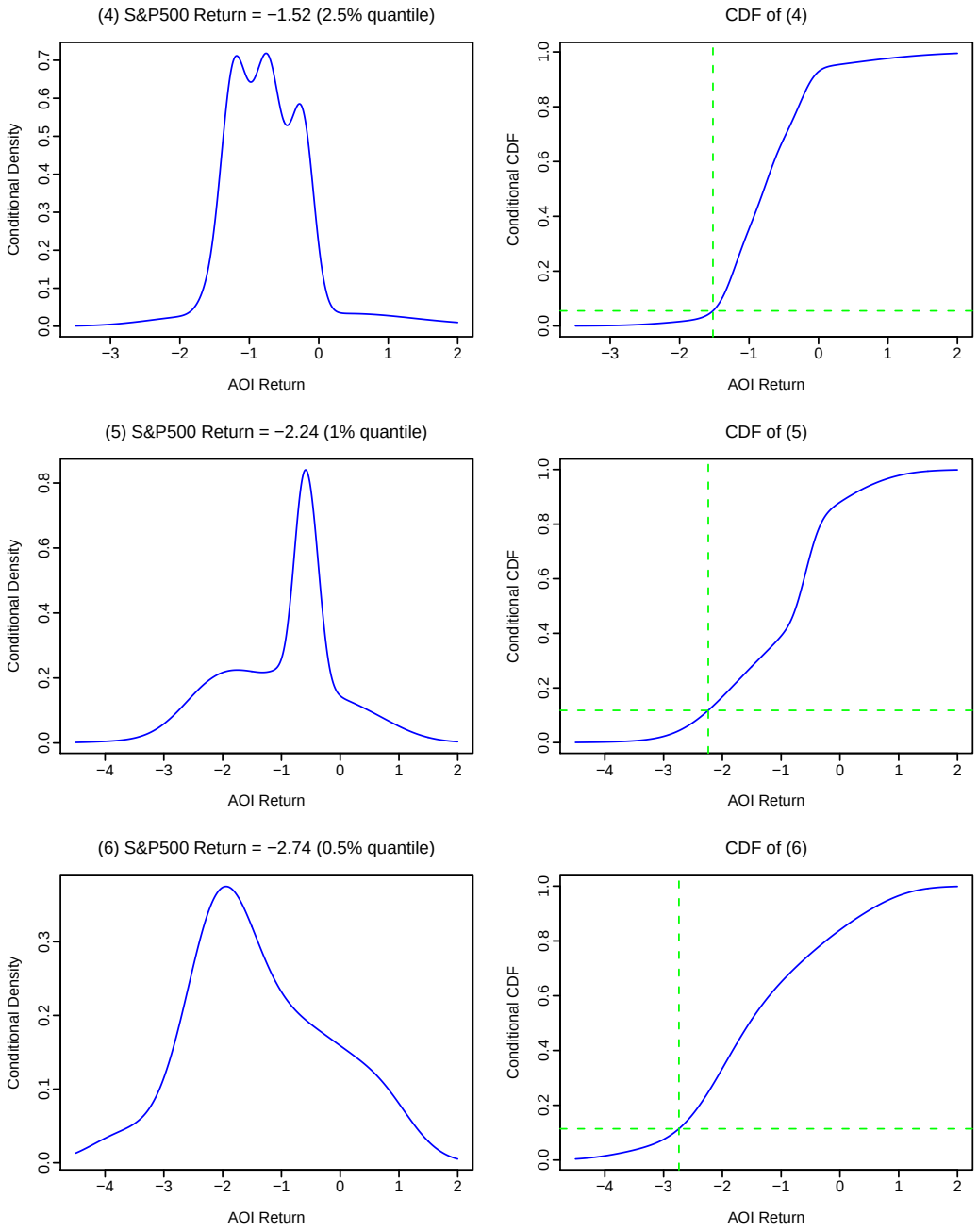


Figure 3.7: *Each graph in the left column represents the conditional density given that the S&P500 return is at the chosen value. Each graph in right column represents the conditional CDF computed through (3.5.2) at different y values for a given x value marked by the vertical line, while the horizontal line marks the y value that is the same as the chosen x value.*



With the tail-adaptive kernel density estimator estimated through our Bayesian sampling algorithm, we are able to estimate the conditional cumulative density function (CDF) of y_t for given $x_t = x$:

$$F(y|x_t = x) = \Pr\{y_t \leq y|x_t = x\} = \int_{-\infty}^{y} \frac{f(z,x)}{f_x(x)} dz. \tag{3.5.2}$$

The conditional CDF was estimated in the same way as we estimated $f(y|x_t = x)$ with the Gaussian kernel function for y_t replaced with the Gaussian CDF function. The interpretation of (3.5.2) is clear and meaningful to market analysts. Given that the U.S. stock market finished daily trading with the S&P500 index return being at x%, the probability that the Australian stock market drops beyond the same daily return level is indicated by (3.5.2).

We used the above-mentioned quantiles of the S&P500 return and derived the conditional CDF values as follows.

$$\Pr\{y_t \leq -0.73|x_t = -0.73\} = 0.27, \quad \Pr\{y_t \leq -0.89|x_t = -0.89\} = 0.22,$$
$$\Pr\{y_t \leq -1.13|x_t = -1.13\} = 0.24, \quad \Pr\{y_t \leq -1.52|x_t = -1.52\} = 0.06,$$
$$\Pr\{y_t \leq -2.24|x_t = -2.24\} = 0.12, \quad \Pr\{y_t \leq -2.74|x_t = -2.74\} = 0.11.$$

The interpretation of these values is clear. Even though the Australian stock market followed the U.S. stock market during the global financial crisis, the probability that the Australian market had a larger drop than the U.S. market was at most 27%.

Each graph in the second columns of Figures 3.6 and 3.7 plots the curve of the conditional CDF function of y_t given that x_t takes each of the above six values. With these

graphs, we are able to approximate different probability values implied by (3.5.2) for different values of y. For example, we computed the one-day VaR of the AORD return conditional on the overnight U.S. market behavior. Given the S&P500 at each of the quantiles of 10%, 7.5% 5%, 2.5%, 1% and 0.5%, the one-day VaRs at the 95% confidence level for every \$100 investment on the AORD are respectively, \$1.3668, \$1.3620, \$1.5421, \$1.5428, \$2.6800 and \$3.2810. Moreover, conditional on the observed S&P500 return on the 17th September 2010, which is the next day out of the sample, we calculated that the one-day VaR at the 95% confidence level for every \$100 investment on the AORD is \$0.5326. Thus, this type of graphs is useful for us to understand the distributional behavior of the All Ordinaries return in the Australian stock market conditional on the overnight S&P500 return in the U.S. stock market.

Moreover, we evaluated the performance of the VaR computed by the proposed method to the VaR obtained through IGARCH (Riskmetrics) model. We followed the steps described in Bao et al. (2006) and calculated the check function of Koenker & Bassett Jr (1978). The existing data period is used as the learning set and we forecasted daily 5% VaR of AORD from 17th September 2010 to 5th May 2011.[1] The check function computed by IGARCH and our proposed method are 0.0356 and 0.0344, respectively. Lower value of check function indicates that the VaR produced based on overnight S&P500 return could be more effective than the conventional VaR computed from previous market information of AORD itself. Future research could be carried out in this direction.

[1] Please see Bao et al. (2006) for a detailed discussion.

3.6 Conclusion

This chapter proposes a kernel density estimator with tail-adaptive bandwidths, which are assigned to the low- and high-density regions, respectively. We have derived the posterior of bandwidth parameters based on Kullback-Leibler information and presented an MCMC sampling algorithm to estimate bandwidths. The Monte Carlo simulation study shows that the kernel density estimator with tail-adaptive bandwidths estimated through our proposed Bayesian sampling algorithm outperforms its competitor, the kernel density estimator with a global bandwidth estimated through either the normal reference rule discussed in Scott (1992) or the Bayesian sampling algorithm proposed by Zhang et al. (2006). The simulation result also shows that the improvement made by the tail-adaptive kernel density estimator is especially obvious when the underlying density is fat-tailed. Even though the probability of the low-density region α has to be chosen before we carry out the sampling procedure, we have found that performance the low-density adaptive kernel estimator is not sensitive to the changes of such probability values. Therefore, it is the users' choice on what the probability of the low-density region should be. Future study could include such a probability value as an additional parameter to be estimated through the sampling procedure.

We applied the tail-adaptive kernel density estimator to the estimation of bivariate density of the paired daily returns of the Australian Ordinary index and S&P500 index during the period of global financial crisis. The tail-adaptive density estimator captures richer dynamics in the tail area than the density estimator with a global

bandwidth estimated through the normal reference rule and a Bayesian sampling algorithm. With the tail-adaptive bandwidths estimated through our proposed Bayesian sampling algorithm, we have derived the estimated conditional density and distribution of the Australian index return given that the U.S. market finished daily trading with different return values. We have found that during the global financial crisis, even though the Australian stock market followed the U.S. stock market, there was no more than 27% chance that the former market had a larger drop than the latter. The graphs of the conditional density and distribution enable market analysts to approximate various probability values conditional on the behavior of the U.S. stock market.

By estimating a separate bandwidth for data in high density region, our method can also be viewed from mode estimation in clustering analysis. However, even our algorithm can be implemented to data in any dimension, the curse of dimensionality is a significant limitation of kernel density estimation itself. A recent study by Ferraty & Vieu (2006) suggested that by introducing some suitable proximity measure between data inside the kernel, this problem can be attacked from a functional setting even for data with infinite dimension. This is a significant discovery in clustering analysis. Therefore, future researches shall be done for possible extension of the proposed method to mode-estimation in high dimensional data.

Chapter 4

Bayesian Adaptive Kernel Density Estimation for Multimodal Distributions

4.1 Introduction

The performance of global bandwidth kernel density estimator is limited when the underlying distribution is multimodal. In this chapter, we propose to remedy this problem with a method called the cluster-adaptive kernel density estimator, which assigns a different bandwidth to sample data in each cluster.

This chapter is organized as follows. In Section 4.2, the study derives the posterior distribution of the cluster-adaptive density estimator bandwidth parameters and describes an MCMC sampling algorithm for estimating the bandwidths. Sections 4.3 and 4.4 present the results of Monte Carlo simulation studies designed to examine the

performance of the mode-adaptive density estimator. In these experiments we consider the issue of bandwidth estimation for univariate, bivariate and 3-dimensional multivariate density estimation using densities designed to have multiple modes with skewness and heavy tails. To demonstrate the efficaciousness of our proposed technique we compare the performance of the tail-adaptive density estimator with the Bayesian global bandwidth estimator and the NRR bandwidth procedure. The results indicate that. In Section 4.5 we apply the mode-adaptive density estimator to the estimation. Section 4.6 concludes the chapter.

4.2 Adaptive kernel density estimator

In order to identify the clusters in the data, we employ the CRA cluster algorithm proposed by Cuevas et al. (2000, 2001). The CRA clustering algorithm is briefly described as follows:

1. Initialize the cluster membership vector $\boldsymbol{m}$ of size n to be a zero vector. Let $m_1 = 1$.

2. Obtain the estimated level set of high density region $\{\hat{f} > c\}$. Define data point in the high density region to be $x_j, j = 1, 2, \cdots, k-1$, and data points in the low density region to be $x_j, j = k, k+1, \cdots, n$, where $k-1$ is the sample size of the level set of high density region.

3. Loop on $i = 1, \cdots, k-1$,

(a) Start with ball center in (2.4.1), $\boldsymbol{z}_i$. Find the minimum distance,

$$d_i = \min_i \min_j \|\boldsymbol{z}_i - \boldsymbol{z}_j\|,$$

for j from $i+1$ to k and remember the location of j where the distance is minimum and exchange $x_i + 1$ with x_j.

(b) If $d_i \leq 2\epsilon$, then $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+1}$ are in the same connected component and let $m_{i+1} = m_i$.

(c) If $d_i > 2\epsilon$, then $\boldsymbol{x}_{i+1}$ starts a new connected component and let $m_{i+1} = m_i + 1$.

(d) End of loop.

4. The estimated number of cluster $t = m_k$.

5. The membership of the data in the low density region, $m_i, i = k+1, \cdots, n$, can be assigned equal to the its nearest neighbor in the existing connected components. Alternatively, we can let those data points to be in an independent set of low density region.

6. If there exist any connect component with size less than 5% of the sample size, it is too small to be a meaningful cluster. Therefore, it can be merged to its nearest neighbour component, and let $t = t - 1$.

7. Lastly, construct data clusters based on $m_i, i = 1, \cdots, n$, and let $\widehat{T} = t$.

The CRA algorithm described by Cuevas et al. (2001) also involves bootstrapping to create artificial clusters mainly for small sample problems.

4.2.1 Posterior of bandwidth parameters

The Bayesian approach treats all parameters and data of a statistical model as random quantities. Let $\boldsymbol{x}$ denote the vector of observed data, let θ denote the vector of parameters. Denote the likelihood as $p(\boldsymbol{y}|\theta)$ and a prior density as $p(\theta)$, and the Bayes theorem gives the posterior density of θ as

$$\pi(\theta|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\theta)p(\theta)}{\int p(\boldsymbol{y}|\theta)p(\theta)}. \tag{4.2.1}$$

The denominator of (4.2.1) is the normalizing constant, which is most of the time unknown. Therefore, the Bayesian inference is focused on

$$\pi(\theta|\boldsymbol{y}) \propto p(\boldsymbol{y}|\theta)p(\theta). \tag{4.2.2}$$

Markov Chain Monte Carlo (MCMC) is described by Gelman et al. (2004) as "a general method based on drawing values of θ from approximate distributions and then correcting those draws to better approximate the target posterior distribution". Here, we denote the vector of non-zero elements in the bandwidth matrix as $\boldsymbol{h}$, and treat it as the unknown parameters where we can estimate them through MCMC simulation.

In order to make the bandwidth selector to be adaptive with respect to different clusters, let ω_j represent the cluster membership for $\boldsymbol{x}_j$, and define $H(\omega_j)$ as the adaptive bandwidth matrix depending on cluster membership of $\boldsymbol{x}_j$. The possible values of ω_j for $j = 1, \cdots, n$, are integers $1, \cdots, T$, where T is estimated by the CRA clustering. Hence, there are a total of T bandwidth matrices to estimate.

Moreover, it is also possible to combine the cluster-adaptive and tail-adaptive shown in Chapter 3. Simply by allowing $\boldsymbol{x}_j$ in LDR to have their own membership, e.g. $T+1$. In such a case, there is one additional bandwidth matrix to be estimated. We call the resulting estimator cluster+tail adaptive bandwidth kernel density estimator.

Therefore, the general form of kernel estimator based on cluster can be written as

$$\hat{f}_{H(\omega_j)}(\boldsymbol{x}) = \frac{1}{n}\sum_{j=1}^{n} |H(\omega_j)|^{-1/2} K\Big(H(\omega_j)^{-1/2}(\boldsymbol{x}-\boldsymbol{x}_j)\Big), \tag{4.2.3}$$

and the leave-one-out estimator is

$$\hat{f}_{H(\omega_j),i}(\boldsymbol{x}_i) = \frac{1}{n-1}\sum_{\substack{j=1 \\ j\neq i}}^{n} |H(\omega_j)|^{-1/2} K\Big(H(\omega_j)^{-1/2}(\boldsymbol{x}_i-\boldsymbol{x}_j)\Big), \tag{4.2.4}$$

If the bandwidth matrix is diagonal, define $\boldsymbol{h}(\omega_j) = (h_1(\omega_j), h_2(\omega_j), \cdots, h_d(\omega_j))'$ as the vector of adaptive bandwidths for the sample point j. The prior of each component of $\boldsymbol{h}(\omega_j)$ is chosen to be

$$p(h_k(\omega_j)|\lambda) \propto \frac{1}{1+\lambda h_k^2(\omega_j)}, \tag{4.2.5}$$

for $k = 1, \cdots, d$, where λ is a hyperparameter controlling the shape of the prior density.

By Bayes theorem, the posterior of $\boldsymbol{h}(\omega_j)$ is

$$\pi(\boldsymbol{h}(\omega_j)|\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n) \propto \prod_{i=1}^{n} \hat{f}_{\boldsymbol{h}(\omega_j),i}(\boldsymbol{x}_i) \times \left[\prod_{k=1}^{d} \frac{1}{1+\lambda h_k^2(\omega_j)}\right]. \tag{4.2.6}$$

The likelihood in the posterior is flat when the elements of $\boldsymbol{h}$ are large. Nonetheless, the prior penalises the large updates of $\boldsymbol{h}$ by giving low prior probability. We found

that the MCMC results are insensitive to the values of the hyperparameter λ. Hence, we assume $\lambda = 1$.

4.2.2 Metropolis-Hastings Algorithm

The posterior given by (4.2.6) has a non-standard form. The important issue here is how to draw random numbers from this posterior. One way of this is to use the random-walk Metropolis-Hastings (MH) algorithm (Metropolis et al. 1953, Hastings 1970).

Denote $H(q), q = 1, \cdots, T$, as the bandwidth matrices. The sampling procedure is as follows

1. At iteration i, given current state $H^{(i)}(q)$.

2. For $q = 1$ to T,

   (a) Sample a candidate $\widetilde{H}(q)$ from a proposal candidate density $\alpha(\widetilde{H}(q)|H^{(i)}(q))$ which is normal. Therefore, each element of $H^{(i)}(q)$ is updated through

   $$\tilde{h} = h^{(i)} + \tau\epsilon,$$

   where $\epsilon \sim N(0,1)$, and τ is a tuning parameter.

   (b) Accept $\widetilde{H}(q)$, with probability $\delta(\widetilde{H}(q), H^{(i)}(q))$ where

   $$\delta(\widetilde{H}(q), H^{(i)}(q)) = \min\left\{1, \frac{\pi(\widetilde{H}(q)|\boldsymbol{x})}{\pi(H^{(i)}(q)|\boldsymbol{x})}\right\}.$$

   (c) If $\widetilde{H}(q)$ is accepted, the next state is $H^{(i+1)}(q) = \widetilde{H}(q)$.

(d) If $\widetilde{H}(q)$ is rejected, the next state is $H^{(i+1)}(q) = H^{(i)}(q)$.

3. End of iteration.

Therefore, the whole MCMC algorithm for estimating cluster based adaptive bandwidth can be described as follows

1) Derive an initial kernel estimator of f with a global bandwidth chosen through NRR or estimated through the sampling algorithm of Zhang et al. (2006).

2) Based on the initial density estimator, apply the CRA clustering algorithm and assign cluster membership to sample data.

3) Based on the cluster membership, estimate cluster-adaptive bandwidths via Random-walk MH algorithm.

4) Run MH algorithm for 3,000 iterations as a burn-in period, and the following 10,000 iterations be recorded.

5) After each iteration, if there are any successful updates to the values of the bandwidth matrix parameters, obtain new density for $\boldsymbol{x}$, and then re-run CRA clustering algorithm.

6) Obtain the mean and other statistics based on the simulated posterior sample.

Random-walk MH algorithm uses a symmetric density, e.g. normal density, to generate candidates. As long as the Random-walk MH algorithm works, it is generally not necessary to explore other proposal densities.

4.2.3 Convergence Diagnostics

The samples drawn from the posterior $\pi(\theta|\boldsymbol{y})$ form a sequence denoted by $\{\theta^i : i = 1, 2, \cdots, M\}$, where M is the number of recorded iterations. We are interested in parameters which are estimated through the ergodic averages in the form of $\bar{\theta} = \frac{1}{M}\sum_{i=1}^{M}\theta^i$. The central limit theorem of ergodic averages shows

$$\sqrt{M}\left(\bar{\theta} - \mathrm{E}_\pi(\theta)\right) \to N(0, \sigma_f^2), \tag{4.2.7}$$

where σ_f is positive and constant. However, the algorithms is inefficient if σ_f is too large comparing to the variance of θ draw from $\pi(\cdot)$. Kim et al. (1998) shows the simulation inefficient factor

$$\mathrm{SIF} = \frac{\sigma_f^2}{var_\pi(\theta)}, \tag{4.2.8}$$

in order to measure the efficiency of the MCMC estimating of $E_\pi(\theta)$. The $var_\pi(\theta)$ is the variance of the posterior sample mean,

$$var_\pi(\theta) = \frac{1}{M-1}\sum_{i=1}^{M}\left(\theta^i - \bar{\theta}\right). \tag{4.2.9}$$

Roberts (1996) showed the batch mean for estimating σ_f^2 as follows. Given sufficient large n that makes $M = m * n$, let

$$\theta_k = \frac{1}{n}\sum_{i=(k-1)n+1}^{kn}\theta^i, \tag{4.2.10}$$

for $k = 1, 2, \cdots, m$. The σ_f^2 can be estimated by

$$\hat{\sigma}_f^2 = \frac{n}{m-1} \sum_{i=1}^{m} \left(\theta_k - \bar{\theta}\right)^2. \tag{4.2.11}$$

4.2.4 Restricted bin-adaptive estimator

In addition to cluster-adaptive estimator, we propose an alternative bin-adaptive estimator for multimodal distributions. In the study of Sain (2002), the density of $\boldsymbol{x}$ was calculated based on bin center $\boldsymbol{t_j}$, $j = 1, ..., m$ as shown in (2.3.25). Bins were defined as equal spaced mesh points over the support of the density. If there are m bins on each dimension, m^d number of bins are created in total. Even if diagonal bandwidth matrix is used, the number of parameters to estimate in total is m^{d+1}. This approach therefore requires d to be small due to the curse of dimensionality.

The advantage of using bin center $\boldsymbol{t_j}$ instead of $\boldsymbol{x_j}$ is that the former is computationally less intensive. In this study, we are more concerned about the accuracy of the estimation rather than the computation time, hence we still propose to use $\boldsymbol{x_j}$ instead of $\boldsymbol{t_j}$ for bandwidth selection. To be able to apply the binning algorithm to high-dimensional data, we have to make compromise on the number of bandwidths to estimate. We propose the restricted bin-adaptive estimator. Let m be the number of bins on each dimension and $\tau_{j,k}$ be the bin membership for the jth observation on dimension k. Hence for any k, the possible values of $\tau_{j,k}$ are $1, 2, \ldots, m$. The restricted bin-adaptive kernel density estimator can be shown as

$$\hat{f}_{RB}(\boldsymbol{x}) = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{h(\tau_{j,1}) h(\tau_{j,2}) \cdots h(\tau_{j,d})} K\left(\frac{x_1 - x_{j,1}}{h(\tau_{j,1})}, \frac{x_2 - x_{j,2}}{h(\tau_{j,2})}, \cdots, \frac{x_d - x_{j,d}}{h(\tau_{j,d})}\right), \tag{4.2.12}$$

and the leave-one-out estimator is

$$\hat{f}_{RB,i}(\boldsymbol{x}_i) = \frac{1}{n-1}\sum_{\substack{j=1\\ j\neq i}}^{n} \frac{1}{h(\tau_{j,1})h(\tau_{j,2})\cdots h(\tau_{j,d})} K\left(\frac{x_{i,1}-x_{j,1}}{h(\tau_{j,1})}, \frac{x_{i,2}-x_{j,2}}{h(\tau_{j,2})}, \cdots, \frac{x_{i,d}-x_{j,d}}{h(\tau_{j,d})}\right), \tag{4.2.13}$$

where $\boldsymbol{h}$ is diagonal bandwidth matrix, defined $\boldsymbol{h}(\tau_j) = (h(\tau_{j,1}), h(\tau_{j,2}), \cdots, h_d(\tau_{j,d}))'$ as the vector of adaptive bandwidths for the sample point j. Under this restriction, a separate bandwidth parameter is used in each bin on each dimension. This means the total number of bandwidth parameters to estimate is $m \times d$. Hence in high dimensional data, the number of bandwidths to estimate is much less than m^{d+1}.

The MCMC algorithm for bin-adaptive bandwidth estimation is relatively straitforward, because the bin membership can be assigned at beginning of the algorithm and fixed throughout the estimation process. There are several ways to assign bin memberships. The equal-distance-binning method given by Sain (2002) is very easy to apply. It works very much like a histogram, where the bin-width are the same for all bins. However, the number of data points in each bin can be very different. For example, a bin near the mode is going to have a lot more data points than a bin in the tail. If the underlying density is multimodal or fat-tailed, there maybe bins with zero data points. From a computation point of view, bins with very few data points can results in highly varying estimates. To avoid this problem, we suggest using an equal-proportion-binning algorithm, where the number of data points in each bin are equal. As a result, a bin near the mode will have equal data points to a bin in the tail.

4.3 A Monte Carlo simulation study

To investigate the performance of the proposed tail-adaptive kernel density estimator, we approximate Kullback-Leibler information between the density estimator and its corresponding true density via Monte Carlo simulation. Kullback-Leibler information defined in (2.3.18) is a measure of discrepancy between the true density and its estimator. To approximate Kullback-Leibler information, we draw a large number of random vectors $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ from true density $f(\boldsymbol{x})$ and compute

$$\hat{d}_{KL}\left(f(x), \hat{f}(x)\right) = \frac{1}{N}\sum_{i=1}^{N} \log\left(f(\boldsymbol{x}_i)/\hat{f}(\boldsymbol{x}_i)\right), \tag{4.3.1}$$

where $\hat{f}(\cdot)$ denote a density estimator of $f(\cdot)$. The performance of a bandwidth estimate is examined through the performance of the resulting kernel density estimator. A bandwidth estimation method is better than its competitor if Kullback-Leibler information resulted from the former is less than that resulted from the latter.

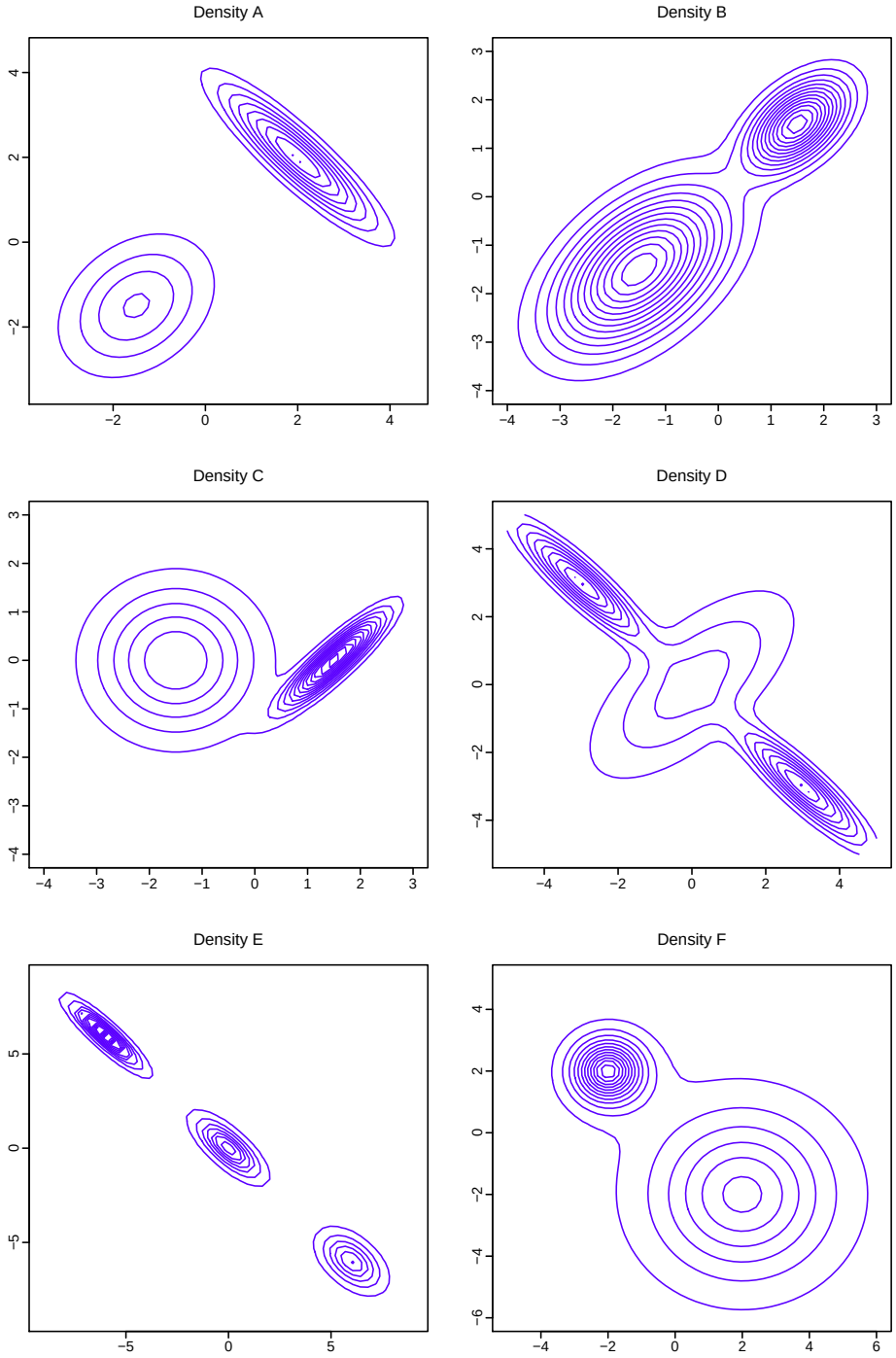
4.3.1 True densities

We conduct Monte Carlo simulation by simulating samples from six target densities labeled A, B, C, D, E and F. Figure 4.1 shows the contour plot for bivariate densities. These densities are multimodal. Density A to D are normal densities. Density E and F are Student t densities. Their specifications are explained as follows.

Density A is a mixture of two equally weighted normal densities with bimodality:

$$f_A(\boldsymbol{x}|\mu, \Sigma) = \frac{1}{2}\phi(\boldsymbol{x}|\mu_1, \Sigma_1) + \frac{1}{2}\phi(\boldsymbol{x}|\mu_2, \Sigma_2),$$

Figure 4.1: *Contour graphs of target bivariate densities.*



where $\phi(\boldsymbol{x}|\mu,\Sigma)$ is a multivariate normal density with mean μ and variance-covariance matrix Σ are given as

$$\mu_1 = \begin{pmatrix} -1.5 \\ -1.5 \end{pmatrix},\ \Sigma_1 = \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix},\ \mu_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix},\ \Sigma_2 = \begin{pmatrix} 1 & -0.9 \\ -0.9 & 1 \end{pmatrix}.$$

Density B is a mixture of two normal densities with different weights but an equal height at the modes:

$$f_B(\boldsymbol{x}|\mu,\Sigma) = \frac{3}{4}\phi(\boldsymbol{x}|\mu_1,\Sigma_1) + \frac{1}{4}\phi(\boldsymbol{x}|\mu_2,\Sigma_2).$$

where

$$\mu_1 = \begin{pmatrix} -1.5 \\ -1.5 \end{pmatrix},\ \Sigma_1 = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix},\ \mu_2 = \begin{pmatrix} 1.5 \\ 1.5 \end{pmatrix},\ \Sigma_2 = \begin{pmatrix} 1/3 & 1/6 \\ 1/6 & 1/3 \end{pmatrix}.$$

Note that Density A and B are also used in Chapter 3.

Density C is a bi-modal normal density with different weights and spread on each mode:

$$f_C(\boldsymbol{x}|\mu,\Sigma) = \frac{3}{4}\phi(\boldsymbol{x}|\mu_1,\Sigma_1) + \frac{1}{4}\phi(\boldsymbol{x}|\mu_2,\Sigma_2).$$

where

$$\mu_1 = \begin{pmatrix} -1.5 \\ 0 \end{pmatrix},\ \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ \mu_2 = \begin{pmatrix} 1.5 \\ 0 \end{pmatrix},\ \Sigma_2 = \begin{pmatrix} 1/3 & 3/10 \\ 3/10 & 1/3 \end{pmatrix}.$$

Density D is a mixture of three normal densities with trimodal feature and different orientations:

$$f_D(\boldsymbol{x}|\mu,\Sigma) = \frac{1}{2}\phi(\boldsymbol{x}|\mu_1,\Sigma_1) + \frac{1}{4}\phi(\boldsymbol{x}|\mu_2,\Sigma_2) + \frac{1}{4}\phi(\boldsymbol{x}|\mu_3,\Sigma_2).$$

where

$$\mu_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \ \Sigma_1 = \begin{pmatrix} 3 & 2 \\ 2 & 3 \end{pmatrix}, \ \mu_2 = \begin{pmatrix} 3 \\ -3 \end{pmatrix}, \ \Sigma_2 = \begin{pmatrix} 1 & -0.9 \\ -0.9 & 1 \end{pmatrix} \mu_3 = \begin{pmatrix} -3 \\ 3 \end{pmatrix}.$$

Density D is designed to allow two of the three clusters to have the same covariance matrix to reduce the differences among the clusters, which would allow the competing estimators, e.g. the global bandwidth estimators, to enjoy more advantage.

Density E is another tri-modal mix but with different covariance matrices, which is contrast to Density D. Density E is a mixture of three Student t densities the :

$$f_E(\boldsymbol{x}|\mu, \Sigma, \nu) = \frac{1}{3} t_d(\boldsymbol{x}|\mu_1, \Sigma_1, \nu) + \frac{1}{3} t_d(\boldsymbol{x}|\mu_2, \Sigma_2, \nu) + \frac{1}{3} t_d(\boldsymbol{x}|\mu_3, \Sigma_3, \nu).$$

where $t_d(\boldsymbol{x}|\mu, \Sigma, \nu)$ denotes Student t distribution. The degrees of freedom ν is 5. The location parameter μ and dispersion matrix Σ are specified as follows:

$$\mu_1 = \begin{pmatrix} -6 \\ 6 \end{pmatrix}, \ \mu_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \ \mu_3 = \begin{pmatrix} 6 \\ -6 \end{pmatrix}, \ \Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix},$$

where $\rho = -0.9, -0.75$ and -0.5 for Σ_1, Σ_2 and Σ_3, respectively.

Density F is a mixture of two Student t densities with degrees of freedom $\nu = 5$:

$$f_F(\boldsymbol{x}|\mu, \Sigma, \nu) = \frac{4}{5} t_d(\boldsymbol{x}|\mu_1, \Sigma_1, \nu) + \frac{1}{5} t_d(\boldsymbol{x}|\mu_2, \Sigma_2, \nu).$$

where

$$\mu_1 = \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \ \Sigma_1 = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix} \mu_2 = \begin{pmatrix} -2 \\ 2 \end{pmatrix}, \ , \Sigma_2 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}.$$

The contour plot of each of the six bivariate densities is given in Figure 4.1. We can find that these densities exhibit different distributional and clustering properties. Density A to D are normal densities, while density E and F are Student t densities. In addition, the clusters in each density are designed to have different concentrations.

4.3.2 Accuracy of our Bayesian bandwidth estimation

We generate samples of sizes $n = 500, 1000, 2000$ from each of the six bivariate densities. The kernel function for estimating multivariate densities was chosen to be the product of univariate Gaussian kernels was used as the kernel function for estimating multivariate densities. The bandwidth matrix in estimating multivariate densities is chosen to be a diagonal matrix. We wish to examine the performance of the proposed cluster and cluster+tail adaptive bandwidth kernel density estimator with five other kernel density estimators: two global bandwidth matrix estimators, namely the NRR method by Scott (1992) and the Bayesian method presented by Zhang et al. (2006); the tail-adaptive kernel density estimator discussed in Chapter 3 with $\alpha = 0.05$ and the proposed restricted bin-adaptive bandwidth kernel density estimator with $m = 6$.

For each generated sample data set, we apply the random-walk Metropolis-Hastings algorithm to the update of all bandwidths in the univariate situation (or all components of the bandwidth matrices in the bivariate situation) with the acceptance probability calculated through (4.2.6). There are 3,000 iterations during the burn-in period, and the recorded period contains 10,000 iterations. We then compute the batch-mean standard deviation and the simulation inefficient factor (SIF) to monitor the mixing performance (or loosely speaking, the convergence performance). As the simulated

Table 4.1: *MCMC results based on sample data of size 1000 generated from* f_C

	Bandwidths	Mean	Standard deviation	Batch-mean standard deviation	SIF	Acceptance rate
Cluster adaptive	$1/h_1(1)$	2.40	0.29	0.0100	12.10	0.23
	$1/h_2(1)$	2.33	0.26	0.0090	12.64	
	$1/h_1(2)$	8.20	1.68	0.0930	30.61	
	$1/h_2(2)$	7.48	1.20	0.0606	25.24	
Bin $m = 6$	$h_{1,1}$	0.57	0.07	0.0040	28.55	0.21
	$h_{2,1}$	0.42	0.07	0.0052	50.22	
	$h_{3,1}$	0.35	0.07	0.0054	62.65	
	$h_{4,1}$	0.39	0.09	0.0077	73.09	
	$h_{5,1}$	0.14	0.02	0.0014	31.42	
	$h_{6,1}$	0.12	0.02	0.0016	45.38	
	$h_{1,2}$	0.40	0.05	0.0036	42.03	
	$h_{2,2}$	0.21	0.04	0.0036	67.57	
	$h_{3,2}$	0.14	0.03	0.0032	42.07	
	$h_{4,2}$	0.14	0.03	0.0029	78.81	
	$h_{5,2}$	0.17	0.03	0.0029	75.25	
	$h_{6,2}$	0.28	0.04	0.0035	66.79	

chain is a Markov chain, the SIF value can be roughly interpreted as the number of draws needed so as to produce independent draws. A small SIF value usually indicate good mixing performance. In addition, a plot of the sample path of each parameter, together with its autocorrelation function (ACF) and histogram graphs is also presented for visual inspection of the mixing performance.

Consider a sample generated from $f_C(\boldsymbol{x})$ with sample size $n = 1000$. Figure 4.2 presents graphs of the sample path, its ACF and histogram of each bandwidth. Table 4.1 presents a summary of the MCMC results, in which we found that the SIF values are small, and the batch-mean standard deviations are respectively, much smaller than their counterparts of overall standard deviations. These indicators suggest good mixing performance of the proposed sampling algorithm applied to the cluster-adaptive and restricted bin-adaptive bandwidth estimator.

Figure 4.2: *Plots of posterior draws obtained through our proposed sampling algorithm for cluster-adaptive bandwidths in kernel density estimation based on sample data of size 1000 draw from f_C: (a) $h_1(1)$; (b) $h_2(1)$; (c) $h_1(2)$; and (d) $h_2(2)$.*

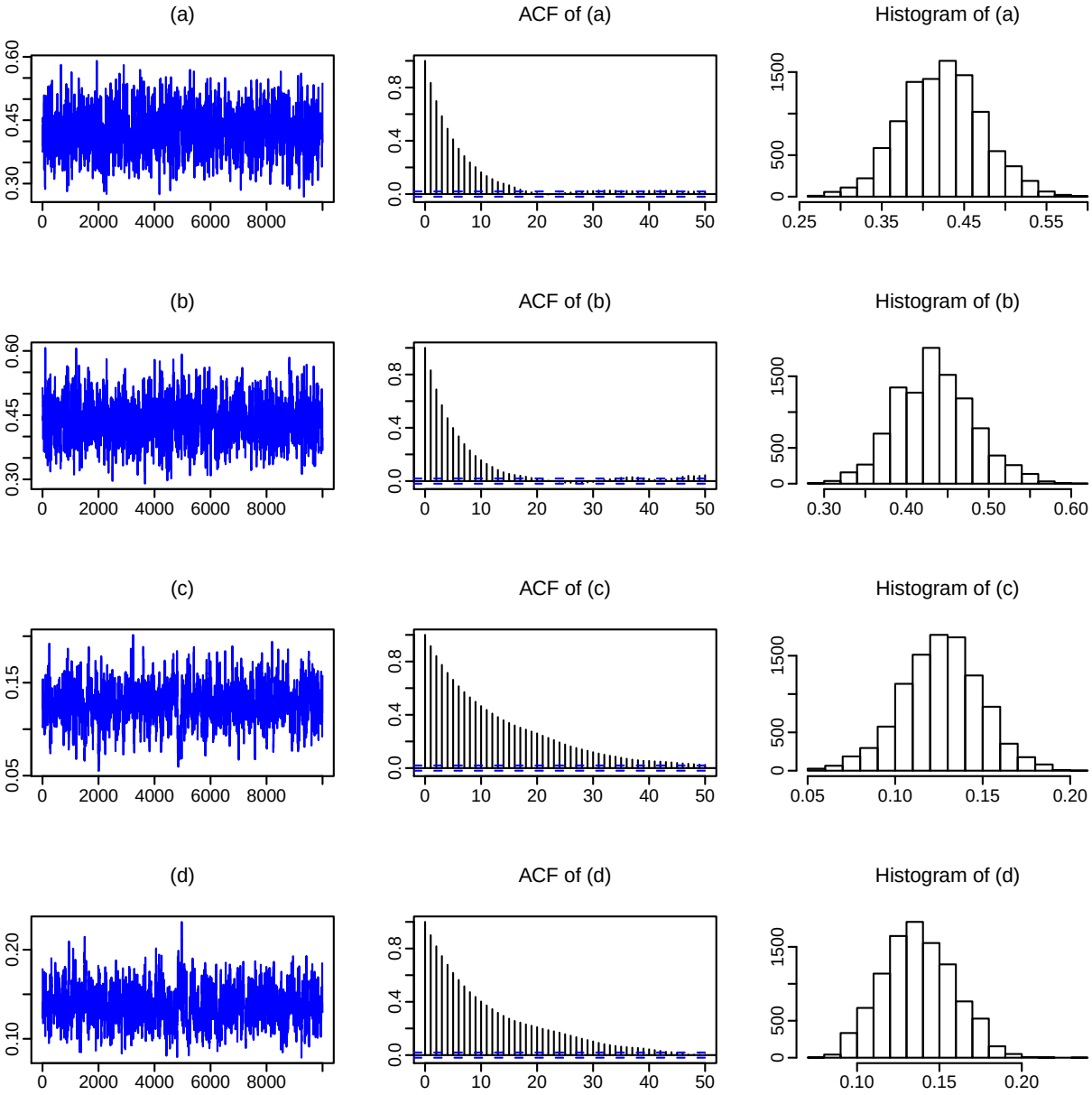


Table 4.2: *Estimated Kullback-Leibler information for bivariate densities*

		Global		Cluster-adaptive			
Density	n	NRR	Bayesian	Tail	Bin	Cluster	Cluster+tail
f_A	500	0.2878	0.0858	0.0772	0.0769	0.0691	0.0578
	1000	0.2382	0.0617	0.0498	0.0488	0.0465	0.0373
	2000	0.1981	0.0402	0.0339	0.0273	0.0273	0.0245
f_B	500	0.1201	0.0499	0.0444	0.0586	0.0523	0.0441
	1000	0.0826	0.0349	0.0332	0.0413	0.0341	0.0324
	2000	0.0653	0.0256	0.0219	0.0239	0.0225	0.0193
f_C	500	0.1539	0.1066	0.0918	0.0913	0.0571	0.0680
	1000	0.1228	0.0779	0.0663	0.0605	0.0416	0.0380
	2000	0.1047	0.0661	0.0512	0.0435	0.0344	0.0315
f_D	500	0.3830	0.1210	0.1002	0.1369	0.0837	0.0755
	1000	0.2993	0.1204	0.0831	0.1186	0.0650	0.0653
	2000	0.2625	0.0776	0.0488	0.0807	0.0443	0.0386
f_E	500	0.8582	0.1802	0.0991	0.1386	0.1461	0.0864
	1000	0.7429	0.1252	0.0647	0.1075	0.1137	0.0600
	2000	0.6510	0.1099	0.0550	0.0903	0.0965	0.0486
f_F	500	0.1599	0.1544	0.0882	0.1776	0.1367	0.0708
	1000	0.1101	0.1181	0.0522	0.0952	0.0860	0.0355
	2000	0.0807	0.1195	0.0411	0.0864	0.0809	0.0334

f_C is a bimodal normal density. The left mode (cluster 1) has more weight but more sparsely distributed, and the right mode (cluster 2) is more densely distributed with less weight (see Figure 4.1). The cluster-adaptive density estimator clearly captures the bimodal feature of the true underlying density. The estimated bandwidth of cluster 1 is much larger than those of cluster 2. This is expected as more smoothing is required for cluster 1 than cluster 2.

One the other hand, the restricted bin-adaptive estimator aims to capture the features of marginal distributions. Focusing on the horizontal axis (dimension 1), the bandwidth of bin 5 and 6 are clearly smaller than bin 1 to 4. This suggests less smoothing is applied to the right mode on dimension 1. Moreover, bin 1 contains the left tail of the distribution and its bandwidth is significantly larger. This feature of bin-adaptive

estimator is similar to the tail-adaptive estimator proposed in Chapter 3. On the vertical axis (dimension 2), the two modes are overlapped. As a result, the estimated bandwidth on dimension 2 are generally smaller than those of dimension 1.

In order to examine the performance of the proposed cluster-adaptive density estimator, we derived the global bandwidths (or bandwidth matrices for the bivariate situation) through the NRR and the Bayesian sampling method and report the resulting Kullback-Leibler information. We note that MISE was not presented in this case. MISE is extremely time consuming to compute, and in the bivariate case shown in Chapter 3, the MISE results are very consistent with the Kullback-Leibler information. Hence we believe calculating MISE again in this case is not necessary.

We generated N=100,000 random numbers (or vectors for the bivariate situation) from the true density and calculated the estimated Kullback-Leibler information defined by (3.3.1). The estimated Kullback-Leibler information for bivariate densities is given in Table 4.2.

Among all six densities considered, the tail-adaptive density estimator consistently outperforms the global-bandwidth density estimator with bandwidth matrix estimated through either the NRR or Bayesian sampling. Restricted bin-adaptive density estimator shows mixed performance comparing to tail-adaptive density estimator. For example, it performs better in f_A and f_C but performs worse in the other densities.

Cluster-adaptive density estimator shows good performance when the underlying density is normal. For example, in f_A to f_D, the cluster-adaptive density estimator out performs both tail- and bin-adaptive density estimators. However, it performers

worse than tail-adaptive density estimator in fat-tailed densities such as f_E and f_F. When combining cluster and tail adaptive bandwidths, the resulting density estimator shows the best performance. The cluster+tail adaptive density estimator delivers the lowest Kullback-Leibler information in all data sets considered.

The results suggest that when the underlying density is multimodal, bandwidth selection based on clustering algorithm can produce more satisfactory results compare to global and bin-adaptive bandwidth selection methods. When the density function is fat-tailed, tail- and cluster-adaptive bandwidth selection can be combined to make adaptive bandwidth selections.

4.4 An application to the Old Faithful geyser data

Old Faithful geyser is located in Yellowstone National Park in Wyoming, USA. Azzalini & Bowman (1990) studied the data of eruption duration and waiting time to the next eruption collected from August 1st to August 15th in 1985. The data set consists of 272 pairs of observations measured in minutes, the scatter plot is shown in Figure 4.3. This data set is famous for its feature of multimodality and has been discussed by many studies (see, for example Scott 1992, Hyndman 1996).

We applied our Bayesian sampling algorithm to estimate bandwidth matrices for the cluster-adaptive kernel density estimator based on this data set. We also applied the Bayesian sampling algorithm proposed by Zhang et al. (2006) and NRR to the estimation of global bandwidth matrix for the kernel estimation of the Old Faithful geyser data.

Figure 4.3: *Scatter plot of eruption time and waiting time to the next eruption (in minutes) of Old Faithful geyser in Yellowstone National Park, USA.*

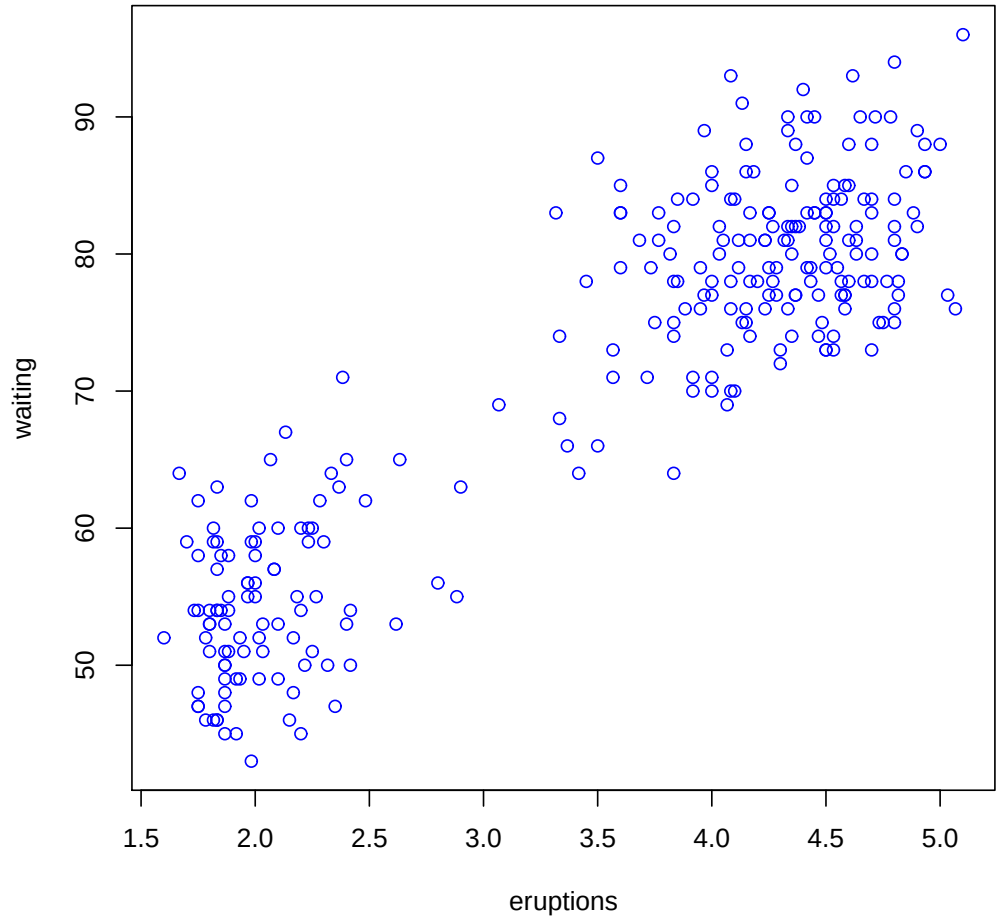


Figure 4.4: *Surface graphs and contour plots of the three density estimators produced by (a) Cluster-adaptive bandwidth; (b) Bayesian global bandwidth; and (c) NRR bandwidth.*

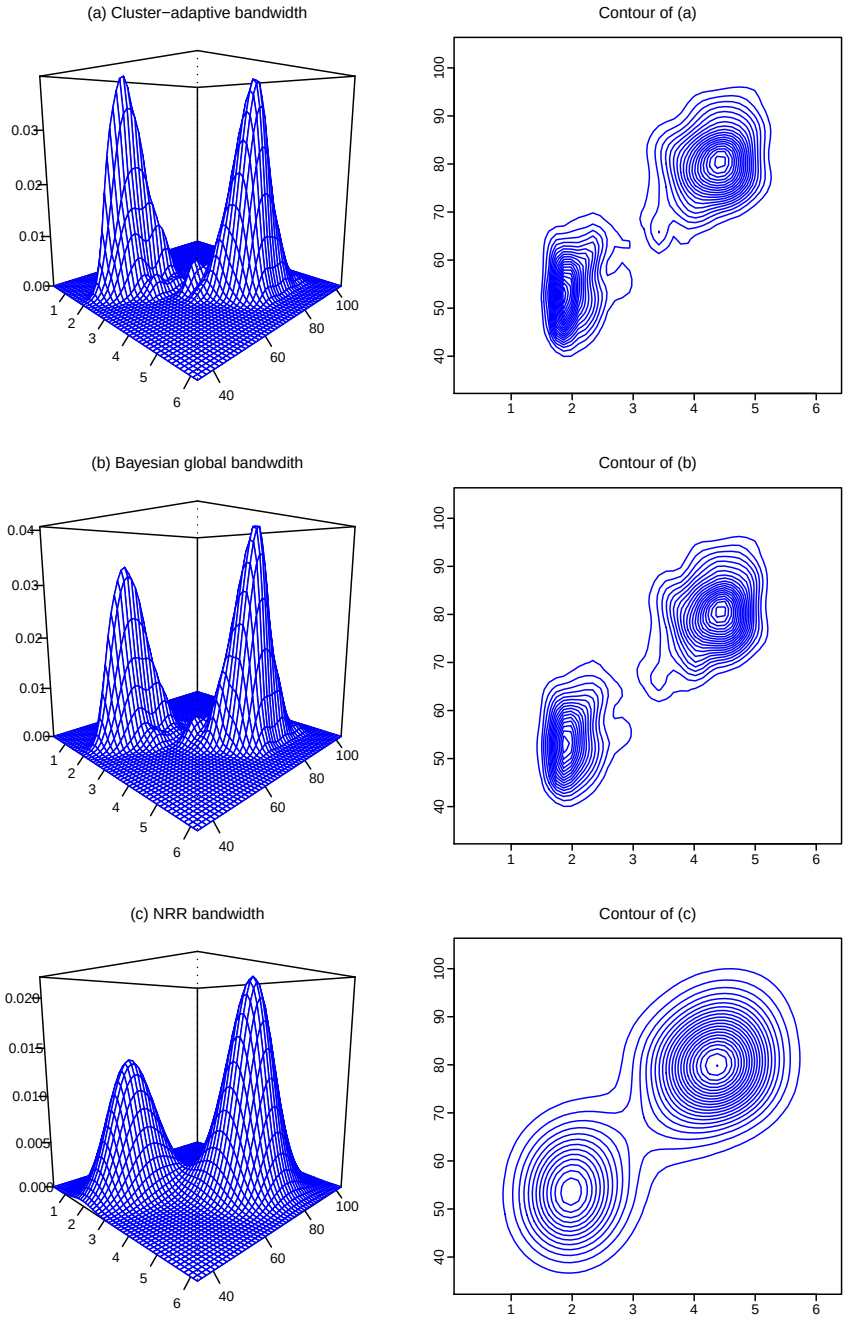


Table 4.3: *A summary of MCMC results obtained through the proposed Bayesian sampling algorithm to the cluster-adaptive kernel density estimator of the Old Faithful geyser data*

	Bandwidths	Mean	Standard deviation	SIF	Acceptance rate	log marginal likelihood
NRR	h_1	0.45				
	h_2	5.33				
Bayesian global bandwidth	h_1	0.15	0.0181	34.07	0.22	-1151.01
	h_2	2.80	0.4381	13.91		
Cluster-adaptive bandwidth	$1/h_1(1)$	0.38	0.0746	14.09	0.28	-1138.26
	$1/h_2(1)$	5.13	0.8169	6.94		
	$1/h_1(2)$	0.40	0.1224	29.04		
	$1/h_2(2)$	8.90	1.6955	12.58		

There are 3,000 iterations in burn-in period and 10,000 iterations in the recorded period for both sampling algorithms. Table 4.3 presents a summary of the results, where the batch-mean standard deviation and SIF measures indicate very good mixing performance of both samplers. Moreover, we calculated the log marginal likelihood of Newton & Raftery (1994) for each of the two density estimators so as to decide which is favoured against the other. The log marginal likelihood for cluster-adaptive kernel density estimator and global kernel density estimator are -1138.26 and -1151.01, respectively. Thus, our cluster-adaptive density estimator is favoured against the global density estimator.

The estimated cluster-adaptive bandwidths are given in the 3rd column of Table 4.3. Based on this result, we calculated the cluster-adaptive density estimator of the Old Faithful geyser data. Figure 4.4 presents the surface and contour plot obtained via cluster-adaptive, Bayesian global and NRR density estimators. The density estimator produced by NRR has obviously been over-smoothed. As shown in the plot, the

Bayesian global density estimator has captures more dynamics than the NRR. However, the main difference of the estimated density based on cluster-adaptive bandwidths is that the height of the density function of the two clusters are approximately equal. This is because the cluster-adaptive density estimator is able to assign a smaller bandwidth to the left cluster (cluster 2). However, the global bandwidth density estimator employs only one bandwidth and over-smoothed the left cluster.

4.5 Conclusion

This chapter proposes cluster-adaptive bandwidth kernel density estimator for data with multimodality. This method employs a clustering algorithm to assign different bandwidths to different clusters identified in the data set. We derived the posterior of bandwidth parameters based on Kullback-Leibler information and presented an MCMC sampling algorithm to estimate bandwidths.

The Monte Carlo simulation study is designed to examine the performance of the proposed methods when the data exhibits multimodality. The results suggest that when the underlying density is a mixture of normal, the kernel density estimator with cluster-adaptive bandwidths estimated through our proposed Bayesian sampling algorithm outperforms its competitor. When the underlying densities are fat-tailed, the combined approach of tail- and cluster-adaptive density estimator performs the best. The restricted bin-adaptive density estimator shows mixed performance comparing to tail-adaptive approach. In an empirical application, we used our Bayesian sampling algorithm to estimate bandwidths matrices for the cluster-adaptive kernel density estimator on the well known data set of eruption duration and waiting time to the next

eruption collected from Old Faithful greyer which is famous of its clustering nature. The results show clear advantage of the proposed cluster-adaptive kernel density estimator over traditional approaches.

In summary, by incorporating clustering information and assigning a different bandwidth matrix to each cluster, the performance of the kernel density estimator can be improved. In light of this result, it is intuitive to believe that improved density estimation performance would further facilitate on improving the clustering accuracy. It should be noted that future research could be undertaken to investigate and provide more insight on such issue.

Chapter 5

Bayesian estimation for a semi-parametric nonlinear volatility model

5.1 Introduction

Empirical evidence has shown that the volatility of financial asset returns is often highly persistent and asymmetrically distributed. The early studies has focused on the parametric approach of ARCH (Engle 1982) and GARCH (Bollerslev 1986) type models to capture the stylised facts. The nonparametric branch of research on volatility models has aimed at addressing the strong parametric assumption of ARCH and GARCH type models, such as the linearity assumption in the volatility equation (e.g. Pagan & Schwert 1990) and the error distribution (e.g. Engle & Gonzalez-Rivera 1991).

Recently, the nonlinear nonstationary heteroscedastic (NNH) model was proposed (Park 2002) as an alternative volatility model. The NNH model assumes the conditional variance as a known parametric nonlinear function of a persistent explanatory variable x_t. Han & Park (2008) extended the NNH model by allowing the ARCH(1) component in the model, while Han & Zhang (2012) proposed a nonparametric version of the NNH model called the nonstationary nonparametric volatility (NNV) model.

The NNV model assumes the nonlinear function of x_t in the model to be unknown and employs the Nadaraya-Watson as its estimator. The Nadaraya-Watson estimator is a well-known nonparametric estimator and its performance is highly dependant on its bandwidth. However, the bandwidth selection issue was not discussed in details by Han & Zhang (2012). In this chapter, we propose a new volatility model which combines the ARCH(1) model and the NNV model. This study aims to develop a Bayesian sampling algorithm to select the optimal bandwidth for the Nadaraya-Watson estimator.

This chapter is organised as follows. In Section 5.2, we present the proposed SNV model and show the posterior distribution of the parameters. Section 5.3 discusses and outlines the evaluation criterion of the proposed models in relation with alternative NNV and NNH models based on daily return data from major global financial markets. Section 5.4 provides the evaluation results in terms of in-sample and out-of-sample performances, including the value-at-risk forecasting performance. Section 5.5 concludes the chapter.

5.2 Semi-Parametric Volatility Models

Let y_t denote an asset's return that is modeled as

$$y_t = \mu + u_t,$$

where u_t is a conditional heteroscedastic error process defined as

$$u_t = \sigma_t \varepsilon_t \,, \tag{5.2.1}$$

with ε_t, for $t = 1, 2, \cdots, n$, being independent and identically distributed (iid) as $N(0, 1)$. As μ reflects the long run average return, it can be pre-estimated by the sample mean of y_t. This is equivalent to pre-centering the sample of observed returns and estimating the model without μ.

Park (2002) proposed a nonlinear nonstationary heteroscedastic (NNH) model that specifies the conditional volatility process as

$$\sigma_t^2 = g(x_{t-1}), \tag{5.2.2}$$

where x_{t-1} is an exogenous variable. In this model, $g(\cdot)$ is a nonnegative nonlinear function that comes from the class of integrable and asymptotically homogeneous functions discussed in Park & Phillips (1999, 2001).

Han & Park (2008) presented an extension to this NNH model by including the lagged squared error into the conditional volatility:

$$\sigma_t^2 = \alpha u_{t-1}^2 + g(x_{t-1}). \tag{5.2.3}$$

This model is called the ARCH-NNH model and allows x_t to have a unit root or a near unit root.

Han & Zhang (2012) presented a nonstationary nonparametric volatility (NNV) model, which is a nonparametric version of the NNH model given by

$$\sigma_t^2 = m(x_{t-1}), \tag{5.2.4}$$

where $m(\cdot)$ is an unknown smooth function and $m(x_t) > 0$ for all t. They suggested estimating $m(x_t)$ by the Nadaraya-Watson estimator defined in (2.5.12).

5.2.1 A semiparametric nonlinear volatility model

We propose to extend Han & Zhang (2012) NNV model by including the lagged squared error into the nonlinear volatility equation, and the resulting model is

$$\begin{aligned} y_t &= \sigma_t \varepsilon_t, \\ \sigma_t^2 &= \alpha y_{t-1}^2 + m(x_{t-1}), \end{aligned} \tag{5.2.5}$$

where $m(x_{t-1})$ is an unknown smooth function, and ε_t, for $t = 1, 2, \cdots, n$, are iid with mean 0 and variance 1. As discussed in. We call this model the semiparametric nonlinear volatility (SNV) model.

As the NNV model is a nonparametric version of the NNH model, the proposed SNV model can be interpreted as an semiparametric version of the ARCH-NNH model in (5.2.3), in which the nonnegative nonlinear function $g(x_{t-1})$ is specified to be an unknown smoothing function $m(x_{t-1})$. Further specification is provided below.

As $y_t = \sigma_t \varepsilon_t$ can be re-written as $y_t^2 = \sigma_t^2 \varepsilon_t^2$, we have

$$\begin{aligned} y_t^2 &= \sigma_t^2 + \sigma_t^2 \varepsilon_t^2 - \sigma_t^2 \\ &= \alpha y_{t-1}^2 + m(x_{t-1}) + \sigma_t^2(\varepsilon_i^2 - 1). \end{aligned} \tag{5.2.6}$$

Let $\eta_t = \sigma_t^2(\varepsilon_t^2 - 1)$ and express y_t^2 as

$$y_t^2 = \alpha y_{t-1}^2 + m(x_{t-1}) + \eta_t, \tag{5.2.7}$$

where $E(\eta_t) = 0$ due to the fact that $E\left(\varepsilon_t^2\right) = 1$. This equation can be re-expressed as

$$y_t^2 - \alpha y_{t-1}^2 = m(x_{t-1}) + \eta_t.$$

Therefore, conditional on α, we estimate $m(x_{t-1})$ by the leave-one-out NW estimator:

$$\widetilde{m}\left(x_t|h\right) = \frac{\sum_{i=1;i\neq t}^{n} K_h\left(x_i - x_t\right)\left(y_i^2 - \alpha y_{i-1}^2\right)}{\sum_{i=1,i\neq t}^{n} K_h\left(x_i - x_t\right)}, \tag{5.2.8}$$

where $K_h\left(\cdot\right) = K(\cdot/h)/h$ with $K(\cdot)$ being a kernel function and h a bandwidth. Moreover, it can be seen that if α is assumed to be zero, the proposed SNV model in (5.2.5) becomes the same as the NNV model proposed by Han & Zhang (2012).

Note that as discussed by Han & Zhang (2012), it is not necessary to assume the independence of ε_t and x_t. Han & Zhang (2012) provided the asymptotic distribution of $\tilde{m}$ to be mixed normal. In the proposed SNV model, it is also possible to derive the asymptotic distribution of α. However, as the focus of this study is to propose a new algorithm for selecting the optimal bandwidth, we leave it for future research.

5.2.2 Posterior of bandwidth parameters

In the ARCH-NNV model given by (5.2.4), Han & Zhang (2012) suggested using cross-validation to select a bandwidth that minimises the QLIKE loss function. However, this method cannot be used to choose bandwidth in our model because α is unknown in (5.2.8). In this paper, we overcome this problem by treating the bandwidth as a parameter and estimating μ, α and h simultaneously.

If the density of ε_t is known and denoted as $f_0(\varepsilon_t)$, and $m(\cdot)$ is known, the density of y_t will be

$$p\left(y_t\right) = f_0\left(\frac{y_t}{\sigma_t}\right)\frac{1}{\sigma_t},$$

where $\sigma_t^2 = \alpha y_{t-1}^2 + m(x_{t-1})$.

As $f_0(\varepsilon_t)$ is unknown, we propose to approximate it by a Gaussian kernel density given by

$$f\left(\varepsilon_t \middle| b\right) = \frac{1}{n-1}\sum_{i=1;i\neq t}^{n}\frac{1}{b}\phi\left(\frac{\varepsilon_t - \varepsilon_i}{b}\right),$$

which is a mixture of $(n-1)$ normal density functions with a common variance b^2 and individual mean values located at the corresponding errors. This Gaussian kernel

error density was proposed by Zhang & King (2013) for a GARCH(1,1) model without the exogenous variable in its conditional volatility equation.

As $m(x_{t-1})$ is also unknown, we propose to plug-in the leave-one-out NW estimator given by (5.2.8) into the conditional volatility equation of (5.2.5). Thus, the density of y_t is approximated by

$$\widetilde{p}\left(y_t|b\right) = \frac{1}{n-1}\sum_{i=1;i\neq t}^{n}\frac{1}{b\sigma_t}\phi\left(\frac{y_t/\sigma_t - y_i/\sigma_i}{b}\right),$$

where $\sigma_t^2 = \alpha y_{t-1}^2 + \widetilde{m}\left(x_{t-1}|h\right)$.

Let $\boldsymbol{\theta} = \left(b, \alpha, h\right)'$ denote the vector of parameters, and $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)'$ denote a vector of observed returns. The likelihood function of $\boldsymbol{y}$ given $\boldsymbol{\theta}$ is approximated as

$$\ell\left(\boldsymbol{y}|\boldsymbol{\theta}\right) = \prod_{t=1}^{n}\left\{\frac{1}{(n-1)\sigma_t}\sum_{\substack{i=1\\ i\neq t}}^{n}\frac{1}{b}\phi\left(\frac{u_t/\sigma_t - u_t/\sigma_i}{b}\right)\right\}, \tag{5.2.9}$$

where $\sigma_i^2 = \alpha y_{i-1}^2 + \widetilde{m}\left(x_{i-1}\right)$, for $i = 1, 2, \cdots, n$.

To facilitate posterior estimation of $\boldsymbol{\theta}$, we need to assume priors of these three parameters. The prior of α is assumed to be the uniform density defined on $(0,1)$, which reflects the restriction imposed on this parameter. The prior densities of h and b are assumed to be the Cauchy density expressed respectively as

$$\pi(h) = \frac{2}{\pi(1+h^2)}, \qquad \text{and} \qquad \pi(b) = \frac{2}{\pi(1+b^2)}.$$

The joint prior of $\boldsymbol{\theta}$ denoted as $\pi(\boldsymbol{\theta})$, is the product of these three marginal priors.

The posterior of θ is (up to a normalizing constant)

$$\pi\left(\boldsymbol{\theta}\middle|\boldsymbol{y}\right) \propto \pi\left(\boldsymbol{\theta}\right) \times \ell\left(\boldsymbol{y}|\boldsymbol{\theta}\right), \tag{5.2.10}$$

from which we sample $\boldsymbol{\theta}$ through the random-walk Metropolis algorithm. The sampling procedure is as follows.

Step I Choose arbitrary initial values for $\boldsymbol{\theta}$ denoted as $\boldsymbol{\theta}^{(0)}$ subject to constraints of parameters.

Step II Generate a random number v_k from $N(0,1)$ and update $\boldsymbol{\theta}_k^{(0)}$ through

$$\widetilde{\boldsymbol{\theta}}_k^{(1)} = \boldsymbol{\theta}_k^{(0)} + \tau_k v_k,$$

where τ_k is a tuning parameter, for $k = 1, 2$ and 3.

Step III Accept $\widetilde{\boldsymbol{\theta}}_k^{(1)}$ with a probability given by

$$\min\left\{1, \frac{\pi\left(\widetilde{\boldsymbol{\theta}}_k^{(1)}\middle|\boldsymbol{y}\right)}{\pi\left(\widetilde{\boldsymbol{\theta}}_k^{(0)}\middle|\boldsymbol{y}\right)}\right\}, \text{ for } k = 1, 2 \text{ and } 3.$$

Step IV If $\widetilde{\boldsymbol{\theta}}_k^{(1)}$ is accepted, let $\boldsymbol{\theta}_k^{(1)} = \widetilde{\boldsymbol{\theta}}_k^{(1)}$; otherwise, let $\boldsymbol{\theta}_k^{(1)} = \widetilde{\boldsymbol{\theta}}_k^{(0)}$.

Step V Repeat **Step II** to **Step IV** until the simulated chain $\left\{\boldsymbol{\theta}^{(i)} : i = 1, 2, \cdots\right\}$, achieves acceptable mixing performance.

Upon completion of this sampling procedure, the ergodic average of each simulated chain is an estimate of the corresponding component $\boldsymbol{\theta}$.

5.3 Model Comparison via stock-index Data

5.3.1 Data

We model daily stock-index returns, which are the S&P 500, All Ordinaries, DAX 30, Dow Jones industrial average, FTSE 100, Hong Kong Hang Seng, Nasdaq 100, and Nikkei 225. The exogenous variable is the S&P 500 VIX index, which is the implied volatility calculated based on S&P 500 index options by the Chicago Board Options Exchange (CBOE) and has been widely regarded as an index of investor sentiment and market volatility or the "fear index".

In the literature of GARCH models, it has been of great interest to study the impact of the VIX index on volatilities. It has been found that the VIX index can obviously contribute to volatility forecast (see, for example, Fleming et al. 1995, Blair et al. 2001, Kanas 2012, Han & Zhang 2012). Hence we adopt the VIX index in the purpose of volatility estimation in this study. It is generally accepted that financial series follows a unit root process, please see Han & Zhang (2012) for a unit root test of VIX.

The sample period is from 3 January 2007 to 31 May 2012 excluding non-trading days, where were downloaded from Datastream. As each country has different number of non-trading days, the total sample size varies depending on the series. The market indices used and their sample sizes are shown in Table 5.1.

Table 5.1: *Data series and sample sizes*

Name	Sample size
S&P 500	1363
ASX All Ordinaries (AORD)	1369
DAX 30 (DAX)	1379
Dow Jones Industrials (DJI)	1364
FTSE 100 (FTSE)	1369
Hong Kong Hang Seng (HKHS)	1336
NASDAQ 100 (NASDAQ)	1363
NIKKEI 225 (NIKKEI)	1325

5.3.2 Model Comparison

In order to evaluate the performance of the proposed SNV model, we compare the in-sample and out-sample forecasting performance of this SNV model with that of several competitive models.

Han & Zhang (2012) suggested using the cross-validation method, which minimises the QLIKE loss function to select an optimal bandwidth for the NNV model. The QLIKE loss function is given by

$$h_{\text{QLIKE}} = \arg\min_{h} \frac{1}{n} \sum_{t=1}^{n} \left\{ \frac{\sigma_t^2}{\widehat{m}\left(x_{t-1}\right)} - \log \frac{\sigma_t^2}{\widehat{m}\left(x_{t-1}\right)} - 1 \right\}, \tag{5.3.1}$$

where $\hat{m}(\cdot)$ is the leave-one-out estimator in (5.2.8) with α being set to zero.

As σ_t^2 is unobservable, a proxy of σ_t^2 has to be used. Two commonly used proxies are the squared return and realized volatility (see Hansen & Lunde 2006, Patton 2011, for example).

The squared return as a proxy of σ_t^2 has the benefit of being readily available, but is often volatile. Hansen & Lunde (2006) argued that the realized volatility is more

reliable than and preferable to the squared return. Han & Zhang (2012) used realized volatility as the proxy of σ_t^2.[1]

In addition to h_{QLIKE} obtained through (5.3.1), it is possible to choose bandwidth through least squares cross-validation (LSCV):

$$h_{\text{LSCV}} = \arg\min_h \frac{1}{n}\sum_{t=1}^{n}\left(\widehat{\sigma}_{t-1}^2 - \sigma_{t-1}^2\right)^2, \tag{5.3.2}$$

In our proposed SNV model, it is possible to simultaneously select α and h in (5.2.8) based on QLIKE and LSCV criteria:

$$\boldsymbol{\theta}_{\text{QLIKE}} = \arg\min_{\theta} \frac{1}{n}\sum_{t=1}^{n}\left\{\frac{\sigma_t^2}{\widehat{\sigma}_t^2} - \log\frac{\sigma_t^2}{\widehat{\sigma}_t^2} - 1\right\},$$
$$\boldsymbol{\theta}_{\text{LSCV}} = \arg\min_{\theta} \frac{1}{n}\sum_{t=1}^{n}\left(\widehat{\sigma}_t^2 - \sigma_t^2\right)^2,$$

where $\boldsymbol{\theta}$ is the parameter vector.

The models under comparison in this chapter are given below:

a) NNV model given by (5.2.4) with with its bandwidth chosen via QLIKE cross-validation.

b) SNV model given by (5.2.5) with bandwidth selected via QLIKE cross-validation.

c) NNV model with its bandwidth selected through LSCV.

d) SNV model with its bandwidth selected through LSCV.

[1] The realized volatility data are produced by the realized kernel method discussed in Barndorff-Nielsen et al. (2008). The data were downloaded from Oxford-Man Institute of Quantitative Finance's realized library by Heber, Lunde, Shephard & Sheppard (2009).

e) NNV model given by (5.2.4) with the standard Gaussian distribution of ε_t.

f) SNV model given by (5.2.5) with the standard Gaussian distribution of ε_t.

g) NNV model given by (5.2.4) with the Gaussian kernel density of ε_t.

h) SNV model given by (5.2.5) with the Gaussian kernel density of ε_t.

5.3.3 Evaluation Criterion

In order to evaluate the performance of the above listed volatility models, we employed nine loss functions shown in Patton (2011),

$$\text{QLIKE: } L(\hat{\sigma}_t^2, \sigma_t^2) = \frac{\sigma_t^2}{\hat{\sigma}_t^2} - \log \frac{\sigma_t^2}{\hat{\sigma}_t^2} - 1 \tag{5.3.3}$$

$$\text{MSE: } L(\hat{\sigma}^2, \sigma^2) = \left(\hat{\sigma}^2 - \sigma^2\right)^2 \tag{5.3.4}$$

$$\text{MAE: } L(\hat{\sigma}^2, \sigma^2) = \left|\hat{\sigma}^2 - \sigma^2\right| \tag{5.3.5}$$

$$\text{MSE-LOG: } L(\hat{\sigma}^2, \sigma^2) = \left(\log \hat{\sigma}^2 - \log \sigma^2\right)^2 \tag{5.3.6}$$

$$\text{MAE-LOG: } L(\hat{\sigma}^2, \sigma^2) = \left|\log \hat{\sigma}^2 - \log \sigma^2\right| \tag{5.3.7}$$

$$\text{MSE-SD: } L(\hat{\sigma}^2, \sigma^2) = (\log \hat{\sigma} - \log \sigma)^2 \tag{5.3.8}$$

$$\text{MAE-SD: } L(\hat{\sigma}^2, \sigma^2) = |\log \hat{\sigma} - \log \sigma| \tag{5.3.9}$$

$$\text{MSE-prop: } L(\hat{\sigma}^2, \sigma^2) = \left(\frac{\hat{\sigma}^2}{\sigma^2} - 1\right)^2 \tag{5.3.10}$$

$$\text{MAE-prop: } L(\hat{\sigma}^2, \sigma^2) = \left|\frac{\hat{\sigma}^2}{\sigma^2} - 1\right| \tag{5.3.11}$$

Patton (2011) discussed the necessary and sufficient conditions on the functional form of a loss function in order to obtain robust ranking of forecasts derived under different volatility models. Among these nine loss functions, MSE and QLIKE were found to be

the most robust. However, our proposed SNV model is estimated through Bayesian sampling, and Model *a* to Model *d* will enjoy a certain degree of home-team advantage under the MSE and QLIKE criteria. Therefore, in addition to the above nine loss functions, we consider a neutral evaluation criterion described as follows.

Value-at-risk (VaR) is a commonly used risk measure which reveals the maximum possible decrease in the value of a portfolio at a pre-determined confidence level. Given the stochastic process u_t in (5.2.1), the conditional distribution function of ε_t is denoted as $F\left(\varepsilon_t\right) = P\left\{\varepsilon_t \leq \varepsilon | I_{t-1}\right\}$, where I_{t-1} is the information set at time $t-1$. At the confidence level of $100(1-\gamma)\%$, the VaR denoted by $q_t\left(\gamma\right)$ is calculated as

$$q_t\left(\gamma\right) = \mu + \sigma_t F^{-1}\left(\gamma\right). \tag{5.3.12}$$

To estimate the VaR forecasts, $F(\cdot)$ needs to be specified or estimated. For Model *e* and Model *f*, $F(\cdot)$ is the standard Gaussian distribution function. For Model *g* and Model *h*, the density of ε is unknown but is approximated by the Gaussian kernel density. This enabled us to estimate $F(\cdot)$ based on residuals calculated through $\widehat{\varepsilon_t} = u_t/\sigma_t$, for $t = 1, 2, \cdots, n$.

It is noted that VaR forecast based on SNV model was not discussed in Han & Zhang (2012). The SNV model is fully nonparametric and does not assume any particular distribution function for ε. Therefore, we used the kernel density estimation technique to obtain the distribution function of ε. The bandwidth is selected based on likelihood cross-validation method discussed in Härdle (1990) and Pagan & Ullah (1999).

To evaluate the VaR forecasting performance, we followed a back-testing procedure discussed by Bao, Lee & Saltoglu (2006) and calculate the empirical coverage:

$$\widehat{\gamma} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\Big\{r_t < \widehat{q}_t\big(\gamma\big)\Big\}, \tag{5.3.13}$$

where T denotes the number of observations during the out-of-sample period, $\widehat{q}_t(\gamma)$ is the VaR forecast with the nominal coverage being γ, and $\mathbf{1}\{\cdot\}$ is an indicator function and equals 1 for a true argument and 0 otherwise.

We also calculated the quantile loss function, which is also known as the check function defined by Koenker & Bassett Jr (1978):

$$Q(\gamma) = E\Big[\gamma - \mathbf{1}\Big\{r_t < q_t\big(\gamma\big)\Big\}\Big]\Big[r_t - q_t\big(\gamma\big)\Big]. \tag{5.3.14}$$

The check function is calculated based on VaR forecasts during the out-of-sample period:

$$\widehat{Q}\big(\gamma\big) = \frac{1}{T}\sum_{i=n+1}^{N} \Big[\gamma - \mathbf{1}\Big\{r_t < \widehat{q}_t\big(\gamma\big)\Big\}\Big]\Big[r_t - \widehat{q}_t\big(\gamma\big)\Big]. \tag{5.3.15}$$

5.4 Performance Evaluation Results

For each return series, the sample was divided into in-sample and out-of-sample periods, where first 1,000 observations were included in the in-sample period and used for estimation. A one-day-ahead forecast was made. Rolling the sample for estimation forward for one day, we estimated the model and made a one-day-ahead forecast.

Table 5.2: *MCMC results of Model* g *based on In-sample S&P500 data*

	Parameters	Mean	Standard deviation	Batch-mean standard deviation	SIF	Acceptance rate
Model g	α	0.2740	0.0419	0.0012	8.57	0.25
	h	4.2272	0.2879	0.0083	8.35	0.32
	b	0.0177	0.0161	0.0004	6.06	0.31

This rolling-sample procedure continues until the last observation during the out-of-sample period was forecast. During both in- and out-of-sample periods, we estimated Model a to Model h and calculated the loss functions.

We applied the random-walk Metropolis-Hastings algorithm to the update each parameter with the acceptance probability calculated through (5.2.10). The algorithm has a burn-in period of 3,000 iterations and a recorded period of 10,000 iterations. The batch-mean standard deviation in Roberts (1996) and the simulation inefficient factor (SIF) shown in Kim et al. (1998) are computed to examine the mixing performance. As the simulated chain is a Markov chain, the SIF value can be roughly interpreted as the number of draws needed so as to produce independent draws. A small SIF value usually indicate good mixing performance. It can be observed that the value of α is clearly not zero, which means the proposed SNV model distinguishes itself from the NNV model proposed by Han & Zhang (2012) where α is set to zero.

The plot of the sample path of each parameter, together with its autocorrelation function (ACF) and histogram graphs is also presented for visual inspection of the mixing performance.

To demonstrate the mixing performance, we take the estimation of Model Model g as an example. We implemented the sampling algorithm to Model G with a sample

Figure 5.1: *Plots of posterior draws obtained through our proposed sampling algorithm for Model g based on In-sample S&P500 data: (a) α (b) h (c) b*

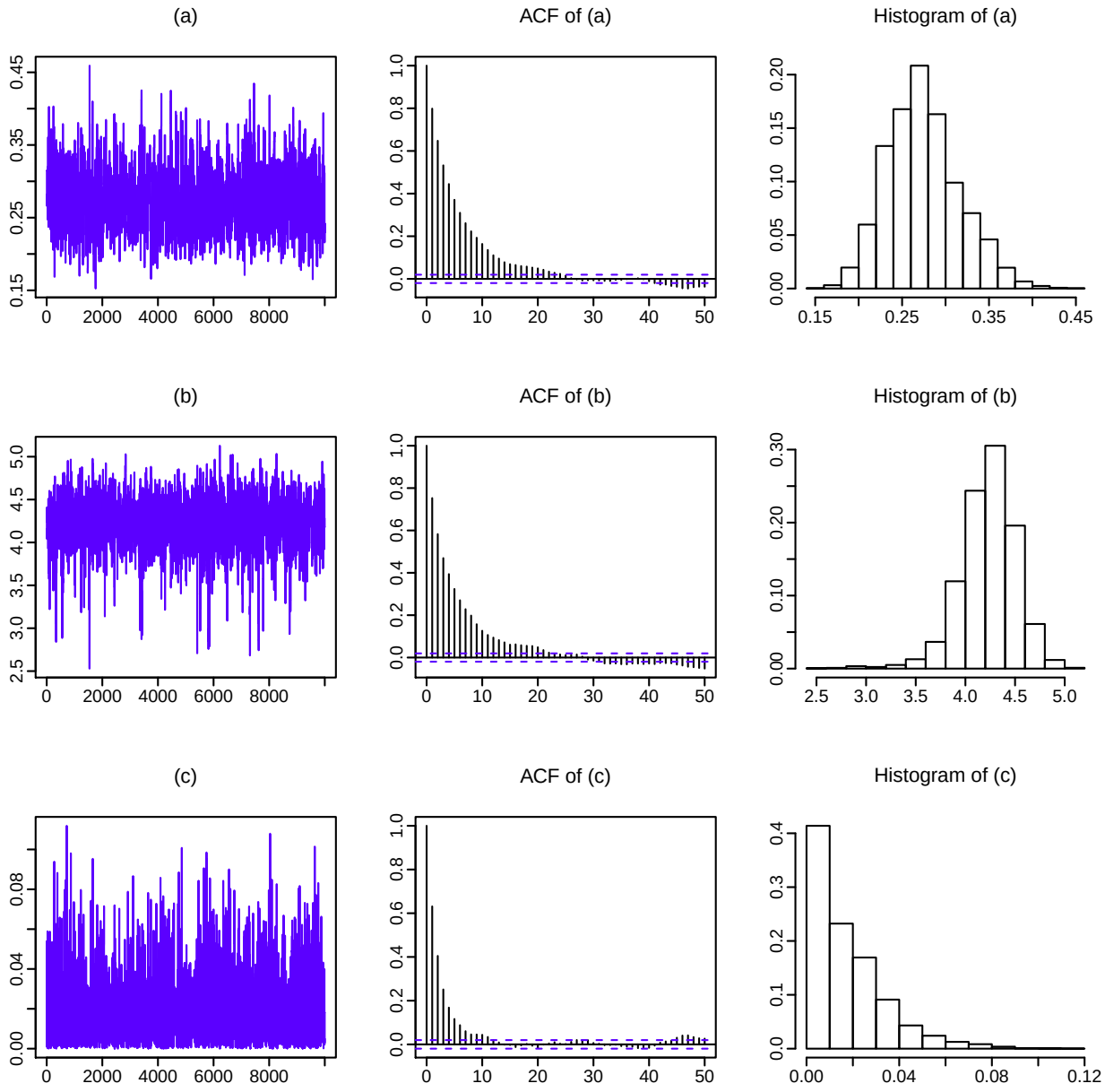


Table 5.3: *In-sample performance evaluation of volatility models for stock returns (Part One)*

	QLIKE		LSCV		Bayesian			
	NNV	SNV	NNV	SNV	NNV	SNV	NNV	SNV
Model	a	b	c	d	e	f	g	h
S&P500								
QLIKE	0.2569	0.2208	0.4878	0.4134	0.2622	0.2610	0.2858	0.2829
MSE	18.2302	14.7277	15.0996	11.4952	15.6401	15.3840	14.4595	14.1054
MAE	1.6430	1.5246	1.6349	1.3933	1.5585	1.5490	1.5062	1.4923
MSE-LOG	0.6079	0.5581	1.4401	1.2243	0.6368	0.6336	0.7212	0.7130
MAE-LOG	0.6433	0.6118	0.9854	0.8957	0.6624	0.6602	0.7027	0.6976
MSE-SD	0.3889	0.3354	0.4878	0.3687	0.3591	0.3548	0.3508	0.3446
MAE-SD	0.4202	0.3949	0.5566	0.4876	0.4177	0.4158	0.4235	0.4201
MSE-prop	0.7954	0.4069	0.9328	0.6465	0.7467	0.7496	0.7389	0.7414
MAE-prop	0.5225	0.4719	0.6505	0.5876	0.5225	0.5205	0.5327	0.5286
AORD								
QLIKE	0.4503	0.4192	0.5198	0.5120	0.4588	0.4475	0.4533	0.4391
MSE	5.1477	7.3569	2.9214	2.8300	4.2209	3.7434	4.6345	4.3450
MAE	1.2953	1.3078	1.2133	1.1990	1.2588	1.2146	1.2805	1.2486
MSE-LOG	1.3603	1.2450	1.6182	1.5994	1.3941	1.3580	1.3753	1.3256
MAE-LOG	1.0021	0.9261	1.1206	1.1136	1.0279	1.0166	1.0154	0.9948
MSE-SD	0.4091	0.4253	0.3576	0.3485	0.3812	0.3578	0.3952	0.3800
MAE-SD	0.5046	0.4762	0.5314	0.5265	0.5071	0.4966	0.5065	0.4953
MSE-prop	0.5370	0.5676	0.5288	0.4983	0.4990	0.4699	0.5076	0.4898
MAE-prop	0.6111	0.5988	0.6464	0.6392	0.6147	0.6062	0.6116	0.6019
DAX								
QLIKE	0.2085	0.1853	0.3345	0.2397	0.2101	0.2026	0.2141	0.2032
MSE	12.7519	12.6825	11.8034	9.5490	13.7176	13.3987	14.5061	14.1982
MAE	1.3710	1.3540	1.3482	1.2452	1.4048	1.3898	1.4246	1.4047
MSE-LOG	0.5141	0.4705	0.9078	0.6333	0.5123	0.5036	0.5157	0.5014
MAE-LOG	0.5753	0.5518	0.7571	0.6299	0.5731	0.5689	0.5734	0.5659
MSE-SD	0.3070	0.2874	0.3616	0.2658	0.3247	0.3154	0.3389	0.3273
MAE-SD	0.3748	0.3639	0.4491	0.3791	0.3786	0.3751	0.3815	0.3760
MSE-prop	0.4995	0.3040	0.7433	0.4475	0.5213	0.4227	0.5717	0.4535
MAE-prop	0.4509	0.4285	0.5420	0.4601	0.4553	0.4458	0.4611	0.4471
DJI								
QLIKE	0.2437	0.2041	0.4562	0.2692	0.2461	0.2448	0.2672	0.2641
MSE	13.5443	11.9881	16.5141	10.0750	12.9530	12.7301	12.1741	11.9023
MAE	1.2494	1.2299	1.5173	1.1572	1.2449	1.2370	1.2234	1.2133
MSE-LOG	0.5086	0.4632	1.2132	0.6567	0.5252	0.5220	0.6008	0.5925
MAE-LOG	0.5712	0.5447	0.8927	0.6459	0.5826	0.5807	0.6228	0.6182
MSE-SD	0.2922	0.2637	0.4594	0.2514	0.2856	0.2816	0.2810	0.2757
MAE-SD	0.3468	0.3336	0.4996	0.3557	0.3493	0.3476	0.3576	0.3549
MSE-prop	1.2352	0.5766	1.3643	0.7520	1.1768	1.1864	1.1330	1.1419
MAE-prop	0.5184	0.4681	0.6720	0.5170	0.5174	0.5154	0.5287	0.5250

of 1,000 observation of the S&P 500 daily return. Figure 5.1 presents graphs of the path of the simulated chain, auto correlation function (ACF) and histogram of each

Table 5.4: *In-sample performance evaluation of volatility models for stock returns (Part Two)*

	QLIKE		LSCV		Bayesian			
	NNV	SNV	NNV	SNV	NNV	SNV	NNV	SNV
Model	a	b	c	d	e	f	g	h
FTSE								
QLIKE	0.2955	0.2719	0.3624	0.2737	0.2992	0.2931	0.2972	0.2919
MSE	12.1725	11.7555	6.7436	10.9763	9.9769	9.9625	10.3623	10.3675
MAE	1.4094	1.4015	1.3584	1.3983	1.4060	1.3891	1.4049	1.3907
MSE-LOG	0.8440	0.7724	1.0566	0.7766	0.8529	0.8403	0.8479	0.8369
MAE-LOG	0.7595	0.7267	0.8774	0.7311	0.7709	0.7643	0.7658	0.7600
MSE-SD	0.3992	0.3810	0.3476	0.3739	0.3795	0.3727	0.3838	0.3781
MAE-SD	0.4440	0.4299	0.4778	0.4313	0.4481	0.4430	0.4463	0.4420
MSE-prop	0.3770	0.3289	0.4766	0.3360	0.3897	0.3464	0.3852	0.3479
MAE-prop	0.5020	0.4850	0.5557	0.4868	0.5063	0.4989	0.5034	0.4970
HKHS								
QLIKE	0.5468	0.4425	0.6444	0.6267	0.5492	0.5292	0.5484	0.5293
MSE	38.3661	27.3490	13.9994	12.4173	28.5265	26.5870	29.8416	24.8137
MAE	2.6008	2.3328	2.4882	2.4025	2.5317	2.4551	2.5402	2.4375
MSE-LOG	1.7779	1.3716	2.1164	2.0639	1.7865	1.7220	1.7841	1.7212
MAE-LOG	1.1485	0.9580	1.2954	1.2806	1.1549	1.1351	1.1533	1.1362
MSE-SD	0.9448	0.7986	0.8561	0.8046	0.9009	0.8438	0.9075	0.8304
MAE-SD	0.7555	0.6413	0.8192	0.8023	0.7542	0.7357	0.7539	0.7347
MSE-prop	0.5197	0.5273	0.6022	0.5389	0.5150	0.4639	0.5148	0.4632
MAE-prop	0.6400	0.5916	0.6922	0.6781	0.6411	0.6256	0.6405	0.6259
NASDAQ								
QLIKE	0.3503	0.3108	0.5448	0.4049	0.3565	0.3543	0.3633	0.3606
MSE	20.0362	21.2053	6.7512	8.2828	16.0591	15.6460	13.4840	13.2775
MAE	1.9614	1.8772	1.6508	1.6821	1.9157	1.9004	1.8857	1.8709
MSE-LOG	1.0153	0.8970	1.7537	1.2254	1.0346	1.0280	1.0581	1.0490
MAE-LOG	0.8514	0.7717	1.1162	0.9502	0.8708	0.8686	0.8851	0.8812
MSE-SD	0.5748	0.5447	0.5124	0.4276	0.5329	0.5248	0.5051	0.4992
MAE-SD	0.5439	0.5025	0.6252	0.5515	0.5483	0.5458	0.5516	0.5484
MSE-prop	0.4687	0.4185	0.6044	0.4403	0.4514	0.4468	0.4561	0.4544
MAE-prop	0.5469	0.5140	0.6360	0.5770	0.5547	0.5530	0.5603	0.5580
NIKKEI								
QLIKE	0.4440	0.4983	0.4993	0.4983	0.4473	0.4466	0.4478	0.4467
MSE	26.5937	15.7158	14.2491	15.7158	25.1156	24.5380	25.0473	24.3438
MAE	2.2351	2.0877	2.0440	2.0877	2.2425	2.2399	2.2441	2.2401
MSE-LOG	1.3791	1.5676	1.5715	1.5676	1.3878	1.3838	1.3891	1.3834
MAE-LOG	1.0177	1.1070	1.1078	1.1070	1.0266	1.0271	1.0276	1.0277
MSE-SD	0.7335	0.6388	0.6192	0.6388	0.7180	0.7095	0.7173	0.7066
MAE-SD	0.6393	0.6625	0.6587	0.6625	0.6444	0.6442	0.6449	0.6445
MSE-prop	0.4076	0.4451	0.4471	0.4451	0.4086	0.4074	0.4090	0.4074
MAE-prop	0.5905	0.6235	0.6234	0.6235	0.5934	0.5934	0.5938	0.5937

parameter. Table 5.2 presents a summary of the MCMC results. It is shown that for

Table 5.5: *Out-of-sample performance evaluation of volatility models for stock returns (Part One)*

	QLIKE		LSCV		Bayesian			
	NNV	SNV	NNV	SNV	NNV	SNV	NNV	SNV
Model	a	b	c	d	e	f	g	h
S&P500								
QLIKE	0.2284	0.2069	0.6426	0.5108	0.2368	0.2320	0.2469	0.2421
MSE	1.9302	1.6116	2.9737	2.2026	1.9224	1.8482	2.0062	1.9358
MAE	0.8054	0.7445	1.3909	1.0781	0.8104	0.7960	0.8322	0.8195
MSE-LOG	0.6153	0.5571	2.1523	1.6531	0.6510	0.6382	0.6883	0.6755
MAE-LOG	0.6403	0.6033	1.2632	1.0690	0.6582	0.6504	0.6687	0.6612
MSE-SD	0.1696	0.1459	0.4670	0.3259	0.1716	0.1663	0.1813	0.1760
MAE-SD	0.3140	0.2924	0.6197	0.4974	0.3198	0.3150	0.3269	0.3225
MSE-prop	0.3172	0.2778	0.6108	0.5183	0.3107	0.3010	0.3264	0.3158
MAE-prop	0.4733	0.4442	0.6824	0.6155	0.4735	0.4674	0.4812	0.4745
AORD								
QLIKE	0.4661	0.3863	0.6407	0.6307	0.4852	0.4865	0.4722	0.4596
MSE	1.3882	1.1948	0.9427	0.8802	1.2555	1.1327	1.3111	1.2584
MAE	0.8028	0.7256	0.8864	0.8682	0.8064	0.7929	0.8012	0.7847
MSE-LOG	1.4742	1.1455	2.1417	2.0964	1.5366	1.5348	1.4911	1.4414
MAE-LOG	1.0458	0.9522	1.3156	1.3053	1.0805	1.0900	1.0579	1.0433
MSE-SD	0.2665	0.2044	0.2961	0.2841	0.2623	0.2508	0.2627	0.2528
MAE-SD	0.4171	0.3759	0.5102	0.5033	0.4274	0.4274	0.4202	0.4129
MSE-prop	0.4026	0.3686	0.4972	0.4943	0.4142	0.4158	0.4063	0.3980
MAE-prop	0.5873	0.5702	0.6766	0.6757	0.6004	0.6053	0.5916	0.5865
DAX								
QLIKE	0.1628	0.1651	0.3712	0.2996	0.1645	0.1566	0.1646	0.1645
MSE	3.9370	4.0679	5.7013	4.7543	3.9859	3.8557	4.0050	3.9859
MAE	0.7946	0.8756	1.3575	1.0575	0.8003	0.7932	0.8091	0.8003
MSE-LOG	0.3555	0.3539	0.9666	0.6528	0.3516	0.3379	0.3515	0.3516
MAE-LOG	0.4387	0.4520	0.7907	0.6214	0.4375	0.4317	0.4373	0.4375
MSE-SD	0.1603	0.1743	0.3637	0.2499	0.1625	0.1562	0.1639	0.1625
MAE-SD	0.2572	0.2750	0.4760	0.3634	0.2579	0.2549	0.2594	0.2579
MSE-prop	0.4535	0.4946	0.8999	1.5224	0.4781	0.4477	0.4792	0.4781
MAE-prop	0.4178	0.4300	0.6156	0.5647	0.4272	0.4144	0.4280	0.4272
DJI								
QLIKE	0.2289	0.2104	0.5904	0.4081	0.2318	0.2148	0.2454	0.2274
MSE	1.2797	1.1058	2.2880	1.4664	1.2888	1.1372	1.3126	1.1600
MAE	0.6510	0.6072	1.1635	0.8192	0.6580	0.6191	0.6762	0.6360
MSE-LOG	0.5629	0.5196	1.9064	1.2164	0.5825	0.5404	0.6318	0.5863
MAE-LOG	0.6155	0.5854	1.1807	0.9167	0.6265	0.5986	0.6466	0.6166
MSE-SD	0.1302	0.1163	0.3746	0.2159	0.1324	0.1199	0.1396	0.1265
MAE-SD	0.2787	0.2625	0.5466	0.3977	0.2830	0.2683	0.2922	0.2767
MSE-prop	0.4362	0.3828	0.6305	0.4939	0.4112	0.3697	0.4181	0.3782
MAE-prop	0.4974	0.4697	0.6748	0.5932	0.4961	0.4728	0.5043	0.4796

each parameter, the corresponding SIF value is very small, and the batch-mean standard deviation is much smaller than the standard deviation. Therefore, the simulated chains have achieved very good mixing status.

Table 5.6: *Out-of-sample performance evaluation of volatility models for stock returns (Part Two)*

	QLIKE		LSCV		Bayesian			
	NNV	SNV	NNV	SNV	NNV	SNV	NNV	SNV
Model	a	b	c	d	e	f	g	h
FTSE								
QLIKE	0.3166	0.3037	0.4875	0.3082	0.3283	0.3244	0.3246	0.3203
MSE	1.9455	1.8556	1.4376	1.7195	1.6827	1.6346	1.7118	1.6738
MAE	0.8886	0.8636	1.0237	0.8582	0.8794	0.8731	0.8773	0.8718
MSE-LOG	0.9027	0.8630	1.5014	0.8759	0.9363	0.9232	0.9241	0.9095
MAE-LOG	0.8329	0.8101	1.1094	0.8226	0.8593	0.8564	0.8519	0.8481
MSE-SD	0.2346	0.2234	0.2853	0.2172	0.2237	0.2189	0.2240	0.2196
MAE-SD	0.3912	0.3802	0.5035	0.3833	0.3979	0.3960	0.3953	0.3932
MSE-prop	0.3510	0.3366	0.4637	0.3374	0.3597	0.3540	0.3590	0.3533
MAE-prop	0.5343	0.5229	0.6388	0.5288	0.5479	0.5471	0.5448	0.5438
HKHS								
QLIKE	0.5774	0.4351	0.7795	0.8089	0.5895	0.5807	0.5905	0.5850
MSE	7.1413	4.9808	5.3420	5.5694	6.7686	6.3058	6.8244	6.2456
MAE	1.8653	1.3092	2.1244	2.1332	1.8700	1.8172	1.8751	1.8208
MSE-LOG	1.8776	1.3666	2.5974	2.6343	1.9139	1.8797	1.9171	1.8934
MAE-LOG	1.2311	0.9601	1.5382	1.5362	1.2531	1.2411	1.2555	1.2490
MSE-SD	0.7155	0.4841	0.8012	0.8227	0.7096	0.6817	0.7121	0.6819
MAE-SD	0.6990	0.5142	0.8610	0.8592	0.7082	0.6961	0.7097	0.6996
MSE-prop	0.4734	0.4545	0.5944	0.8631	0.4819	0.4755	0.4829	0.4787
MAE-prop	0.6584	0.5862	0.7613	0.8062	0.6669	0.6624	0.6683	0.6657
NASDAQ								
QLIKE	0.3834	0.3466	0.8371	0.6281	0.4092	0.4041	0.4121	0.4068
MSE	2.3723	3.3286	3.3016	1.9133	2.2610	2.1803	2.3532	2.2610
MAE	1.0763	1.0031	1.7270	1.2553	1.0973	1.0811	1.1073	1.0902
MSE-LOG	1.1453	1.0572	2.9441	2.0699	1.2276	1.2101	1.2496	1.2298
MAE-LOG	0.9441	0.8531	1.5673	1.3003	0.9980	0.9893	0.9930	0.9857
MSE-SD	0.2928	0.2970	0.6646	0.4081	0.2949	0.2875	0.3032	0.2947
MAE-SD	0.4553	0.4116	0.7794	0.6029	0.4742	0.4686	0.4749	0.4695
MSE-prop	0.3706	0.3301	0.6129	0.5106	0.3914	0.3855	0.3876	0.3822
MAE-prop	0.5687	0.5179	0.7481	0.6826	0.5956	0.5904	0.5875	0.5839
NIKKEI								
QLIKE	0.7122	0.5052	0.8692	0.7897	0.7317	0.6906	0.7350	0.6871
MSE	6.5265	13.9616	6.5213	4.7150	6.2636	5.0411	6.2675	5.0536
MAE	1.6694	1.3939	1.9329	1.7749	1.6871	1.6233	1.6930	1.6182
MSE-LOG	2.3192	1.6149	2.8654	2.6568	2.3796	2.3142	2.3904	2.3032
MAE-LOG	1.3497	1.0781	1.5967	1.5294	1.3903	1.3741	1.3964	1.3674
MSE-SD	0.6873	0.4812	0.7853	0.6800	0.6846	0.6278	0.6866	0.6267
MAE-SD	0.6931	0.5347	0.8242	0.7741	0.7099	0.6937	0.7129	0.6906
MSE-prop	1.0077	0.4783	1.1212	0.6280	1.0096	0.5607	1.0108	0.5585
MAE-prop	0.7319	0.6121	0.8194	0.7630	0.7484	0.7078	0.7509	0.7050

Since the values of α and the bandwidth has not particular meanings by themselves and is not an important factor in model comparison, we have not reported the parameter values in the following sections.

Table 5.7: *Empirical coverage of VaR forecast by volatility models for stock returns*

		QLIKE		LSCV		Bayesian			
		NNV	SNV	NNV	SNV	NNV	SNV	NNV	SNV
Model		a	b	c	d	e	f	g	h
10%	S&P500	0.1077	0.1105	0.1105	0.1105	0.0967	0.0967	0.1077	0.1077
	AORD	0.1033	0.1359	0.1033	0.1141	0.0870	0.0842	0.1168	0.1223
	DAX	0.1561	0.1667	0.1667	0.1561	0.1429	0.1402	0.1561	0.1561
	DJI	0.0854	0.1019	0.1047	0.1047	0.0937	0.0937	0.0882	0.0882
	FTSE	0.1196	0.1304	0.1196	0.1304	0.1114	0.1087	0.1223	0.1196
	HKHS	0.1642	0.2000	0.1761	0.1910	0.0716	0.0746	0.1701	0.1731
	NASDAQ	0.1271	0.1381	0.1326	0.1326	0.0856	0.0912	0.1271	0.1271
	NIKKEI	0.1204	0.1636	0.1358	0.1420	0.0586	0.0586	0.1389	0.1420
5%	S&P500	0.0580	0.0674	0.0691	0.0718	0.0608	0.0608	0.0580	0.0580
	AORD	0.0489	0.0761	0.0489	0.0571	0.0598	0.0598	0.0571	0.0625
	DAX	0.1138	0.1323	0.1270	0.1270	0.0926	0.0952	0.1138	0.1138
	DJI	0.0523	0.0579	0.0551	0.0551	0.0634	0.0634	0.0523	0.0523
	FTSE	0.0679	0.0842	0.0707	0.0870	0.0679	0.0679	0.0734	0.0734
	HKHS	0.1045	0.1224	0.1075	0.1164	0.0448	0.0448	0.1075	0.1104
	NASDAQ	0.0663	0.1077	0.0718	0.0746	0.0580	0.0608	0.0663	0.0663
	NIKKEI	0.0556	0.1080	0.0617	0.0833	0.0340	0.0340	0.0648	0.0895
1%	S&P500	0.0166	0.0387	0.0193	0.0276	0.0193	0.0193	0.0166	0.0166
	AORD	0.0136	0.0272	0.0190	0.0190	0.0109	0.0109	0.0190	0.0190
	DAX	0.0556	0.0635	0.0529	0.0582	0.0423	0.0423	0.0556	0.0556
	DJI	0.0165	0.0248	0.0165	0.0193	0.0138	0.0165	0.0165	0.0165
	FTSE	0.0190	0.0299	0.0217	0.0299	0.0136	0.0163	0.0217	0.0190
	HKHS	0.0478	0.0597	0.0478	0.0478	0.0090	0.0090	0.0478	0.0478
	NASDAQ	0.0193	0.0525	0.0193	0.0359	0.0221	0.0276	0.0221	0.0221
	NIKKEI	0.0185	0.0432	0.0154	0.0216	0.0123	0.0123	0.0185	0.0247

5.4.1 In-sample Performance Comparison

In-sample values of loss functions are calculated for each return series, and the results are presented in Tables 5.3 and 5.4. Model *a* and *b* are both estimated through the QLIKE cross-validation method. It is expected that these models to perform well under QLIKE loss criterion. As expected, Model *c* and Model *d* perform the best under the MSE because they were estimated under LSCV.

The NNV model with ARCH component Model *b* outperforms NNV model Model *a* in most cases. Similarly, Model *d* with linear ARCH component outperforms Model

Table 5.8: *Predictive quantile loss of VaR forecast by volatility models for stock returns*

		QLIKE		LSCV		Bayesian			
		NNV	SNV	NNV	SNV	NNV	SNV	NNV	SNV
Model		a	b	c	d	e	f	g	h
10%	S&P500	0.2514	0.2528	0.2531	0.2526	0.2276	0.2275	0.2515	0.2514
	AORD	0.2010	0.2070	0.2016	0.2029	0.1902	0.1903	0.2023	0.2040
	DAX	0.3298	0.3361	0.3366	0.3339	0.2963	0.2967	0.3309	0.3304
	DJI	0.2271	0.2263	0.2266	0.2264	0.2068	0.2066	0.2271	0.2271
	FTSE	0.2332	0.2358	0.2336	0.2359	0.2187	0.2188	0.2343	0.2341
	HKHS	0.2913	0.3076	0.2927	0.2983	0.2587	0.2589	0.2930	0.2940
	NASDAQ	0.2576	0.2657	0.2602	0.2604	0.2414	0.2415	0.2579	0.2577
	NIKKEI	0.2472	0.2586	0.2485	0.2512	0.2466	0.2448	0.2494	0.2522
5%	S&P500	0.1621	0.1663	0.1643	0.1646	0.1429	0.1429	0.1623	0.1622
	AORD	0.1240	0.1284	0.1245	0.1252	0.1146	0.1151	0.1249	0.1259
	DAX	0.2207	0.2323	0.2285	0.2285	0.1862	0.1864	0.2225	0.2214
	DJI	0.1482	0.1489	0.1486	0.1483	0.1272	0.1272	0.1482	0.1482
	FTSE	0.1454	0.1500	0.1455	0.1499	0.1317	0.1320	0.1463	0.1460
	HKHS	0.1981	0.2143	0.1985	0.2039	0.1554	0.1561	0.1997	0.2010
	NASDAQ	0.1645	0.1758	0.1662	0.1672	0.1521	0.1517	0.1648	0.1647
	NIKKEI	0.1615	0.1729	0.1615	0.1634	0.1638	0.1615	0.1625	0.1646
1%	S&P500	0.0562	0.0674	0.0576	0.0618	0.0438	0.0440	0.0562	0.0561
	AORD	0.0352	0.0403	0.0362	0.0365	0.0264	0.0269	0.0364	0.0373
	DAX	0.0890	0.1007	0.0828	0.0952	0.0587	0.0575	0.0901	0.0900
	DJI	0.0507	0.0557	0.0507	0.0536	0.0378	0.0371	0.0509	0.0507
	FTSE	0.0452	0.0480	0.0456	0.0490	0.0343	0.0342	0.0457	0.0456
	HKHS	0.0827	0.1020	0.0812	0.0846	0.0390	0.0385	0.0848	0.0859
	NASDAQ	0.0563	0.0722	0.0577	0.0621	0.0478	0.0484	0.0569	0.0569
	NIKKEI	0.0689	0.0762	0.0684	0.0708	0.0689	0.0645	0.0706	0.0719

c. Improvements by including a linear ARCH component in Bayesian SNV and NNV models (Model f to h) can also be consistently observed. Model e to h are estimated through Bayesian sampling and performed the second best under both QLIKE and MSE measures and only slightly worse than the corresponding home-team models. This suggests Bayesian method is able to return stable performance and no worse than the QLIKE and LSCV methods.

We note that in-sample performance between models with standard Gaussian density (Model e and f) and Gaussian kernel density function (Model g and h) are mixed. It is also important to emphasise that the in-sample performance of the proposed SNV

models (Model e and g) are able to produce similar accuracy as NNV models estimated under QLIKE, Model a and b, based on additional realized volatility information.

5.4.2 Out-of-sample Performance Comparison

The out-of-sample period contains observations after the first 1,000 observation of each return series are used as out-of-sample period. The one-step-ahead forecast technique is performed, in which we use a uniform window of sample size 1,000 for all data series. Tables 5.5 and 5.6 present the loss function values calculated based on observations in the out-of-sample period for each return series.

For almost all cases, models with a linear ARCH component in the conditional variance equations such as Model b, d, f and h, outperformed models without such a component under the same estimation method. This finding is consistent with in-sample evaluation shown in Tables 5.2 and 5.3. Therefore, the inclusion of the linear ARCH component leads to better volatility forecasts and is empirically justified.

Forecasts based on QLIKE method (Model a and b) produced the lowest QLIKE loss measure in the out-of-sample period. Note that models estimated through QLIKE methods relies on information from both the VIX index and the realized volatility information which caused the QLIKE measure to be naturally biased. However, the LSCV method did not always produce the lowest MSE loss measure (i.e in DAX, DJIA and NASDAQ return series). This suggests the LSCV method is less reliable comparing to QLIKE method. When additional realised volatility information is available, QLIKE method is preferred over LSCV.

Bayesian sampling, which uses the same amount of information as the LSCV method, performs strongly in both in-sample and out-of-sample periods. The QLIKE loss measures of Model e to g are very close to those estimated through the QLIKE method. Bayesian method produced lower MSE loss measures than LSCV method in cases such as S&P 500, DAX, DJIA, NASDAQ and NIKKEI. In NASDAQ and NIKKEI, Bayesian sampling even performed the best under the MSE loss measure among all three methods.

5.4.3 Value-at-risk Performance Comparison

Our proposed SNV model is estimated through Bayesian sampling, and Model a to Model d will enjoy a certain degree of home-team advantage under the MSE and QLIKE criteria. Therefore, in addition to the above nine loss functions, we VaR as a neutral evaluation criterion. We note that this is essential a back-testing technique commonly used in financial econometric analysis. Back-testing is a way to examine the performance of a model or strategy on past data. As mentioned above, the out-of-sample period contains observations after the first $1,000$ observation of each return series are used as out-of-sample period. The one-step-ahead forecast is estimated based on a uniform window of sample size $1,000$ for all data series, which is then compared against actual past data consistent with the back-testing technique.

Table 5.7 presents the empirical coverage $\hat{\gamma}$ at nominal rates of $\gamma = 10\%, 5\%$, and 1%.

For the three levels of nominal rates, the calculated empirical coverages from all models are very close to the nominal rates for S&P 500, AORD, DJI, FTSE and NASDAQ.

All models under-estimated the VaR of DAX returns. Under QLIKE and LSCV methods, models with a linear ARCH component (Model b and d) under-estimated VaR in most cases.

For HKHS and NIKKEI series, the Gaussian kernel density (Model g and h) tend to under-estimate the VaR. Model e and f with a standard Gaussian density over-estimated VaR in NIKKEI return series.

Among the models estimated through Bayesian sampling, the standard Gaussian density Model e and f produced marginally more accurate empirical coverage than models with Gaussian kernel density Model g and h.

Based on the VaR forecast produced by competing volatility models, Table 5.8 presents the results of their quantile loss defined in (5.3.14). Coinciding with the performance shown in empirical coverage evaluation, Model b and d produced slightly higher quantile loss than their counterparts (Model a and c).

The proposed Bayesian models with standard Gaussian density (Model e and f) produced the lowest quantile loss in almost all cases across all nominal rates. Such result supports the consistent strong performance of the proposed SNV model across various evaluation criterion.

5.5 Conclusion

In this chapter, we proposed a new volatility model which combines the ARCH(1) model and the NNV model. We estimated the proposed model based on a Bayesian

sampling algorithm which selects the optimal bandwidth for the Nadaraya-Watson estimator.

The empirical performance of the proposed SNV model and Bayesian estimation method are evaluated and compared against the NNV models based on QLIKE and LSCV bandwidth selection methods. Based on financial return data of eight major global stock markets, both in-sample and out-of-sample performance are examined. Through the calculation of the loss functions given by Patton (2011), the Bayesian method shows strong performance results in in-sample period and even stronger result in out-of-sample period.

In addition, we examined the empirical VaR performance of the competing models. The proposed SNV model with standard Gaussian density showed the best performance in most cases. The empirical performance of the proposed SNV model is highly competitively comparing to the existing models.

As for the future research, more study may be carried out to evaluate the potentials of Bayesian bandwidth selection for the kernel Gaussian density in the SNV model, such as making the bandwidth adaptive on the local data points. Another area worth exploring is to build a SNV model in high-dimensional setting, which allows potential spillovers and correlations across different markets.

Chapter 6

Conclusion

This thesis has devoted to the investigation of several important topics in bandwidth selection for kernel density estimation based on sample data that may be of irregular distributions. The first issue evolves in bandwidth selection given the characteristics of the observed data in multivariate setting. In Chapter 3, we proposed a kernel density estimator with tail-adaptive bandwidths. We derived the posterior of bandwidth parameters based on the Kullback-Leibler information and presented an MCMC sampling algorithm to estimate bandwidths. The Monte Carlo simulation study shows that the kernel density estimator with tail-adaptive bandwidths estimated through the proposed sampling algorithm outperforms its competitor, the kernel density estimator with a global bandwidth estimated through either the normal reference rule discussed in Scott (1992) or the sampling algorithm proposed by Zhang et al. (2006).

The tail-adaptive kernel density estimator was applied to the estimation of bivariate density of the paired daily returns of the Australian Ordinary index and S&P500 index during the period of global financial crisis. The results showed that this estimator

could capture richer dynamics in the tail area than the density estimator with a global bandwidth estimated through the normal reference rule and a Bayesian sampling algorithm. With the tail-adaptive bandwidths estimated through our proposed Bayesian sampling algorithm, we have derived the estimated conditional density and distribution of the Australian index return given that the U.S. market finished daily trading with different return values. We have found that during the global financial crisis, even though the Australian stock market followed the U.S. stock market, there was no more than 27% chance that the former market had a larger drop than the latter.

The second issue was to investigate adapted bandwidth selection for multimodal distributions or data exhibits clustering behaviours. Chapter 4 proposed a cluster-adaptive bandwidth kernel density estimator for data with multimodality. This method employs a clustering algorithm to assign different bandwidths to different clusters identified in the data set. We have derived the posterior of bandwidth parameters based on Kullback-Leibler information and presented an MCMC sampling algorithm to sample these bandwidths. The Monte Carlo simulation study was designed to examine the performance of the proposed methods when the data exhibits multimodality. The results showed that when the underlying density is mixture of normal, the kernel density estimator with cluster-adaptive bandwidths estimated through our proposed Bayesian sampling algorithm outperforms its competitor. When the underlying densities are fat-tailed, the combined approach of tail- and cluster-adaptive density estimator performs the best.

In an empirical application, the study estimated bandwidth matrices for the cluster-adaptive kernel density estimator on the well known data set of eruption duration

and waiting time to the next eruption collected from Old Faithful greyer, which is famous of its clustering nature. The results again showed clear advantage of the proposed cluster-adaptive kernel density estimator over traditional approaches. As for the future research, by incorporating clustering information and assigning a different bandwidth matrix to each cluster, the performance of the kernel density estimator can be improved. In light of this result, it is intuitive to believe that improved density estimation performance would further facilitate improving the clustering accuracy. It should be noted that future research may be conducted to investigate and provide more insight on such issue.

The third topic extends the Bayesian bandwidth selection technique to volatility models for time series data. The study is motivated by the fact that very limited attention has been invested on the estimation of nonparametric and nonlinear volatility models through Bayesian approaches. Chapter 5 introduced a new volatility model, i.e. the SNV model, which combines the ARCH(1) model and the NNV model (Han & Zhang 2012) estimated via a proposed Bayesian sampling algorithm. This algorithm selects the optimal bandwidth for the Nadaraya-Watson estimator. Based on financial return data of eight major global stock markets, both in-sample and out-of-sample performance of the proposed model against competing models were examined. Through the calculation of the loss functions given by Patton (2011), the SNV model and the Bayesian estimation method showed strong and consistent results.

In addition, the study evaluated the empirical VaR performance of the competing models. The proposed SNV model showed the best performance in most cases. Overall, the empirical performance of the proposed Bayesian SNV model is highly competitive compared to the existing nonparametric nonlinear volatility models. More study may be carried out to evaluate the potentials of Bayesian bandwidth selection for the unknown density function in the SNV model, such as making the bandwidth adaptive on the local data points. Another area to explore is to build the SNV model in high-dimensional setting which could allow for potential spillovers and correlations across different markets.

Bibliography

Abramson, I. (1982*a*), 'Arbitrariness of the pilot estimator in adaptive kernel methods', *Journal of Multivariate Analysis* **12**(4), 562–567.

Abramson, I. (1982*b*), 'On bandwidth variation in kernel estimates — a square root law', *The Annals of Statistics* **10**(4), 1217–1223.

Aderberg, M. (1973), *Cluster Analysis for Applications*, Academic Press, New York.

Azzalini, A. & Bowman, A. (1990), 'A look at some data on the Old Faithful geyser', *Applied Statistics* **39**(3), 357–365.

Azzalini, A. & Capitanio, A. (2003), 'Distributions generated by perturbation of symmetry with emphasis on a multivariate skew t-distribution', *Journal of the Royal Statistical Society. Series B (Methodological)* **65**(2), 367–389.

Azzalini, A. & Torelli, N. (2007), 'Clustering via nonparametric density estimation', *Statistics and Computing* **17**(1), 71–80.

Azzalini, A. & Valle, A. (1996), 'The multivariate skew-normal distribution', *Biometrika* **83**(4), 715–726.

Badrinath, S. & Chatterjee, S. (1988), 'On measuring skewness and elongation in common stock return distributions: The case of the market index', *Journal of Business* **61**(4), 451–472.

Baillie, R., Bollerslev, T. & Mikkelsen, H. (1996), 'Fractionally integrated generalized autoregressive conditional heteroskedasticity', *Journal of Econometrics* **74**(1), 3–30.

Bao, Y., Lee, T. & Saltoglu, B. (2006), 'Evaluating predictive performance of value-at-risk models in emerging markets: a reality check', *Journal of Forecasting* **25**(2), 101–128.

Barndorff-Nielsen, O., Hansen, P., Lunde, A. & Shephard, N. (2008), 'Designing realized kernels to measure the ex post variation of equity prices in the presence of noise', *Econometrica* **76**(6), 1481–1536.

Berkhin, P. (2006), A survey of clustering data mining techniques, *in* J. Kogan, C. Nicholas & M. Teboulle, eds, 'Grouping Multidimensional Data', Springer, Berlin, pp. 25–71.

Blair, B. J., Poon, S.-H. & Taylor, S. J. (2001), 'Forecasting S&P 100 volatility: the incremental information content of implied volatilities and high-frequency index returns', *Journal of Econometrics* **105**(1), 5 – 26.

Bollerslev, T. (1986), 'Generalized autoregressive conditional heteroskedasticity', *Journal of Econometrics* **31**(3), 307–327.

Bollerslev, T., Chou, R. & Kroner, K. (1992), 'ARCH modeling in finance: A review of the theory and empirical evidence', *Journal of Econometrics* **52**(1-2), 5–59.

Bollerslev, T., Engle, R. & Nelson, D. (1994), ARCH models, *in* R. Engle & D. McFadden, eds, 'Handbook of Econometrics', Vol. 4, North Holland Press: Amsterdam, pp. 2959 – 3038.

Bollerslev, T. & Melvin, M. (1994), 'Bidask spreads and volatility in the foreign exchange market: An empirical analysis', *Journal of International Economics* **36**(3), 355–372.

Bollerslev, T. & Wooldridge, J. (1992), 'Quasi-maximum likelihood estimation and inference in dynamic models with time-varying covariances', *Econometric Reviews* **11**(2), 143–172.

Bowman, A. W. (1984), 'An alternative method of cross-validation for the smoothing of density estimates', *Biometrika* **71**, 353–360.

Breiman, L., Meisel, W. & Purcell, E. (1977), 'Variable kernel estimates of multivariate densities', *Technometrics* **19**(2), 135–144.

Brewer, M. (2000), 'A Bayesian model for local smoothing in kernel density estimation', *Statistics and Computing* **10**(4), 299–309.

Brooks, S. P. (1998), 'Markov chain Monte Carlo method and its application', *Journal of the Royal Statistical Society: Series D (The Statistician)* **47**, 69–100.

Bülmann, P. & McNeil, A. J. (2002), 'An algorithm for nonparametric GARCH modelling', *Computational Statistics & Data Analysis* **40**(4), 665 – 683.

Burman, P. & Polonik, W. (2009), 'Multivariate mode hunting: Data analytic tools with measures of significance', *Journal of Multivariate Analysis* **100**(6), 1198–1218.

Cuevas, A., Febrero, M. & Fraiman, R. (2000), 'Estimating the number of clusters', *Canadian Journal of Statistics* **28**(2), 367–382.

Cuevas, A., Febrero, M. & Fraiman, R. (2001), 'Cluster analysis: a further approach based on density estimation', *Computational Statistics & Data Analysis* **36**(4), 441–459.

de Lima, M. & Atuncar, G. (2010), 'A Bayesian method to estimate the optimal bandwidth for multivariate kernel estimator', *Journal of Nonparametric Statistics* **23**(1), 137–148.

Devroye, L. & Wise, G. (1980), 'Detection of abnormal behavior via nonparametric estimation of the support', *SIAM Journal on Applied Mathematics* **38**(3), 480–488.

Ding, Z., Granger, C. & Engle, R. (1993), 'A long memory property of stock market returns and a new model', *Journal of Empirical Finance* **1**(1), 83–106.

Drost, F. & Klaassen, C. (1997), 'Efficient estimation in semiparametric GARCH models', *Journal of Econometrics* **81**(1), 193–221.

Duin, R. (1976), 'On the choice of smoothing parameters for Parzen estimators of probability density functions', *IEEE Transactions on Computers* **100**(25), 1175–1179.

Duong, T. & Hazelton, M. (2003), 'Plug-in bandwidth matrices for bivariate kernel density estimation', *Journal of Nonparametric Statistics* **15**(1), 17–30.

Engle, R. (1982), 'Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation', *Econometrica* **50**(4), 987–1007.

Engle, R. & Bollerslev, T. (1986), 'Modelling the persistence of conditional variances', *Econometric Reviews* **5**(1), 1–50.

Engle, R. F., Ito, T. & Lin, W.-L. (1990), 'Meteor showers or heat waves? heteroskedastic intra-daily volatility in the foreign exchange market', *Econometrica* **58**(3), 525–542.

Engle, R. F. & Ng, V. (1993), 'Measuring and testing the impact of news on volatility', *The Journal of Finance* **48**(5), 1749–1778.

Engle, R. & Gonzalez-Rivera, G. (1991), 'Semiparametric arch models', *Journal of Business & Economic Statistics* **9**(4), 345–359.

Engle, R., Ng, V. & Rothschild, M. (1990), 'Asset pricing with a factor-arch covariance structure: empirical estimates for treasury bills', *Journal of Econometrics* **45**(1), 213–237.

Engle, R. & Patton, A. (2001), 'What good is a volatility model?', *Quantitative Finance* **1**(2), 237–245.

Ester, M., Kriegel, H., Sander, J. & Xu, X. (1996), A density-based algorithm for discovering clusters in large spatial databases with noise, *in* 'Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining (KDD-96)', Portland: AAAI Press, pp. 226–231.

Ferraty, F. & Vieu, P. (2006), *Nonparametric Functional Data Analysis: Theory and Practice*, Springer, New York.

Fleming, J., Ostdiek, B. & Whaley, R. E. (1995), 'Predicting stock market volatility: A new measure', *Journal of Futures Markets* **15**(3), 265–302.

Franke, J., Neumann, M. & Stockis, J. (2004), 'Bootstrapping nonparametric estimators of the volatility function', *Journal of Econometrics* **118**(1-2), 189–218.

French, K. R., Schwert, G. & Stambaugh, R. F. (1987), 'Expected stock returns and volatility', *Journal of Financial Economics* **19**(1), 3 – 29.

Gangopadhyay, A. & Cheung, K. (2002), 'Bayesian approach to the choice of smoothing parameter in kernel density estimation', *Journal of Nonparametric Statistics* **14**(6), 655–664.

Gelman, A., Carlin, J. B., Stern, H. S. & Rubin, D. B. (2004), *Bayesian Data Analysis*, 2nd edn, Chapman & Hall, New York.

Geyer, C. J. (1992), 'Practical Markov chain Monte Carlo', *Statistical Science* **7**, 473–483.

Gilks, W. R., Richardson, S. & Spiegelhalter, D. J. (1996), Introducing Markov chain Monte Carlo, *in* W. R. Gilks, S. Richardson & D. J. Spiegelhalter, eds, 'Markov Chain Monte Carlo in Practice', Chapman & Hall, London, pp. 1–19.

Glosten, L., Jagannathan, R. & Runkle, D. (1993), 'On the relation between the expected value and the volatility of the nominal excess return on stocks', *Journal of Finance* **48**, 1779–1801.

Gordon, A. (1999), *Classification*, 2nd edn, Chapman & Hall, London.

Hall, P. (1982), 'Cross-validation in density estimation', *Biometrika* **69**(2), 383–390.

Hall, P. (1987*a*), 'On Kullback-Leibler loss and density estimation', *The Annals of Statistics* **15**(4), 1491–1519.

Hall, P. (1987*b*), 'On the use of compactly supported density estimates in problems of discrimination', *Journal of Multivariate Analysis* **23**, 131–158.

Hall, P. & Marron, J. (1982), 'Local minima in cross-validation functions', *Journal of the American Statistical Asscoiation* **82**, 1131–1146.

Hall, P. & Yao, Q. (2003), 'Inference in arch and garch models with heavy–tailed errors', *Econometrica* **71**(1), 285–317.

Han, H. & Park, J. (2008), 'Time series properties of ARCH processes with persistent covariates', *Journal of Econometrics* **146**(2), 275–292.

Han, H. & Zhang, S. (2012), 'Non-stationary non-parametric volatility model', *The Econometrics Journal* **15**(2), 204–225.

Hansen, P. & Lunde, A. (2006), 'Consistent ranking of volatility models', *Journal of Econometrics* **131**(1), 97–121.

Härdle, W. (1990), *Applied Nonparametric Regression*, Cambridge University Press, London.

Härdle, W. (1991), *Smoothing Techniques: with Implementation in S*, Springer, New York.

Härdle, W. & Tsybakov, A. (1997), 'Local polynomial estimators of the volatility function in nonparametric autoregression', *Journal of Econometrics* **81**(1), 223–242.

Hartigan, J. A. (1975), *Clustering Algorithms*, John Wiley & Sons, New York.

Hartigan, J. A. (1981), 'Consistency of single linkage for high-density clusters', *Journal of the American Statistical Association* **76**(374), 388–394.

Hartigan, J. A. (1985), 'Statistical theory in clustering', *Journal of Classification* **2**, 63–76.

Hartigan, J. A. (1987), 'Estimation of a convex density contour in two dimensions', *Journal of the American Statistical Association* **82**(397), 267–270.

Hastings, W. K. (1970), 'Monte Carlo sampling methods using Markov chains and their applications', *Biometrika* **57**(1), 97–109.

Heber, G., Lunde, A., Shephard, N. & Sheppard, K. (2009), Oxford-man institute's realized library. Version 0.2, Oxford-Man Institute, University of Oxford.

Holmes, M. P., Gray, A. G. & Isbell Jr, C. L. (2010), 'Fast kernel conditional density estimation: A dual-tree Monte Carlo approach', *Computational Statistics & Data Analysis* **54**(7), 1707–1718.

Horová, I., Vieu, P. & Zelinka, H. (2002), 'Optimal choice of nonparametric estimaties of a density and of its derivatives', *Statistics & Decisions* **20**(4), 355–378.

Hyndman, R. J. (1996), 'Computing and graphing highest density regions', *The American Statistician* **50**(2), 120–126.

Izenman, A. J. (1991), 'Recent developments in nonparametric density estimation', *Journal of the American Statistical Association* **86**(413), 205–224.

Jain, A. (2010), 'Data clustering: 50 years beyond K-means', *Pattern Recognition Letters* **31**(8), 651–666.

Jones, M. C. (1990), 'Variable kernel density estimates and variable kernel density estimates', *Australian & New Zealand Journal of Statistics* **32**(3), 361–371.

Jones, M. C., Marron, J. S. & Sheather, S. J. (1996), 'A brief survey of bandwidth selection for density estimation', *Journal of the American Statistical Association* **91**(433), 401–407.

Kanas, A. (2012), 'Modelling the riskreturn relation for the S&P 100: The role of VIX', *Economic Modelling* **29**(3), 795 – 809.

Kim, S., Shepherd, N. & Chib, S. (1998), 'Stochastic volatility: Likelihood inference and comparison with ARCH models', *Review of Economic Studies* **65**(3), 361–393.

Klemelä, J. (2004), 'Visualization of multivariate density estimates with level set trees', *Journal of Computational and Graphical Statistics* **13**(3), 599–620.

Klemelä, J. (2006), 'Visualization of multivariate density estimates with shape trees', *Journal of Computational and Graphical Statistics* **15**(2), 372–397.

Koenker, R. & Bassett Jr, G. (1978), 'Regression quantiles', *Econometrica* **46**(1), 33–50.

Kulasekera, K. B. & Padgett, W. J. (2006), 'Bayes bandwidth selection in kernel density estimation with censored data', *Journal of Nonparametric Statistics* **18**(2), 129–143.

Lamoureux, C. G. & Lastrapes, W. D. (1990), 'Heteroskedasticity in stock return data: Volume versus garch effects', *The Journal of Finance* **45**(1), 221–229.

Linton, O. (2009), Semiparametric and Nonparametric ARCH Modeling, *in* T. Andersen, R. Davis, J. Kreiss & T. Mikosch, eds, 'Handbook of Financial Time Series', Springer: Berlin, pp. 157–167.

Linton, O. & Mammen, E. (2005), 'Estimating semiparametric ARCH(∞) models by kernel smoothing methods', *Econometrica* **73**, 771–836.

Loftsgaarden, D. O. & Quesenberry, C. P. (1965), 'A nonparametric estimate of a multivariate density function', *The Annals of Mathematical Statistics* **36**(3), 1049–1051.

MacQueen, J. (1967), Some methods for classification and analysis of multivariate observations, *in* 'Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability', Vol. 1, p. 14.

Marron, J. S. & Nolan, D. (1988), 'Canonical kernels for density estimation', *Statistics & Probability Letters* **7**(3), 195 – 199.

Mason, D. M. & Polonik, W. (2009), 'Asymptotic normality of plug-in level set estimates', *The Annals of Applied Probability* **19**(3), 1108–1142.

Masry, E. & Tjøstheim, D. (1995), 'Nonparametric estimation and identification of nonlinear ARCH time series: strong convergence and asymptotic normality', *Econometric Theory* **11**, 258–258.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E. et al. (1953), 'Equation of state calculations by fast computing machines', *The Journal of Chemical Physics* **21**(6), 1087.

Mielniczuk, J., Sarda, P. & Vieu, P. (1989), 'Local data-driven bandwidth choice for density estimation', *Journal of Statistical Planning and Inference* **23**(1), 53 – 69.

Nelson, D. (1991), 'Conditional heteroskedasticity in asset returns: a new approach', *Econometrica* **59**(2), 347–370.

Newton, M. A. & Raftery, A. E. (1994), 'Approximate Bayesian inference with the weighted likelihood bootstrap', *Journal of the Royal Statistical Society. Series B (Methodological)* **56**(1), 3–48.

Nolan, D. & Marron, J. (1989), 'Uniform consistency of automatic and location-adaptive delta-sequence estimators', *Probability Theory and Related Fields* **80**(4), 619–632.

Nolan, O. & Polland, D. (1987), 'U-processes: rates of convergence', *Annals of Statistics* **15**, 780–799.

Pagan, A. R. & Schwert, G. W. (1990), 'Alternative models for conditional stock volat', *Journal of Econometrics* **45**, 267–290.

Pagan, A. & Ullah, A. (1999), *Nonparametric Econometrics*, Cambridge University Press, Cambridge.

Park, B. U. & Marron, J. S. (1990), 'Comparison of data-driven bandwidth selectors', *Journal of the American Statistical Association* **85**(409), 66–72.

Park, J. (2002), 'Nonstationary nonlinear heteroskedasticity', *Journal of econometrics* **110**(2), 383–415.

Park, J. & Phillips, P. (1999), 'Asymptotics for nonlinear transformations of integrated time series', *Econometric Theory* **15**(3), 269–298.

Park, J. & Phillips, P. (2001), 'Nonlinear regressions with integrated time series', *Econometrica* **69**(1), 117–161.

Patton, A. (2011), 'Volatility forecast comparison using imperfect volatility proxies', *Journal of Econometrics* **160**(1), 246–256.

Polak, J., Zhang, X. & King, M. L. (2010), Bandwidth selection for kernel conditional density estimation using the MCMC method. Manuscript presented at Australian Statistical Conference, 6-10 December, Fremantle, Western Australia.

Robert, C. P. & Casella, G. (1999), *Monte Carlo Statistical Methods*, Vol. 58, Springer, New York.

Roberts, G. O. (1996), Markov chain concepts related to sampling algorithms, *in* W. R. Gilks, S. Richardson & D. J. Spiegelhalter, eds, 'Markov Chain Monte Carlo in Practice', Chapman & Hall, London, pp. 45–57.

Rudemo, M. (1982), 'Empirical choice of histograms and kernel density estimators', *Scandinavian Journal of Statistics* **9**(2), 65–78.

Sain, S. R. (2002), 'Multivariate locally adaptive density estimation', *Computational Statistics & Data Analysis* **39**(2), 165–186.

Sain, S. R., Baggerly, K. A. & Scott, D. W. (1994), 'Cross-validation of multivariate densities.', *Journal of the American Statistical Association* **89**(427), 807–817.

Sain, S. R. & Scott, D. W. (1996), 'On locally adaptive density estimation', *Journal of the American Statistical Association* **91**(436), 1525–1534.

Samworth, R. J. & Wand, M. P. (2010), 'Asymptotics and optimal bandwidth selection for highest density region estimation', *The Annals of Statistics* **38**(3), 1767–1792.

Scott, D. W. (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization*, John Wiley & Sons, New York.

Scott, D. W. & Terrell, G. R. (1987), 'Biased and unbiased cross-validation in density estimation', *Journal of the American Statistical Asscoiation* **82**, 1131–1146.

Silverman, B. W. (1986), *Density estimation for statistics and data analysis*, Chapman & Hall, New York.

Simonoff, J. S. (1996), *Smoothing Methods in Statistics*, Springer, New York.

Stuetzle, W. (2003), 'Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample', *Journal of Classification* **20**(1), 25–47.

Sugar, C. & James, G. (2003), 'Finding the number of clusters in a dataset: An Information-Theoretic Approach', *Journal of the American Statistical Association* **98**(463), 750–763.

Terrell, G. R. & Scott, D. W. (1992), 'Variable kernel density estimation', *The Annals of Statistics* **20**(3), 1236–1265.

Tibshirani, R., Walther, G. & Hastie, T. (2001), 'Estimating the number of clusters in a data set via the gap statistic', *Journal of the Royal Statistical Society: Series B (Methodological)* **63**(2), 411–423.

Tierney, L. (1994), 'Markov chains for exploring posterior distributions', *The Annals of Statistics* **22**, 1701–1762.

Tsay, R. S. (2005), *Analysis of Financial Time Series*, Wiley, New Jersey.

Vieu, P. (1999), 'Multiple kernel procedure: An asymptotic support', *Scandinavian Journal of Statistics* **26**(1), 61–72.

Walther, G. (1997), 'Granulometric smoothing', *The Annals of Statistics* **25**(6), 2273–2299.

Wand, M. P. & Jones, M. C. (1994), 'Multivariate plug-in bandwidth selection', *Computational Statistics* **9**(2), 97–116.

Wand, M. P. & Jones, M. C. (1995), *Kernel Smoothing*, Chapman & Hall, New York.

Wang, Q. & Phillips, P. (2009*a*), 'Asymptotic theory for local time density estimation and nonparametric cointegrating regression', *Econometric Theory* **25**, 710–738.

Wang, Q. & Phillips, P. (2009*b*), 'Structural nonparametric cointegrating regression', *Econometrica* **77**(6), 1901–1948.

Wong, M. A. & Lane, T. (1983), 'A kth nearest neighbour clustering procedure', *Journal of the Royal Statistical Society. Series B (Methodological)* **45**(3), 362–368.

Xu, R., Wunsch, D. et al. (2005), 'Survey of clustering algorithms', *Neural Networks, IEEE Transactions on Neural Networks* **16**(3), 645–678.

Yang, L. (2006), 'A semiparametric GARCH model for foreign exchange volatility', *Journal of Econometrics* **130**(2), 365–384.

Yang, L., Hardle, W. & Nielsen, J. (1999), 'Nonparametric autoregression with multiplicative volatility and additive mean', *Journal of Time Series Analysis* **20**(5), 579–604.

Zhang, X., Brooks, R. D. & King, M. L. (2009), 'A Bayesian approach to bandwidth selection for multivariate kernel regression with an application to state-price density estimation', *Journal of Econometrics* **153**(1), 21–32.

Zhang, X. & King, M. L. (2011), Bayesian semiparametric GARCH models. Manuscript presented at Bayes on the Beach, 4-5 October, 2010, Surfers Paradise, Australia.

Zhang, X. & King, M. L. (2013), Gaussian Kernel GARCH Models. Working Paper, Monash University.

Zhang, X., King, M. L. & Hyndman, R. J. (2006), 'A Bayesian approach to bandwidth selection for multivariate kernel density estimation', *Computational Statistics & Data Analysis* **50**(11), 3009–3031.