# A Bayesian approach to parameter estimation for kernel density estimation via transformations

Qing Liu*,[1], David Pitt[2], Xibin Zhang[3], Xueyuan Wu[1]

[1]Centre for Actuarial Studies, Faculty of Business and Economics, The University of Melbourne

[2]Department of Applied Finance and Actuarial Studies, Faculty of Business and Economics, Macquarie University

[3]Department of Econometrics and Business Statistics, Monash University

September 2010

This version: January 2011

**Abstract:** In this paper, we present a Markov chain Monte Carlo (MCMC) simulation algorithm for estimating parameters in the kernel density estimation of bivariate insurance claim data via transformations. Our data set consists of two types of auto insurance claim costs and exhibits a high-level of skewness in the marginal empirical distributions. Therefore, the kernel density estimator based on original data does not perform well. However, the density of the original data can be estimated through estimating the density of the transformed data using kernels. It is well known that the performance of a kernel density estimator is mainly determined by the bandwidth, and only in a minor way by the kernel. In the current literature, there have been some developments in the area of estimating densities based on transformed data, where bandwidth selection usually depends on pre-determined transformation parameters. Moreover, in the bivariate situation, the transformation parameters were estimated for each dimension individually. We extend the Bayesian sampling algorithm proposed by Zhang, King and Hyndman (2006) and present a Metropolis-Hastings sampling procedure to sample the bandwidth and transformation parameters from their posterior density. Our contribution is to estimate the bandwidths and transformation parameters simultaneously within a Metropolis-Hastings sampling procedure. Moreover, we demonstrate that the correlation between the two dimensions is better captured through the bivariate density estimator based on transformed data.

**Key words**: bandwidth parameter; kernel density estimator; Markov chain Monte Carlo; Metropolis-Hastings algorithm; power transformation; transformation parameter.

*Corresponding author. Centre for Actuarial Studies, Faculty of Business and Economics, The University of Melbourne, VIC 3010, Australia. Email: q.liu5@pgrad.unimelb.edu.au.

## 1 Introduction

Kernel density estimation is one of the widely used non-parametric estimation techniques for estimating the probability density function of a random variable. For a univariate random variable $X$ with unknown density $f(x)$, if we draw a sample of $n$ independent and identically distributed observations $x_1, x_2, \ldots, x_n$, the kernel density estimator is given by (Wand and Jones, 1995)

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K\left(\frac{x-x_i}{h}\right),$$

where $h$ is the bandwidth that controls the amount of smoothness, and $K(\cdot)$ is the kernel function which is usually chosen to be a symmetric density function. Wand, Marron and Ruppert (1991) argued that the classical kernel density estimator does not perform well when the underlying density is asymmetric because such an estimation requires different amounts of smoothing at different locations. Therefore, they proposed to transform the data with the intention that the use of a global bandwidth is appropriate for the kernel density estimator after transformation. The power transformation is one such transformation for this purpose.

There are a number of alternative transformation methods that have been studied in the literature. For example, Hjort and Gald (1995) advocated a semi-parametric estimator with a parametric start. Clements, Hurn and Lindsay (2003) introduced the Mobius-like transformation. Buch-Larsen, Nielsen, Guillén and Bolancé (2005) proposed an estimator obtained by transforming the data with a modification of the Champernowne cumulative density function and then estimating the density of the transformed data through the kernel density estimator. These transformation methods are particularly useful with insurance data because the distributions of insurance claim data are often skewed and present heavy-tailed features. However, these transformations often involve some parameters, which have to be determined before the kernel density estimation is conducted. In this paper, we aim to present a sampling algorithm to estimate the bandwidth and transformation parameters simultaneously.

It is well established in the literature that the performance of a kernel density estimator is largely determined by the choice of bandwidth and only in a minor way, by kernel choice (see for example, Izenman, 1991; Scott, 1992; Simonoff, 1996). Many data-driven methods for bandwidth selection have been proposed and studied in the literature (see for example, Marron, 1988; Sheather and Jones, 1991; Scott, 1992; Bowman and Azzalini; 1997). However, Zhang, King and Hyndman (2006) indicated that kernel density estimation for multivariate data has received significantly less attention than its univariate counterpart due to the increased difficulty in deriving an optimal data-driven bandwidth as the dimension of the data increases. They proposed MCMC algorithms to estimate bandwidth parameters for multivariate kernel density estimation.

The data set we use in this paper has two dimensions, and therefore we could use the sampling algorithm presented by Zhang et al. (2006) to estimate bandwidth parameters. However, their algorithm has so far only been used to estimate a density for directly observed data. As our data are highly positively skewed and have to be transformed for the purpose of density estimation, we extend their MCMC algorithm so that it estimates not only the bandwidth parameters but also the transformation parameters for the bivariate insurance claim data. Bolancé, Guillén, Pelican and Vernic (2008) analysed this data set using the kernel density estimation via transformations. This approach captures a certain degree of correlation between the two dimensions by using the product kernel. However, the parameters involved in the transformed kernel density estimator were estimated by dealing with each dimension individually, and this is likely to underestimate the correlation between the two dimensions. In this paper, we present MCMC algorithms for estimating the bandwidth and transformation parameters for not only univariate data but also bivariate data. We investigate the differences in estimated correlations calculated through both sampling algorithms.

The rest of the paper is organised as follows. In Section 2, we provide a brief summary of the data and demonstrate the motivation for the paper. Section 3 presents MCMC algorithms for estimating bandwidth parameters and transformation parameters for kernel density estimation via transformations for univariate and bivariate data. In Section 4, we examine the performance of our MCMC algorithms in choosing bandwidths and estimating transformation parameters for the bivariate insurance claim data. Section 5 concludes the paper.

## 2 Data and motivation

Our data set is the one analysed by Bolancé et al. (2008), whose data were collected from a major automobile insurance company in Spain. The data contain 518 paired claims. Each claim contains two types of losses, which are respectively, property damage $X_1$ and medical expenses $X_2$. It is intuitive that a serious car accident might cause serious damage to the cars, and the passengers involved in the accident might also be seriously injured. Therefore, we expect that the two types of claims are positively correlated.

Figure 1 presents a scatter plot of claims of bodily injury costs against property damage costs, as well as a scatter plot of the logarithms of such claim costs. The two graphs suggest that there exists an obvious positive correlation between the two types of costs.

Bolancé et al. (2008) investigated modelling the data using both the classical kernel density estimation method and the transformed kernel density estimation method. They found that the transformed kernel estimation approach obviously performs better than the classical kernel estimation method in terms of calculating the conditional tail expectation (CTE). They firstly estimated the transformation parameters by looking at each dimension of the bivariate costs, and then used the product kernel for the kernel density estimator for the bivariate transformed data. The use of the product kernel can capture a certain degree of correlation between the two dimensions. We wish to see whether there is an improvement in the correlation captured if we take both dimensions into account during the parameter estimation process. In this paper, we propose to estimate the bandwidths and transformation parameters for the bivariate data through our new Bayesian sampling algorithm.

# 3 Bayesian sampling algorithms

## 3.1 Kernel density estimation for transformed data

The kernel density estimation technique is often of great interest in estimating the density for a set of data. However, when the underlying true density has heavy tails, the kernel density estimator (with a global bandwidth being used) can perform quite poorly. Wand et al. (1991) suggested transforming the data and obtaining the kernel density estimator for the transformed data. The density estimator for the untransformed data is the derived kernel density estimator for the transformed data multiplied by the Jacobian of such a transformation. Wand et al. (1991) found that compared to working with kernel density estimation for untransformed data, significant gains can be achieved by working with density estimation for transformed data.

The shifted power transformation is one such transformation that is effective in changing the degree of positive skewness in data (see for example, Wand et al., 1991). Such a transformation is given by

$$\tilde{y} = \tilde{T}_{\lambda_1,\lambda_2}(x) = \begin{cases} (x+\lambda_1)^{\lambda_2}\text{sign}(\lambda_2) & \text{if } \lambda_2 \neq 0 \\ \ln(x+\lambda_1) & \text{if } \lambda_2 = 0 \end{cases},$$

where $\lambda_1 > -\min\{x_1, x_2, \cdots, x_n\}$, and $\lambda_2 < 1$. To ensure that this transformation is scale preserving, $\tilde{y}$ is further transformed as

$$y = T_{\lambda_1,\lambda_2}(x) = \left(\frac{\sigma_x}{\sigma_{\tilde{y}}}\right)\tilde{y},$$

where $\sigma_x^2$ and $\sigma_{\tilde{y}}^2$ are the variances of $x$ and $\tilde{y}$, respectively. Let $y_i = T_{\lambda_1,\lambda_2}(x_i)$, for $i = 1, 2, \cdots, n$. The kernel density estimator for the univariate transformed data is

$$\tilde{f}_{h,\lambda_1,\lambda_2}(y) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K\left(\frac{y - y_i}{h}\right),$$

and the kernel density estimator for the untransformed data is

$$\hat{f}_{h,\lambda_1,\lambda_2}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K\left(\frac{T_{\lambda_1,\lambda_2}(x) - T_{\lambda_1,\lambda_2}(x_i)}{h}\right)T'_{\lambda_1,\lambda_2}(x).$$

Wand et al. (1991) investigated data-driven selection methods for the choice of transformation parameters and bandwidth or smoothing parameter for univariate data. However, the transformation parameters have to be pre-determined for chosen bandwidths. Moreover, when the dimension of data increases, the estimation of these parameters becomes increasingly difficult. In this paper, we aim to estimate the transformation parameters and bandwidth parameters simultaneously.

### 3.2 Bivariate kernel density estimation via transformation

Let $\boldsymbol{X} = (X_1, X_2)^\top$ denote a bivariate random vector with density $f(\boldsymbol{x})$, and let $\boldsymbol{x}_i = (x_{i1}, x_{i2})^\top$, for $i = 1, 2, \cdots, n$, be an independent random sample drawn from $f(\boldsymbol{x})$. The transformed data are denoted as $\boldsymbol{y}_i = (y_{i1}, y_{i2})^\top = (T_{\lambda_{11},\lambda_{21}}(x_{i1}), T_{\lambda_{12},\lambda_{22}}(x_{i2}))^\top$, for $i = 1, 2, \cdots, n$. The kernel density estimator for the bivariate transformed data is given by

$$\hat{f}(\boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_1 h_2}\mathcal{K}\left(\frac{y_1 - y_{i1}}{h_1}, \frac{y_2 - y_{i2}}{h_2}\right), \tag{1}$$

where $h_1$ and $h_2$ are bandwidths for the two dimensions, and $\mathcal{K}(\cdot,\cdot)$ is a bivariate kernel function which is usually the product of two univariate kernels. Therefore, this bivariate kernel estimator can be re-written as

$$\hat{f}(\boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_1}K\left(\frac{y_1 - y_{i1}}{h_1}\right)\frac{1}{h_2}K\left(\frac{y_2 - y_{i2}}{h_2}\right). \tag{2}$$

The bivariate kernel density estimator for the original data is

$$\hat{f}_{\boldsymbol{h},\boldsymbol{\lambda_1},\boldsymbol{\lambda_2}}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}\left\{\prod_{k=1}^{2}\frac{1}{h_k}K\left(\frac{T_{\lambda_{1k},\lambda_{2k}}(x_k) - T_{\lambda_{1k},\lambda_{2k}}(x_{ik})}{h_k}\right)T'_{\lambda_{1k},\lambda_{2k}}(x_k)\right\}, \tag{3}$$

where $\boldsymbol{x} = (x_1, x_2)^\top$, $\boldsymbol{h} = (h_1, h_2)^\top$ is a vector of bandwidths, $\boldsymbol{\lambda_1} = (\lambda_{11}, \lambda_{21})^\top$ is a vector of transformation parameters for $x_1$, and $\boldsymbol{\lambda_2} = (\lambda_{12}, \lambda_{22})^\top$ is a vector of transformation parameters for $x_2$.

In the current literature, there are two limitations in using the kernel density estimation via transformations. First, the transformation parameters have to be pre-determined so that bandwidth parameters can be chosen through some currently available method. Second, when estimating the density of the insurance claim data, Bolancé et al. (2008) obtained the marginal kernel density estimator for each dimension via transformations. They derived the CTE through the estimated marginal densities. Their approach does capture a certain degree of correlation between the two dimensions by using the product kernel, while their parameter estimations was conducted for each dimension individually. In this paper, we aim to derive the posterior density of the transformation parameters and bandwidths and present a Metropolis-Hastings sampling algorithm to sample both types of parameters from their posterior density.

### 3.3 Bayesian sampling algorithms

Zhang et al. (2006) presented an MCMC sampling algorithm for estimating bandwidth parameters for kernel density estimation based on untransformed data. Treating bandwidths as parameters, they derived the posterior density of the bandwidths through the likelihood cross-validation criterion. This criterion involves choosing an optimal bandwidth that minimises the Kullback-Leibler information, which is a measure of the discrepancy between the true underlying density $f(y)$ and the density estimator $\hat{f}_h(y)$ and is defined as

$$d_{KL}(f, \hat{f}_h) = \int_R \log \left\{ \frac{f(y)}{\hat{f}_h(y)} \right\} f(y) dy.$$

Zhang et al. (2006) showed that minising Kullback-Leibler information is approximately equivalent to maximising

$$\hat{E} \ \log \left\{ \hat{f}_h(y) \right\} = \frac{1}{n} \sum_{i=1}^{n} \log \ \hat{f}_h(y_i) = \frac{1}{n} \sum_{i=1}^{n} \log \left\{ \frac{1}{n} \sum_{j=1}^{n} \frac{1}{h} K \left( \frac{y_i - y_j}{h} \right) \right\}, \tag{4}$$

with respect to $h$. However, if we directly maximise (4) with respect to $h$, the resulting bandwidth would be zero. One way of dealing with this problem is to estimate $f_h(y_i)$ based on the observations without $y_i$, and to approximate $\hat{E}$ $\log\{\hat{f}_h(y)\}$ by (Härdle, 1991)

$$L(y_1, y_2, \cdots, y_n | h) = \frac{1}{n} \sum_{i=1}^{n} \log \ \hat{f}_{(i),h}(y_i), \tag{5}$$

where $\hat{f}_{h,i}(y_i)$ is the leave-one-out estimator given by

$$\hat{f}_{(i),h}(y_i) = \frac{1}{n-1} \sum_{j=1;j\neq i}^{n} \frac{1}{h} K\left(\frac{y_i - y_j}{h}\right).$$

The log-likelihood of $\{y_1, y_2, \cdots, y_n\}$ for given $h$ could be approximated by $nL(y_1, y_2, \cdots, y_n|h)$. Therefore, the posterior density of $h$ is proportional to the product of the prior density of $h$ and this likelihood function.

The Bayesian sampling algorithm proposed by Zhang et al. (2006) is mainly used for estimating bandwidths in kernel density estimation for untransformed data. As our data are highly positively skewed, the original data should be transformed for the purpose of density estimation. We extend the sampling algorithm of Zhang et al. (2006) by deriving the posterior density of the bandwidth parameters and transformation parameters. Thus, we can estimate not only the bandwidth parameters but also the transformation parameters simultaneously for our kernel density estimator of the transformed data. When data are transformed through some transformation parameters, the kernel-form density estimator of the original data is given by (3), which is a function of bandwidth parameters and transformation parameters. We find that the sampling algorithm of Zhang et al. (2006) can be extended by including additional transformation parameters to sample both types of parameters from their posterior density constructed through (3).

### 3.3.1 Univariate kernel density estimation

We now investigate the issue of using Bayesian sampling techniques to estimate the transformation parameters, $\lambda_{1k}$ and $\lambda_{2k}$, and the bandwidth, $h_k$, based on univariate data, $(x_{1k}, \cdots, x_{nk})^\top$, for $k = 1$ and 2, respectively. As the parameters are estimated for each of the two dimensions respectively, any possible correlation between the two dimensions can only be captured through the use of the product kernel. For each dimension, we have three unknown parameters, namely $h_k$ (the bandwidth), $\lambda_{1k}$ and $\lambda_{2k}$ (the transformation parameters for shifted power transformation family). The posterior density of these three parameters can be obtained through the likelihood cross-validation criterion in the same way as in Zhang et al. (2006). We assume the prior density of $H_k$, whose realised value is $h_k$, is a normal density given by

$$p_0(h_k) = \frac{1}{\sqrt{2\pi\sigma_{h_k}^2}} \exp\left\{-\frac{(h_k - \mu_{h_k})^2}{2\sigma_{h_k}^2}\right\},$$

which is truncated at 0 so as to maintain the domain of positive bandwidths, for $k = 1$ and 2. The prior density of $\Lambda_{1k}$ is assumed to be the normal density given by

$$p_1(\lambda_{1k}) = \frac{1}{\sqrt{2\pi\sigma^2_{\lambda_{1k}}}} \exp\left\{-\frac{(\lambda_{1k} - \mu_{\lambda_{1k}})^2}{2\sigma^2_{\lambda_{1k}}}\right\},$$

which is left truncated at -min$\{x_{1k}, x_{2k}, \ldots, x_{nk}\}$, for $k = 1$ and 2. The prior density of $\Lambda_{2k}$ is assumed to be the uniform density on $(-a_k, 1)$ given by

$$p_2(\lambda_{2k}) = \frac{1}{1 + a_k},$$

for $k = 1$ and 2. Therefore, the joint prior density of $(H_k, \Lambda_{1k}, \Lambda_{2k})$ is

$$p(h_k, \lambda_{1k}, \lambda_{2k}) = p_0(h_k) \times p_1(\lambda_{1k}) \times p_2(\lambda_{2k}),$$

where the hyperparameters are $\mu_{h_k}$, $\sigma_{h_k}$, $\mu_{\lambda_{1k}}$, $\sigma_{\lambda_{1k}}$ and $a_k$, for $k = 1$ and 2. The likelihood is approximated as

$$\ell_k(x_{1k}, x_{2k}, \ldots, x_{nk} | h_k, \lambda_{1k}, \lambda_{2k}) = \prod_{i=1}^{n} \hat{f}_{(i),h_k,\lambda_{1k},\lambda_{2k}}(x_{ik}),$$

where $\hat{f}_{(i),h_k,\lambda_{1k},\lambda_{2k}}(x_{ik})$ denotes the leave-one-out estimator of the density of $x_{ik}$ (see for example, Zhang et al., 2006) given by

$$\hat{f}_{(i),h_k,\lambda_{1k},\lambda_{2k}}(x_{ik}) = \frac{1}{n-1} \sum_{j=1;j\neq i}^{n} \frac{1}{h_k} K\left(\frac{T_{\lambda_{1k},\lambda_{2k}}(x_{ik}) - T_{\lambda_{1k},\lambda_{2k}}(x_{jk})}{h_k}\right) T'_{\lambda_{1k},\lambda_{2k}}(x_{ik}),$$

for $k = 1$ and 2.

According to Bayes theorem, the posterior density of ($H_k$, $\Lambda_{2k}$, $\Lambda_{2k}$) is (up to a normalising constant)

$$\pi(h_k, \lambda_{1k}, \lambda_{2k} | x_{1k}, x_{2k}, \cdots, x_{nk}) \propto p(h_k, \lambda_{1k}, \lambda_{2k}) \times \ell_k(x_{1k}, x_{2k}, \ldots, x_{nk} | h_k, \lambda_{1k}, \lambda_{2k}), \tag{6}$$

for $k = 1$ and 2. Using the random-walk Metropolis-Hastings algorithm, we are able to sample $(h_1, \lambda_{11}, \lambda_{21})$ and $(h_2, \lambda_{12}, \lambda_{22})$ from (6) with $k = 1$ and 2, respectively. The ergodic average (or the posterior mean) of each parameter acts as an estimate of that parameter. In terms of univariate kernel density estimation discussed here, our contribution is to present a sampling algorithm that aims to estimate the bandwidth and transformation parameters within a Bayesian sampling procedure for univariate data. Hereafter, we call this sampling algorithm the univariate sampling algorithm.

### 3.3.2 Bivariate kernel density estimation

Given bivariate observations denoted as $\boldsymbol{x}_i$, for $i = 1, 2, \cdots, n$, and the parameter vector denoted as $(\boldsymbol{h}, \boldsymbol{\lambda_1}, \boldsymbol{\lambda_2})$, which were defined immediately after (3), the likelihood function is approximated as (Härdle, 1991)

$$\ell(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n | \boldsymbol{h}, \boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}) = \prod_{i=1}^{n} \hat{f}_{(i),\boldsymbol{h},\boldsymbol{\lambda_1},\boldsymbol{\lambda_2}}(\boldsymbol{x}_i), \tag{7}$$

where

$$\hat{f}_{(i),\boldsymbol{h},\boldsymbol{\lambda_1},\boldsymbol{\lambda_2}}(\boldsymbol{x}_i) = \frac{1}{n-1} \sum_{j=1;j\neq i}^{n} \left\{ \prod_{k=1}^{2} \frac{1}{h_k} K\left( \frac{T_{\lambda_{1k},\lambda_{2k}}(x_{ik}) - T_{\lambda_{1k},\lambda_{2k}}(x_{jk})}{h_k} \right) T'_{\lambda_{1k},\lambda_{2k}}(x_{ik}) \right\}, \tag{8}$$

the leave-one-out estimator of the density of $\boldsymbol{X}$ computed at $\boldsymbol{x}_i$, for $i = 1, 2, \cdots, n$.

Let the joint prior density of $(\boldsymbol{H}, \boldsymbol{\Lambda_1}, \boldsymbol{\Lambda_2})$ be denoted as $p(\boldsymbol{h}, \boldsymbol{\lambda_1}, \boldsymbol{\lambda_2})$, which is the product of marginal priors defined in Section 3.3.1. Then the posterior density of $(\boldsymbol{H}, \boldsymbol{\Lambda_1}, \boldsymbol{\Lambda_2})$ is (up to a normalising constant)

$$\pi(\boldsymbol{h}, \boldsymbol{\lambda_1}, \boldsymbol{\lambda_2} | \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n) \propto p(\boldsymbol{h}, \boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}) \times \ell(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n | \boldsymbol{h}, \boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}), \tag{9}$$

from which we can sample $(\boldsymbol{h}, \boldsymbol{\lambda_1}, \boldsymbol{\lambda_2})$ through an appropriate Bayesian sampling algorithm such as the Metropolis-Hastings sampling algorithm described as follows.

i) Conditional on $(\boldsymbol{\lambda_1}, \boldsymbol{\lambda_2})$, we sample $\boldsymbol{h}$ from (9) using the Metropolis-Hastings algorithm.

ii) Conditional on $\boldsymbol{h}$, we sample $(\boldsymbol{\lambda_1}, \boldsymbol{\lambda_2})$ from (9) using the Metropolis-Hastings algorithm.

The sampling algorithm in the first step is the same as the one presented by Zhang et al. (2006) for directly observed data. Alternatively, we can sample $(\boldsymbol{h}, \boldsymbol{\lambda_1}, \boldsymbol{\lambda_2})$ directly from its posterior density given by (9) using the Metropolis-Hastings algorithm. Hereafter, we call this sampling algorithm the bivariate sampling algorithm.

## 3.4 An application to bivariate insurance claim data

In order to explore the benefits that could be gained by estimating the parameters using bivariate data instead of separately estimating density for each dimension of data, we apply the MCMC algorithms proposed in Section 3.3 in two ways and compare the two sets of results.

First, we estimated $(h_k, \lambda_{1k}, \lambda_{2k})$ for the kernel density estimator of each variable based on univariate data $\{x_{1k}, x_{2k}, \ldots, x_{nk}\}$, for $k = 1$ and 2, using the sampling algorithm presented in Section 3.3.1. The hyperparameters were chosen to be $\mu_{h_k} = 40$ and $\sigma_{h_k} = 5$, for $k = 1$ and 2, and $\mu_{\lambda_{11}} = 1500$, $\sigma_{\lambda_{11}} = 333$, $\mu_{\lambda_{12}} = 90$, $\sigma_{\lambda_{12}} = 30$ and $a_k = 6$, for $k = 1$ and 2. In terms of the normal prior densities, the standard deviation values were deliberately chosen as large values,such that the normal prior densities are very flat. As we did not know the central locations of these normal prior densities, we tried a few values for these central locations by initially running the sampling algorithm several times. In terms of the uniform prior, we actually put a constraint through a reference to Bolancé et al. (2008). No matter what values were chosen for these hyperparameters, a resulting sampler should produce the best mixing performance.

Second, we estimated the bandwidth vector $\mathbf{h} = (h_1, h_2)^\top$, the transformation parameter vectors $\boldsymbol{\lambda_1}$ and $\boldsymbol{\lambda_2}$ in the bivariate density estimator for the bivariate data using the sampling algorithm presented in Section 3.3.2. The hyperparameters were chosen to be $\mu_{h_k} = 40$, $\sigma_{h_k} = 5$ for $k = 1$ and 2, $\mu_{\lambda_{11}} = 2300$, $\sigma_{\lambda_{11}} = 1000$, $\mu_{\lambda_{12}} = 40$, $\sigma_{\lambda_{12}} = 20$, $a_1 = 5$ and $a_2 = 2$. We actually followed the same rule as the above-mentioned in choosing values for these hyperparameters.

We are particularly interested in the correlation coefficient captured through both sampling algorithms. We wish to know whether the correlation between the two dimensions can be better captured using the bivariate sampling algorithm than with the univariate sampling algorithm. We calculate the Pearson's correlation coefficient between $X_1$ and $X_2$ using the estimated densities with the formula

$$\rho = Corr(X_1, X_2) = \frac{E(X_1X_2) - E(X_1)E(X_2)}{\sqrt{\left[E(X_1^2) - E^2(X_1)\right]\left[E(X_2^2) - E^2(X_2)\right]}}\ , \tag{10}$$

where $E(X_i) = \int_0^\infty x f_i(x)dx$, $E(X_i^2) = \int_0^\infty x^2 f_i(x)dx$, for $i = 1$ and 2, and $E(X_1X_2) = \int_0^\infty \int_0^\infty x_1x_2 f(x_1, x_2)dx_1dx_2$. Using the rectangle method, we wrote R functions to numerically approximate the integrals and the double integral in the above expression. Our programs allow for controlling the accuracy of the integrals. We tested our numerical computation on bivariate normal distributions with known densities and found that the error is less than 0.01%.

# 4 Results and discussion

## 4.1 MCMC results

As previously discussed in Section 3.2, we executed both the the univariate and bivariate sampling algorithms. Table 1 presents the results obtained by running the univariate sampling algorithm for each of the two dimensions, respectively. Any possible correlation between the two dimensions is only captured through the use of product kernel, while the parameter estimation procedure did not take the correlation into account. Table 2 provides the results derived by running the bivariate sampling algorithm for the bivariate data.

To prevent false impressions of convergence, we chose the tuning parameter in the random-walk Metropolis-Hastings algorithm so that the acceptance rate was between 0.2 and 0.3 (see for example, Tse, Zhang and Yu, 2004). The burn-in period was chosen to contain 5,000 iterations, and the number of total recorded iterations was 10,000. The simulation inefficiency factor (SIF) was used to check the mixing performance of the sampling algorithm (see for example, Roberts, 1996). The SIF can be approximated as the number of consecutive draws needed so as to derive independent draws. For example, if the SIF value is 20, we should retain one draw for every 20 draws so that the retained draws are independent (see for example, Kim, Shephard and Chib, 1998; Meyer and Yu, 2000; Zhang, Brooks and King, 2009).

Figure 2 provides graphs for simulated chains based on univariate data, and Figure 3 presents graphs for simulated chains based on bivariate data. In each graph, we plotted the simulated chains for the bandwidth and transformation parameters. According to the SIF values presented in Table 1 and the graphs of the simulated chains presented in Figure 2, we found that the simulated chains of parameters for both variables have achieved very good mixing performance.

Table 2 and the graphs of the simulated chains presented in Figure 3 show that the simulated chains of parameters for the bivariate density estimator have achieved reasonable mixing performance. Even though the SIF values of $\lambda_{11}$ and $\lambda_{21}$ are larger than those of the other parameters, they are well below 100, which is usually considered as a benchmark for a reasonable mixing performance. Therefore we could conclude that the inefficiency of the simulated Markov chains is tolerable in view of the number of iterations.

## 4.2 Accuracy of results obtained through the MCMC algorithms

Let $M_1$ denote the univariate sampling algorithm proposed in Section 3.3.1 and $M_2$ denote the bivariate sampling algorithm proposed in Section 3.3.2. In order to examine the performance of the two algorithms for the estimation of bandwidth parameters and transformation parameters, we computed the value of the correlation coefficient given by (10) and the value of log-likelihood function given by (7) based on parameter estimates obtained through $M_1$ and $M_2$, respectively.

The log-likelihood value calculated through parameter estimates given in Table 1 is -9501.00, and log-likelihood calculated through parameter values given in Table 2 is -7636.26. This indicates a dramatic increase of the log-likelihood value obtained through $M_2$ against $M_1$.

The correlation coefficients approximated through the density estimator given by (3) with parameters estimated through $M_1$ and $M_2$ are 0.2 and 0.26, respectively. This indicates that the bivariate sampling algorithm can better capture the correlation between $X_1$ and $X_2$ than the univariate sampling algorithm. As the sample measures of Pearson's correlation coefficient and Spearman's rank correlation coefficient are respectively, 0.73 and 0.58, we have to say that both $M_1$ and $M_2$ tend to underestimate the correlation between the two dimensions. The reason for this outcome is likely to be the use of the product kernel, or equivalently, the use of a diagonal bandwidth matrix for the bivariate Gaussian kernel. A possible remedy to this problem is to use a full bandwidth matrix at the expense of increased complexity of the resulting bivariate density estimator. We leave this for future research.

## 5 Conclusions

This paper presents Bayesian sampling algorithms for estimating bandwidths and transformation parameters in the kernel density estimation via transformations for bivariate data. The proposed sampling algorithms can estimate not only the bandwidth parameters but also the transformation parameters through a Metropolis-Hastings sampling procedure. Our sampling algorithms have achieved very good mixing performance. When estimating the density of bivariate insurance claim data, we have found that our bivariate sampling algorithm has an improvement over what Bolancé et al. (2008) did, where the transformation parameters were estimated by dealing with each variable separately. We calculate the correlation coefficient through our bivariate sampling algorithm in comparison with that calculated through the univariate sampling algorithm. We have found that the correlation between the two dimensions is better captured via the bivariate sampling algorithm than the univariate sampling algorithm.

We have also computed the conditional tail expectation as in Bolancé et al. (2008). However, our results tend to underestimate the empirical conditional tail expectations. This is not surprising because our sampling algorithms were developed based on the Kullback-Leibler information, under which our results are optimal in terms of the entire density rather than the tails of the density. Further research could focus on finding the optimal bandwidth and transformation parameters for bivariate kernel density estimation via transformations, which give a more accurate estimate of the tail of the joint density.

## 6 Acknowledgements

We wish to thank the Editor Angus Macdonald and two anonymous referees for their very constructive comments that have substantially improved the paper. We also extend our thanks to Professor Montserrat Guillén from the University of Barcelona for providing the automobile insurance data used in the paper.

## References

[1] Bolancé, C., Guillén, M., Nielsen, J.P., 2003. Kernel density estimation of actuarial loss functions, Insurance: Mathematics and Economics, 32, 19-36.

[2] Bolancé, C., Guillén, M., Pelican, E., Vernic, R., 2008. Skewed bivariate models and nonparametric estimation for the CTE risk measure, Insurance: Mathematics and Economics, 43, 386-393.

[3] Bowman, A.W., Azzalini, A., 1997. *Applied Smoothing Techniques for Data Analysis*, Oxford University Press, London.

[4] Buch-Larsen, T., Nielsen, J.P., Guillén, M., Bolancé, C., 2005. Kernel density estimation for heavy-tailed distributions using the Champernowne transformation, Statistics, 39(6), 503-518.

[5] Clements, A.E., Hurn, A.S. and Lindsay, K.A., 2003. Mobius-like mappings and their use in kernel density estimation, Journal of the American Statistical Association, 98, 993-1000.

[6] Härdle, W., 1991. *Smoothing Techniques with Implementation in S*, Springer-Verlag, New York.

[7] Hjort, N.L., Glad, I.K., 1995. Nonparametric density estimation with a parametric start, The Annals of Statistics, 23, 882-904.

[8] Izenman, A.J., 1991. Recent developments in nonparametric density estimation, Journal of the American Statistical Association, 86, 205-224.

[9] Kim, S., Shephard, N., Chib, S., 1998. Stochastic volatility: Likelihood inference and comparison with ARCH models, Review of Economic Studies, 65, 361-393.

[10] Marron, J.S., 1988. Automatic smoothing parameter selection: A survey, Empirical Economics, 13, 187-208.

[11] Meyer, R., Yu, J., 2000. BUGS for a Bayesian analysis of stochastic volatility models, Econometrics Journal, 3, 198-215.

[12] Roberts, G.O., 1996. Markov chain concepts related to sampling algorithms. In: Gilks, W.R. Richardson, S., Spiegelhalter, D.J. (Eds.) *Markov Chain Monte Carlo in Practice*, Chapman & Hall, London, 45-57.

[13] Scott, D.W., 1992. *Multivariate Density Estimation: Theory, Practice and Visualisation*, John Wiley & Sons, New York.

[14] Sheather, S.J., Jones, M.C., 1991. A reliable data-based bandwidth selection method for kernel density estimation, Journal of the Royal Statistical Society, Series B, 53, 683-690.

[15] Simonoff, J.S., 1996. *Smoothing Methods in Statistics*, Springer, New York.

[16] Tse, Y.K., Zhang, X., Yu, J., 2004. Estimation of Hyperbolic Diffusion with Markov Chain Monte Carlo Simulation, Quantitative Finance, 4, 158-169.

[17] Wand, M.P., Jones, M.C., 1995. *Kernel Smoothing*, Chapman & Hall, London.

[18] Wand, M.P., Marron, J.S., Ruppert, D., 1991. Transformations in density estimation, Journal of the American Statistical Association, 86, 414, 343-353.

[19] 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, 21-32.

[20] 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, 3009-3031.

Table 1: MCMC results obtained through the univariate sampling algorithm

| $X_1$ | Estimate | SIF | Acceptance rate | $X_2$ | Estimate | SIF | Acceptance rate |
|---|---|---|---|---|---|---|---|
| $h_1$ | 71.031 | 8.76 | 0.203 | $h_2$ | 54.467 | 19.91 | 0.256 |
| $\lambda_{11}$ | 1760.887 | 24.97 | 0.188 | $\lambda_{12}$ | 43.055 | 54.92 | 0.270 |
| $\lambda_{21}$ | -2.302 | 22.58 | 0.238 | $\lambda_{22}$ | -1.466 | 54.51 | 0.210 |

Table 2: MCMC results obtained through the bivariate sampling algorithm

| $X_1$ | Estimate | SIF | Acceptance rate | $X_2$ | Estimate | SIF | Acceptance rate |
|---|---|---|---|---|---|---|---|
| $h_1$ | 124.138 | 6.78 | 0.299 | $h_2$ | 128.536 | 8.91 | 0.279 |
| $\lambda_{11}$ | 2234.750 | 67.93 | 0.225 | $\lambda_{12}$ | 51.741 | 30.96 | 0.291 |
| $\lambda_{21}$ | -3.030 | 66.12 | 0.235 | $\lambda_{22}$ | -0.814 | 28.57 | 0.257 |

Figure 1: (1) Scatter plot of bodily injury claims versus third party liability claims; and (2) Scatter plot of logarithmic bodily injury claims versus logarithmic third party liability claims.

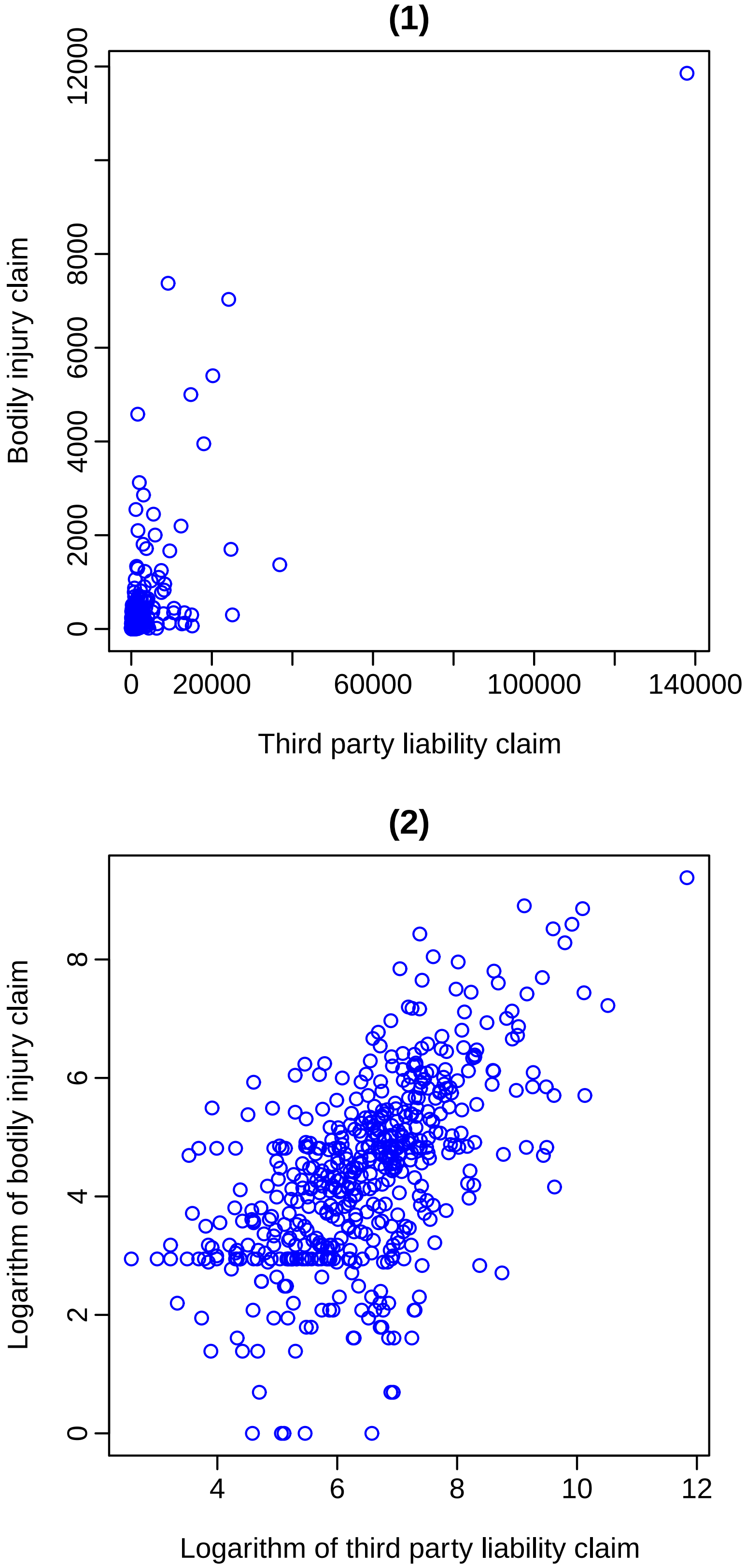


Figure 2: Plots of simulated chains based on univariate data series. The left column contains the simulated chains of $(h, \lambda_1, \lambda_2)$ based on the first series, and the right column contains the simulated chains of the same set of parameters based on the second series. In each of the six graphs, the horizontal axis represents the serial number of draws which retained one draw for every five draws; and the vertical axis represents parameters values.

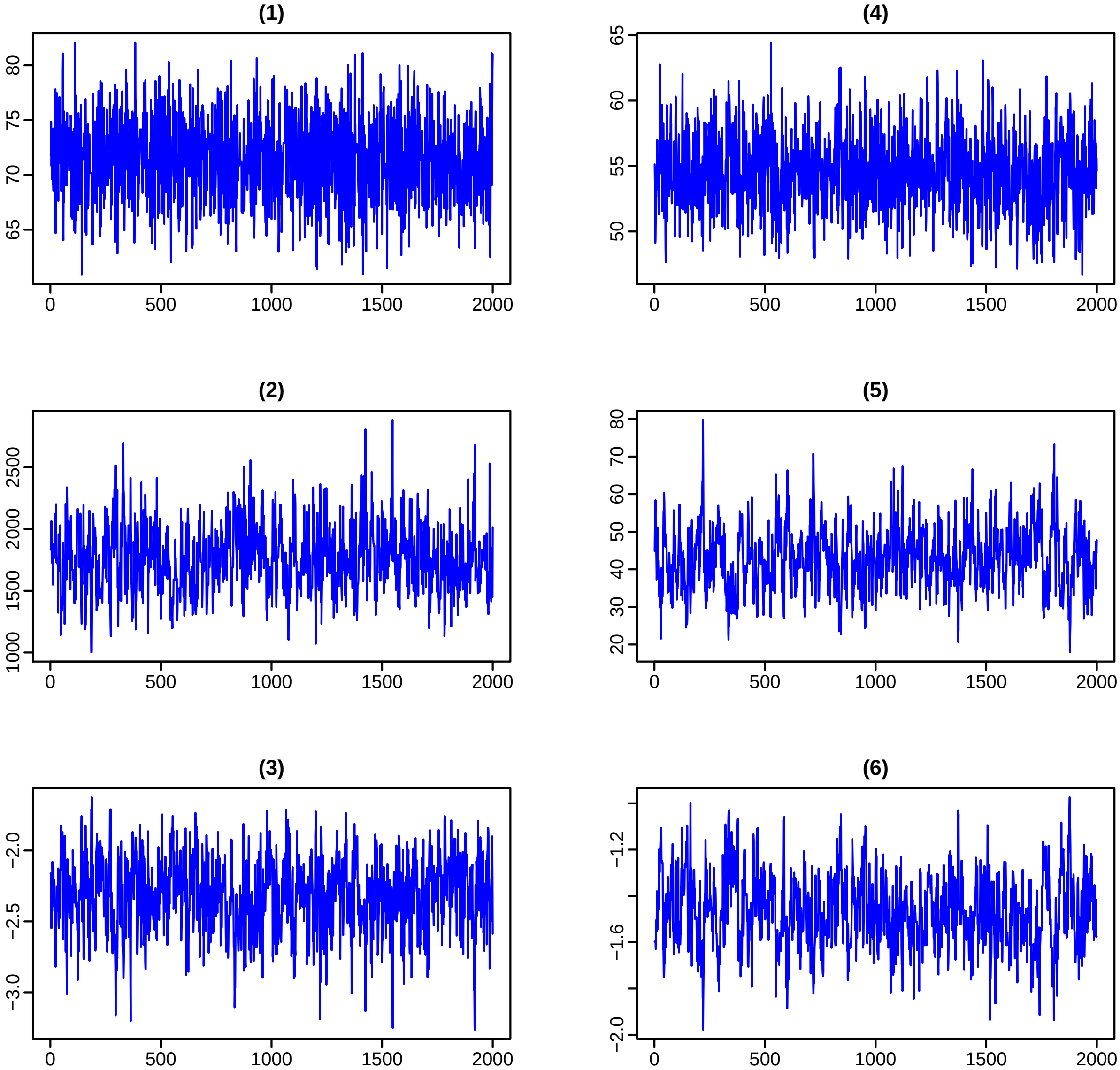
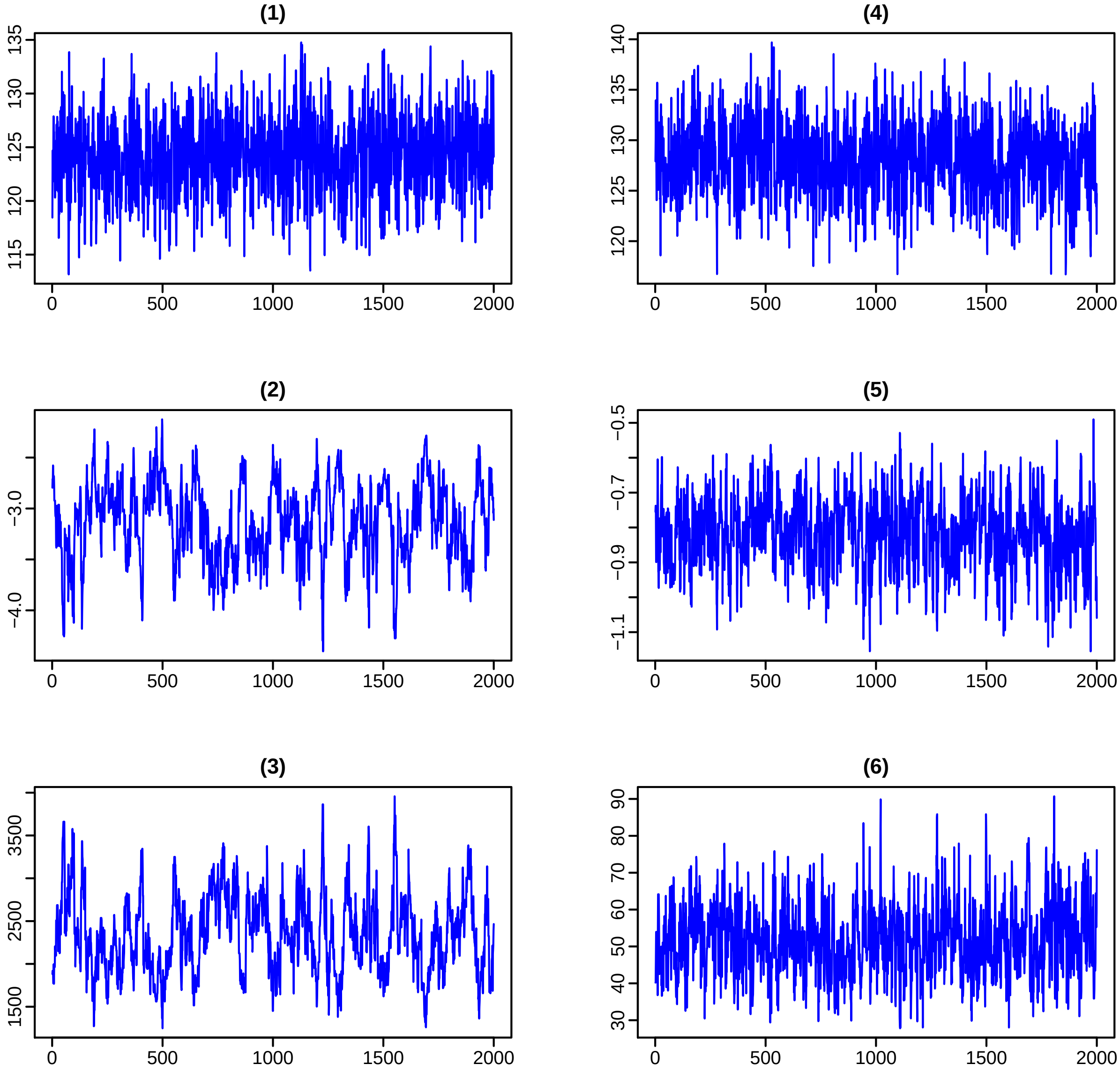


Figure 3: Plots of simulated chains based on bivariate data series. The left column contains the simulated chains of $(h, \lambda_{11}, \lambda_{12})$, and the right column contains the simulated chains of $(h, \lambda_{21}, \lambda_{22})$. In each of the six graphs, the horizontal axis represents the serial number of draws which retained one draw for every five draws; and the vertical axis represents parameters values.


(1)
115
120
125
130
135
0
500
1000
1500
2000
(4)
120
125
130
135
140
0
500
1000
1500
2000
(2)
−4.0
−3.0
0
500
1000
1500
2000
(5)
−1.1
−0.9
−0.7
−0.5
0
500
1000
1500
2000
(3)
1500
2500
3500
0
500
1000
1500
2000
(6)
30
40
50
60
70
80
90
0
500
1000
1500
2000