# StatComp HW2

Jiang Wenxin 16342067

January 8, 2021

## Contents



## 1 Metropolis-Hastings Algorithm

**7.1** The goal of this problem is to investigate the role of the proposal distribution in a Metropolis-Hastings algorithm designed to simulate from the posterior distribution of a parameter $\delta$. In part (a), you are asked to simulate data from a distribution with $\delta$ known. For parts (b)-(d), assume $\delta$ is unknown with a Unif(0,1) prior distribution for $\delta$. For parts (b)-(d), provide an appropriate plot and a table summarizing the output of the algorithm. To facilitate comparisons, use the same number of iterations, random seed, starting values, and burn-in period for all implementations of the algorithm.

a. Simulate 200 realizations from the mixture distribution in Equation (7.6) with $\delta = 0.7$. Draw a histogram of these data.
b. Implement an independence chain MCMC procedure to simulate from the posterior distribution of $\delta$, using your data from part (a).
   Given $\mathbf{X}^{(t)} = \mathbf{x}^{(t)}$, the algorithm generates $\mathbf{X}^{(t+1)}$ as follows:

   (a) Sample a candidate value $\mathbf{X}^*$ from a proposal distribution $g\left(\cdot \mid \mathbf{x}^{(t)}\right)$.

   (b) Compute the Metropolis-Hastings ratio $R\left(\mathbf{x}^{(t)}, \mathbf{X}^*\right)$, where

$$R(\mathbf{x}^{(t)}, \mathbf{X}^*) = \frac{f(\mathbf{X}^*)g(\mathbf{x}^{(t)} \mid \mathbf{X}^*)}{f(\mathbf{x}^{(t)})g(\mathbf{X}^* \mid \mathbf{x}^{(t)})}$$

   (c) Sample a value for $\mathbf{X}^{(t+1)}$ according to the following:

$$\mathbf{X}^{(t+1)} = \begin{cases} \mathbf{X}^* & \text{with probability } \min\left\{R\left(\mathbf{x}^{(t)}, \mathbf{X}^*\right), 1\right\} \\ \mathbf{x}^{(t)} & \text{otherwise.} \end{cases}$$

(d) Increment $t$ and return to step 1.

c. Implement a random walk chain with $\delta^* = \delta^{(t)} + \epsilon$ with $\epsilon \sim \text{Unif}(-1, 1)$.

d. Reparameterize the problem letting $U = \log\{\delta/(1-\delta)\}$ and $U^* = u^{(t)} + \epsilon$. Implement a random walk chain in $U$-space as in Equation (7.8).

e. Compare the estimates and convergence behavior of the three algorithms.

**Ans.**

a. R code available at Section 7.1.1. The histogram of these data shows in Figure 1.

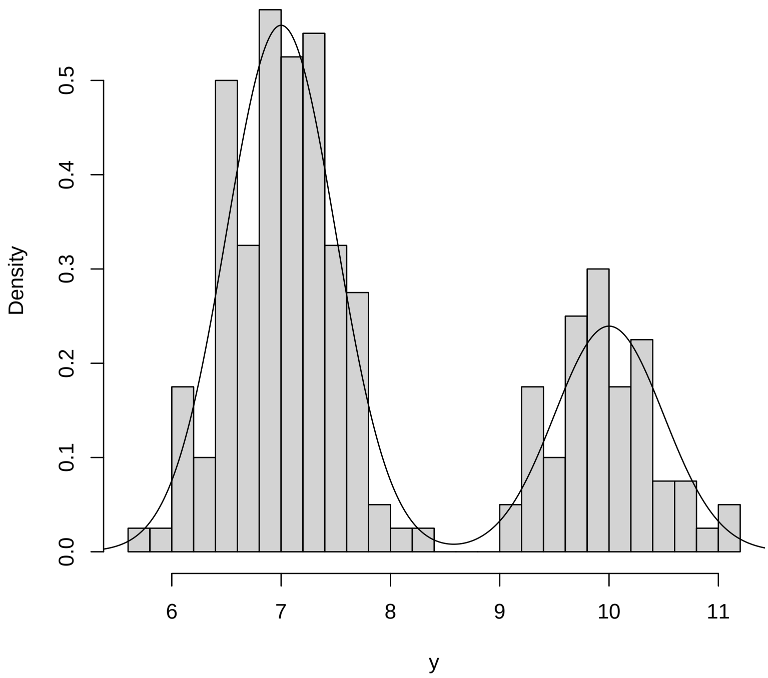



Figure 1: Histogram of Data and Plot of Mixture Distribution

b. In an independence chain, the proposal distribution $g\left(\mathbf{x}^* \mid \mathbf{x}^{(t)}\right) = g\left(\mathbf{x}^*\right)$ and the Metropolis-Hastings ratio is

$$R\left(\mathbf{x}^{(t)}, \mathbf{X}^*\right) = \frac{f\left(\mathbf{X}^*\right) g\left(\mathbf{x}^{(t)}\right)}{f\left(\mathbf{x}^{(t)}\right) g\left(\mathbf{X}^*\right)}.$$

Use the prior distribution Unif(0,1) as a proposal distribution in an independence chain. Thus,

$$R\left(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^*\right) = \frac{L\left(\boldsymbol{\theta}^* \mid \mathbf{y}\right)}{L\left(\boldsymbol{\theta}^{(t)} \mid \mathbf{y}\right)}.$$

R code available at Section 7.1.2. The sample path and the histogram of independence chain show in Figure 2.

c. In a random walk chain $\mathbf{X}^*$ is generated by drawing $\boldsymbol{\epsilon} \sim h(\boldsymbol{\epsilon})$ and $\mathbf{X}^* = \mathbf{x}^{(t)} + \boldsymbol{\epsilon}$. Thus, $g\left(\mathbf{x}^* \mid \mathbf{x}^{(t)}\right) = h\left(\mathbf{x}^* - \mathbf{x}^{(t)}\right)$. Since proposal density $g$ is symmetric about zero. $g\left(\mathbf{x}^* \mid \mathbf{x}^{(t)}\right) = g\left(\mathbf{x}^{(t)} \mid \mathbf{x}^*\right)$. Therefore, the Metropolis-Hastings ratio in this case is

$$R(\delta^{(t)}, \delta^*) = \frac{f\left(\delta^*\right)}{f\left(\delta^{(t)}\right)}.$$

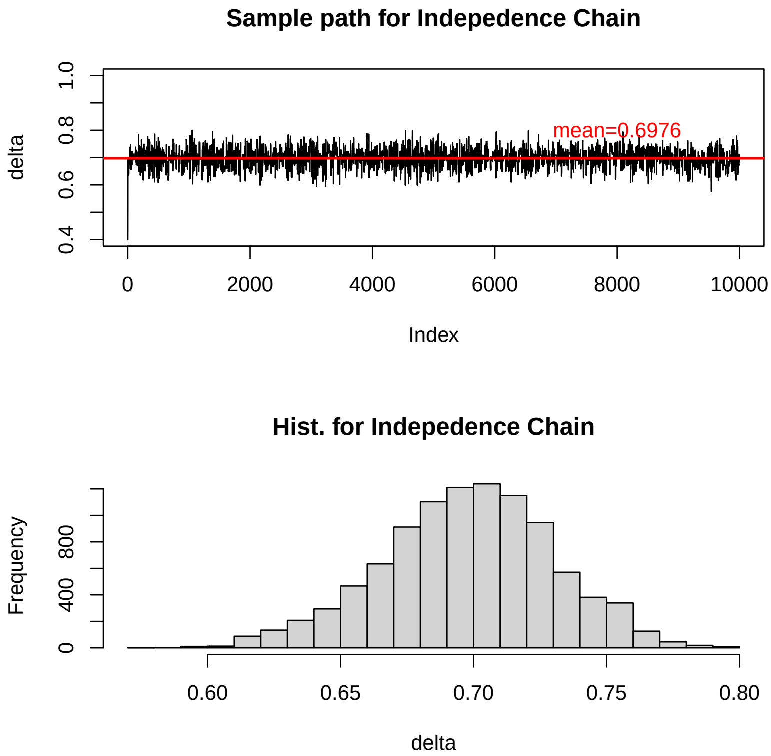



Figure 2: Sample Path and the Histogram of Independence Chain Starting at $\delta = 0.4$

Let $U = \text{logit}\{\delta\} = \log\{\delta/(1-\delta)\}$. R code available at Section 7.1.3. The sample path of random walk chain in $\delta$- space shows in Figure 3.

d. the Metropolis-Hastings ratio in this case is

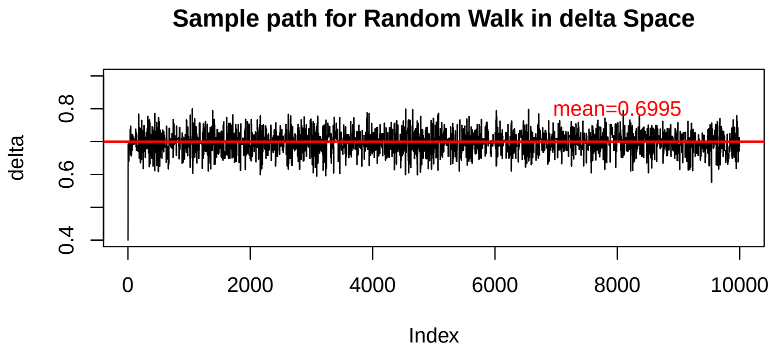
$$R(\delta^{(t)}, \delta^*) = \frac{f\left(\text{logit}^{-1}\{u^*\}\right)|J(u^*)|}{f\left(\text{logit}^{-1}\{u^{(t)}\}\right)\left|J\left(u^{(t)}\right)\right|},$$

where $\delta = \text{logit}^{-1}\{U\} = \exp\{U\}/(1+\exp\{U\})$. R code available at Section 7.1.3. The sample path of random walk chain in U-space shows in Figure 3.

e. The sample path plots above show that all three Metropolis-Hastings algorithms move quickly away from its starting value and seems easily able to sample values from all portions of the parameter space supported by the posterior for $\delta$, which is good mixing. From Table 1, we can obtain that three algorithms converge and do not sensitive to the starting points. This illustrates that proposal distributions we chose could generate reliable estimation of $\delta$.

Table 1: Estimations by Different MH Algorithms with Different Starting Points

| Initial Value | 0.2 | 0.4 | 0.6 | 0.8 |
|---|---|---|---|---|
| Indepedence Chain | 0.6977 | 0.6976 | 0.6986 | 0.6988 |
| Random Walk in delta Space | 0.7001 | 0.6995 | 0.6992 | 0.6989 |
| Random Walk in U Space | 0.6982 | 0.6979 | 0.6979 | 0.6989 |



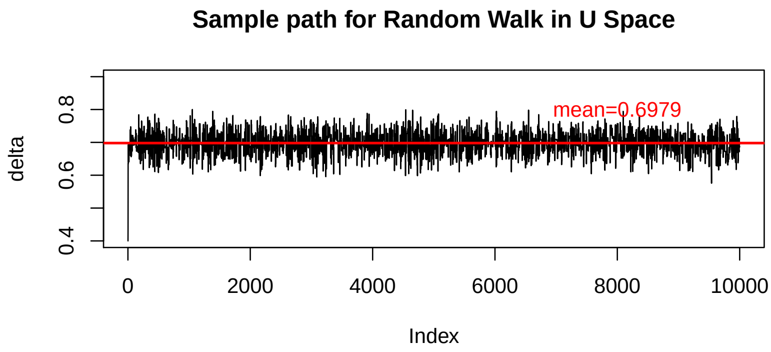




Figure 3: Sample Path of Random Walk in delta Space and U Space Starting at $\delta = 0.4$

## 2 Gibbs Sampler: Poisson Process with Change Point

**7.6** Problem 6.4 introduces data on coal-mining disasters from 1851 to 1962. For these data, assume the model

$$x_j \sim \begin{cases} \text{Poi}\left(\lambda_1\right), & j = 1, \ldots, \theta \\ \text{Poi}\left(\lambda_2\right), & j = \theta + 1, \ldots, 112 \end{cases}$$

Assume $\lambda_i \mid \alpha \sim \text{Gamma}(3, \alpha)$ for $i = 1, 2$, where $\alpha \sim \text{Gamma}(10, 10)$, and assume $\theta$ follows a discrete uniform distribution over $\{1, \ldots, 111\}$. The goal of this problem is to estimate the posterior distribution of the model parameters via a Gibbs sampler.

a. Derive the conditional distributions necessary to carry out Gibbs sampling for the change-point model.
b. Implement the Gibbs sampler. Use a suite of convergence diagnostics to evaluate the convergence and mixing of your sampler.
c. Construct density histograms and a table of summary statistics for the approximate posterior distributions of $\theta, \lambda_1$, and $\lambda_2$. Are symmetric highest posterior density (HPD) intervals appropriate for all of these parameters?
d. Interpret the results in the context of the problem.

**Ans.**

a. Let the set of unknown parameter be $\phi = (\lambda_1, \lambda_2, \theta, \alpha)$. For convenience, donate $n = 112$ as the number of year and

$$\begin{aligned}
\lambda_1 &\sim \Gamma(a_1, \alpha), \\
\lambda_2 &\sim \Gamma(a_2, \alpha), \\
\alpha &\sim \Gamma(b, b), \\
\theta &\sim \text{Unif}(1, n-1).
\end{aligned}$$

Then the posterior is

$$\begin{aligned}
&\pi(\phi \mid \mathbf{x}) \\
&\propto L(\mathbf{x} \mid \phi) p(\lambda_1 \mid \alpha) p(\lambda_2 \mid \alpha) p(\alpha) p(\theta) \\
&= \left[\prod_{i=1}^{\theta} \frac{e^{-\lambda_1} \lambda_1^{x_i}}{x_i!}\right] \left[\prod_{i=\theta+1}^{n} \frac{e^{-\lambda_2} \lambda_2^{x_i}}{x_i!}\right] \left[\frac{\alpha^{a_1}}{\Gamma(a_1)} \lambda_1^{a_1-1} e^{-\alpha\lambda_1}\right] \left[\frac{\alpha^{a_2}}{\Gamma(a_2)} \lambda_2^{a_2-1} e^{-\alpha\lambda_2}\right] \left[\frac{b^b}{\Gamma(b)} \alpha^{b-1} e^{-b\alpha}\right] \frac{1}{n-2} \\
&\propto \left[e^{-(\theta+\alpha)\lambda_1} \lambda_1^{\sum_{i=1}^{\theta} x_i + a_1 - 1}\right] \left[e^{-(n-\theta+\alpha)\lambda_2} \lambda_2^{\sum_{i=\theta+1}^{n} x_i + a_2 - 1}\right] \left[e^{-b\alpha} \alpha^{a_1+a_2+b-1}\right] \\
&\propto \left[e^{-(b+\lambda_1+\lambda_2)\alpha} \alpha^{a_1+a_2+b-1}\right] f(\theta, \lambda_1, \lambda_2 \mid \alpha, \mathbf{x}) \\
&\propto \exp\{\theta(\lambda_2 - \lambda_1)\} \left(\frac{\lambda_2}{\lambda_1}\right)^{\sum_{i=1}^{\theta} x_i} f(\alpha, \lambda_1, \lambda_2 \mid \theta, \mathbf{x}).
\end{aligned}$$

Thus, a Gibbs sampler can be constructed by simulating from the following conditional posterior distributions.

$$\begin{aligned}
\lambda_1 \mid \cdot &\sim \Gamma(a_1 + \sum_{i=1}^{\theta} x_i, \theta + \alpha), \\
\lambda_2 \mid \cdot &\sim \Gamma(a_2 + \sum_{i=\theta+1}^{n} x_i, n - \theta + \alpha), \\
\alpha \mid \cdot &\sim \Gamma(a_1 + a_2 + b, b + \lambda_1 + \lambda_2), \\
\theta \mid \cdot &\sim c \cdot \exp\{(\lambda_2 - \lambda_1)\theta\} \left(\frac{\lambda_1}{\lambda_2}\right)^{\sum_{i=1}^{\theta} x_i},
\end{aligned}$$

where $c$ is a normalization parameter. Then, in this case, MCMC sampling from

$$\begin{aligned}
\lambda_1 \mid \cdot &\sim \Gamma(3 + \sum_{i=1}^{\theta} x_i, \theta + \alpha), \\
\lambda_2 \mid \cdot &\sim \Gamma(3 + \sum_{i=\theta+1}^{112} x_i, 112 - \theta + \alpha), \\
\alpha \mid \cdot &\sim \Gamma(16, 10 + \lambda_1 + \lambda_2), \\
\theta \mid \cdot &\sim c \cdot \exp\{(\lambda_2 - \lambda_1)\theta\} \left(\frac{\lambda_1}{\lambda_2}\right)^{\sum_{i=1}^{\theta} x_i}.
\end{aligned}$$

b. R code at Section 7.2

- **Sample Path Plots:** Figure 4 shows that the chains are mixing well, quickly moves away from its starting value and the sample path will wiggle about vigorously in the region supported by $f$.

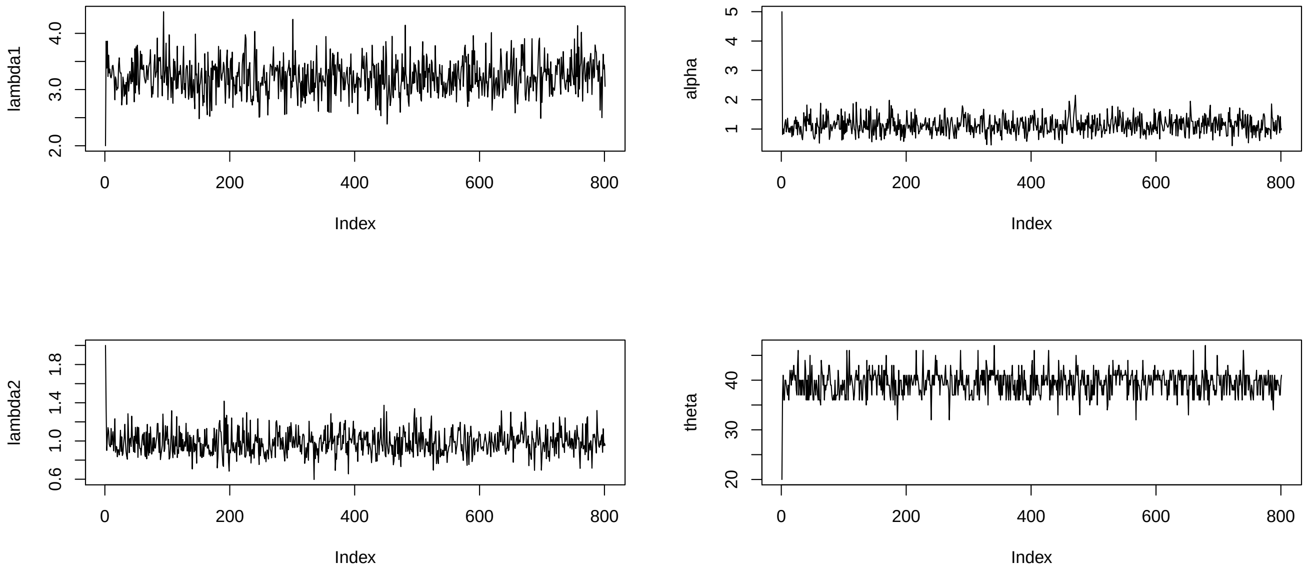



Figure 4: Sample Path Plots of change-points model with Gibbs sampler

- **Cumulative Sum Diagnostic:** Let $h(x) = x$ and set burn-in period equals to 200. Computed using only the iterations of the chain that remain after removing burn-in values, we obtain the cumulative sum of each parameter in $\phi$. The cumsum plots in Figure 5 are very wiggly and have smaller excursions from 0 which indicate the chains are mixing well.

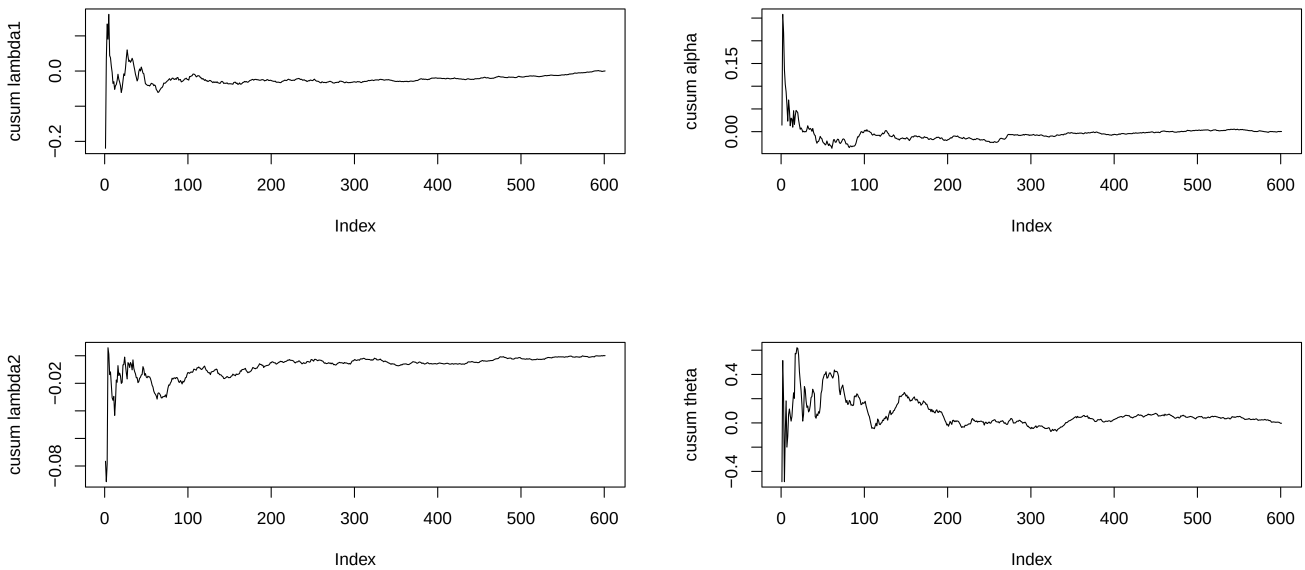



Figure 5: Cumulative Sum Plots of change-points model with Gibbs sampler

- **Correlation Plot:** Figure 6 and Figure 7 show the correlation in the sequence of each parameter and cross parameters at different iteration lags. Cross-correlations are relative low and self-correlations vigorously wiggle about 0. This results indicates that the chains are performing well.

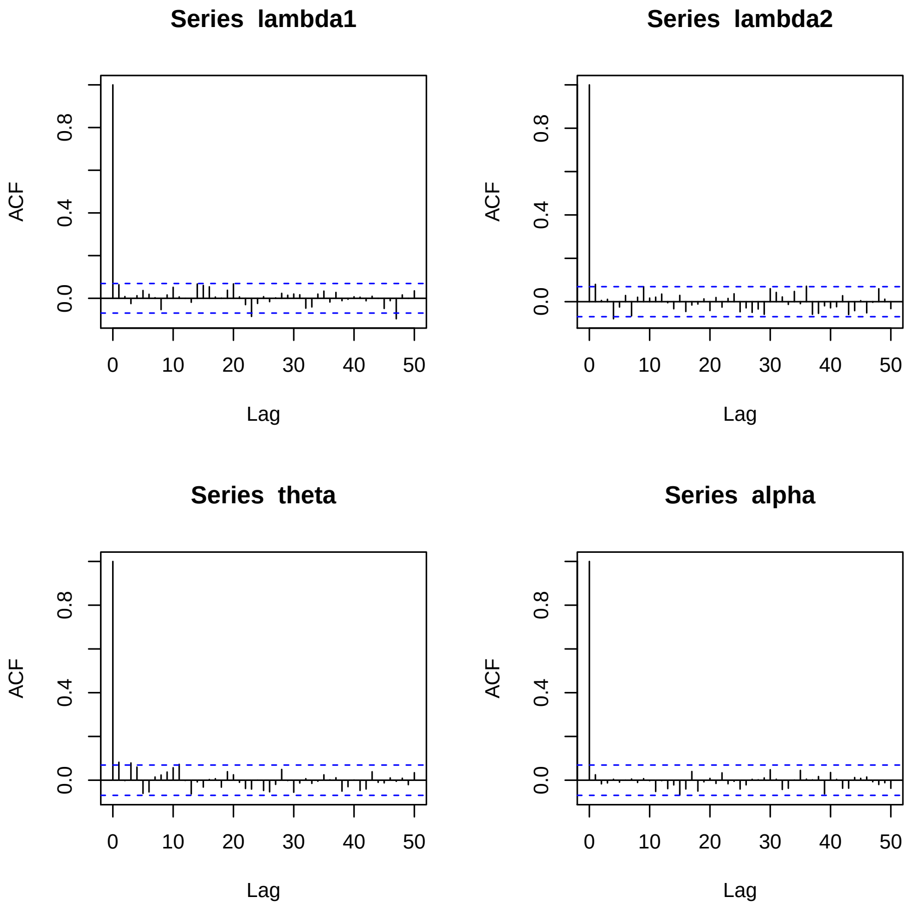



Figure 6: Auto-correlation Plots of change-points model with Gibbs sampler.

c. Figure 8 construct the density histograms of the four parameters and Table 2 summarizes them. Since the distribution of $\lambda_1$ and $\lambda_2$ are unimodal and symmetric, a symmetric HPD is suitable for them. And since the distribution of $\theta$ is multimodal and $\alpha$ are not symmetric, a symmetric HPD is not suitable for them.

d. The discussions above illustrate that chains in Gibbs sampling mixing well.

# 3 Gibbs Sampler: Hierarchical Nested Model

**7.7** Consider a hierarchical nested model

$$Y_{ijk} = \mu + \alpha_i + \beta_{j(i)} + \epsilon_{ijk},$$

Table 2: Summary Statistics

| | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|---|
| $\lambda_1$ | 2.386 | 3.009 | 3.220 | 3.223 | 3.425 | 4.249 |
| $\lambda_2$ | 0.5974 | 0.8999 | 0.9751 | 0.9829 | 1.0543 | 1.3799 |
| $\alpha$ | 0.4302 | 0.9376 | 1.1137 | 1.1310 | 1.3237 | 2.1526 |
| $\theta$ | 32 | 38 | 40 | 39.49 | 41 | 47 |

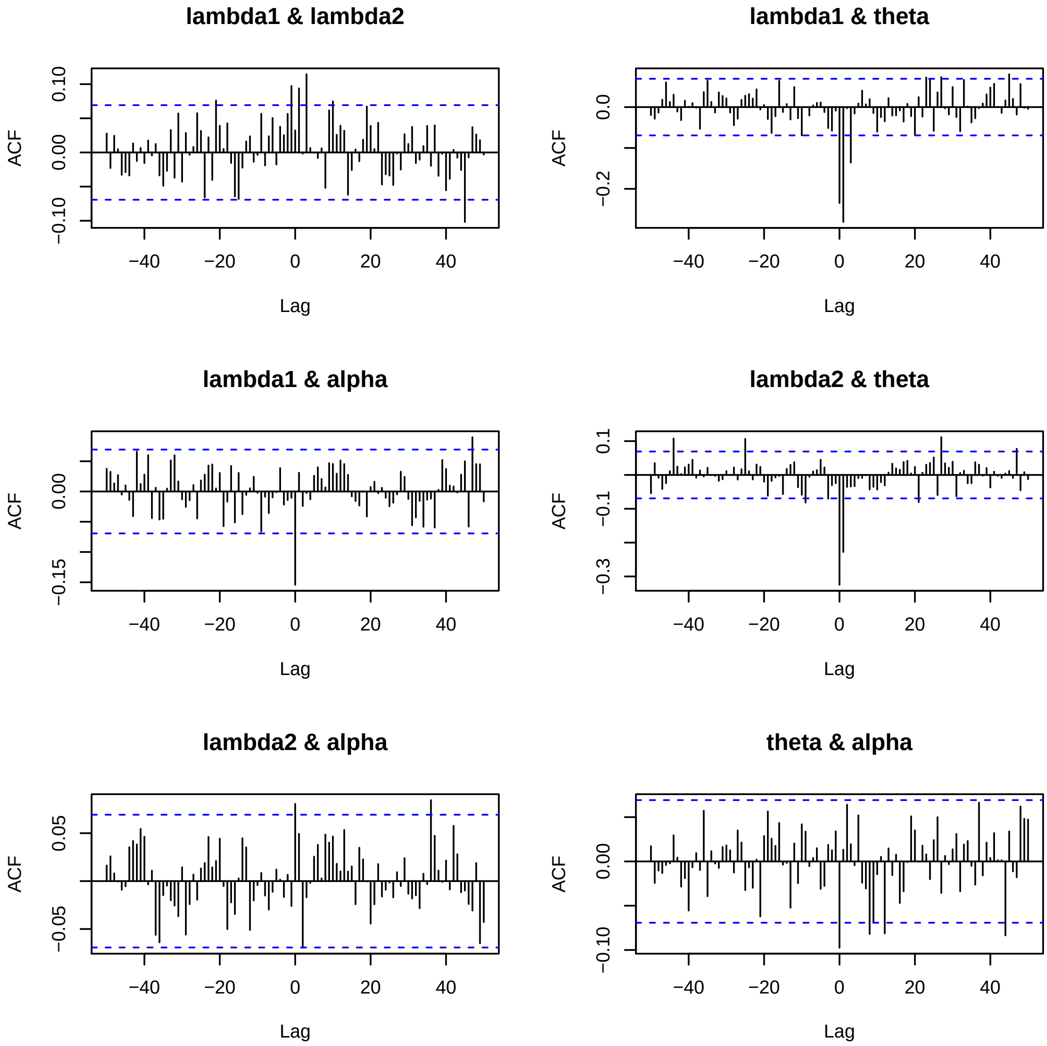



Figure 7: Cross-correlation Plots of change-points model with Gibbs sampler.

where $i = 1, \ldots, I, j = 1, \ldots, J_i$, and $k = 1, \ldots, K$. After averaging over $k$ for each $i$ and $j$, we can rewrite the model (7.29) as

$$Y_{ij} = \mu + \alpha_i + \beta_{j(i)} + \epsilon_{ij}, \quad i = 1, \ldots, I, \quad j = 1, \ldots, J_i,$$

where $Y_{ij} = \sum_{k=1}^{K} Y_{ijk}/K$. Assume that $\alpha_i \sim N\left(0, \sigma_\alpha^2\right), \beta_{j(i)} \sim N\left(0, \sigma_\beta^2\right)$, and $\epsilon_{ij} \sim N\left(0, \sigma_\epsilon^2\right)$, where each set of parameters is independent a priori. Assume that $\sigma_\alpha^2, \sigma_\beta^2$, and $\sigma_\epsilon^2$ are known. To carry out Bayesian inference for this model, assume an improper flat prior for $\mu$, so $f(\mu) \propto 1$. We consider two forms of the Gibbs sampler for this problem.

a. Let $n = \sum_i J_i, y_{\cdot\cdot} = \sum_{ij} y_{ij}/n$, and $y_{i\cdot} = \sum_j y_{ij}/J_i$ hereafter. Show that at iteration $t$, the conditional distributions necessary to carry out Gibbs sampling for this model are given by

$$\mu^{(t+1)} \mid \left(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \mathbf{y}\right) \sim N\left(y_{\cdot\cdot} - \frac{1}{n}\sum_i J_i \alpha_i^{(t)} - \frac{1}{n}\sum_{j(i)} \beta_{j(i)}^{(t)}, \frac{\sigma_\epsilon^2}{n}\right),$$

$$\alpha_i^{(t+1)} \mid \left(\mu^{(t+1)}, \boldsymbol{\beta}^{(t)}, \mathbf{y}\right) \sim N\left(\frac{J_i V_1}{\sigma_\epsilon^2}\left(y_{i\cdot} - \mu^{(t+1)} - \frac{1}{J_i}\sum_{j(i)} \beta_{j(i)}^{(t)}\right), V_1\right),$$

$$\beta_{j(i)}^{(t+1)} \mid \left(\mu^{(t+1)}, \boldsymbol{\alpha}^{(t+1)}, \mathbf{y}\right) \sim N\left(\frac{V_2}{\sigma_\epsilon^2}\left(y_{ij} - \mu^{(t+1)} - \alpha_i^{(t+1)}\right), V_2\right),$$

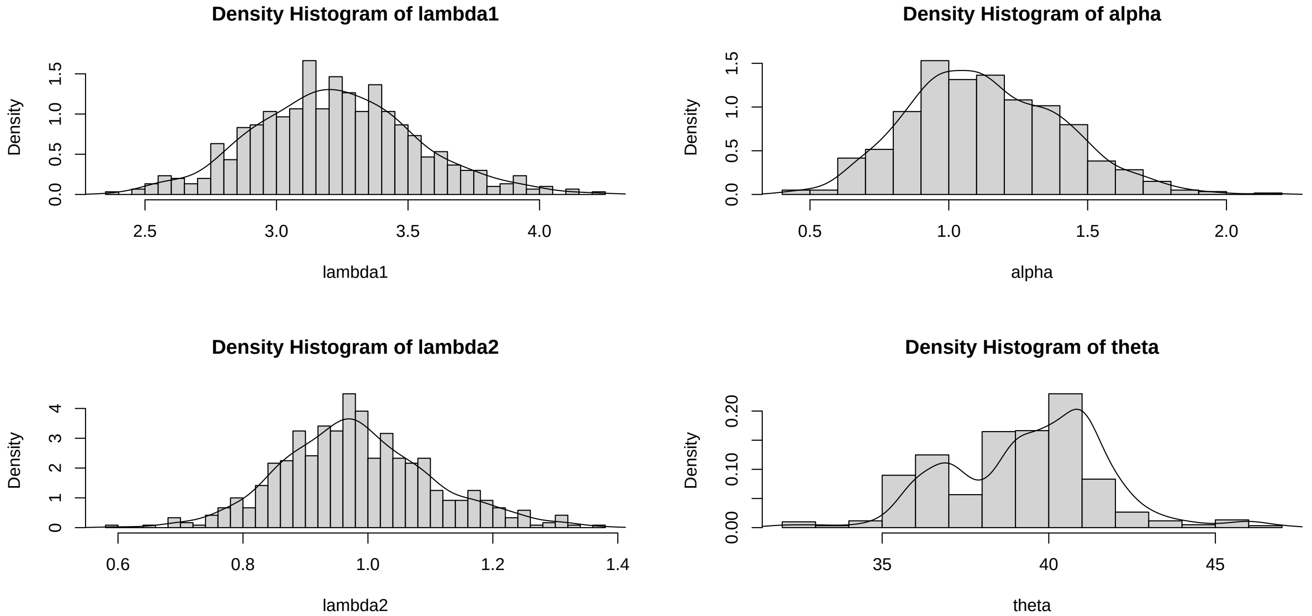



Figure 8: Density histograms of change-points model with Gibbs sampler

where

$$V_1 = \left(\frac{J_i}{\sigma_\epsilon^2} + \frac{1}{\sigma_\alpha^2}\right)^{-1} \quad \text{and} \quad V_2 = \left(\frac{1}{\sigma_\epsilon^2} + \frac{1}{\sigma_\beta^2}\right)^{-1}.$$

b. The convergence rate for a Gibbs sampler can sometimes be improved via reparameterization. For this model, the model can be reparameterized via hierarchical centering (Section 7.3 .1 .4 ). For example, let $Y_{ij}$ follow (7.30), but now let $\eta_{ij} = \mu + \alpha_i + \beta_{j(i)}$ and $\epsilon_{ij} \sim N\left(0, \sigma_\epsilon^2\right)$. Then let $\gamma_i = \mu + \alpha_i$ with $\eta_{ij} \mid \gamma_i \sim N\left(\gamma_i, \sigma_\beta^2\right)$ and $\gamma_i \mid \mu \sim N\left(\mu, \sigma_\alpha^2\right)$. As above, assume $\sigma_\alpha^2, \sigma_\beta^2$, and $\sigma_\epsilon^2$ are known, and assume a flat prior for $\mu$. Show that the conditional distributions necessary to carry out Gibbs sampling for this model are given by

$$\mu^{(t+1)} \mid \left(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\eta}^{(t)}, \mathbf{y}\right) \sim N\left(\frac{1}{I}\sum_i \gamma_i^{(t)}, \frac{1}{I}\sigma_\alpha^2\right),$$

$$\gamma_i^{(t+1)} \mid \left(\mu^{(t+1)}, \boldsymbol{\eta}^{(t)}, \mathbf{y}\right) \sim N\left(V_3\left(\frac{1}{\sigma_\beta^2}\sum_j \eta_{ij}^{(t)} + \frac{\mu^{(t+1)}}{\sigma_\alpha^2}\right), V_3\right),$$

$$\eta_{ij}^{(t+1)} \mid \left(\mu^{(t+1)}, \boldsymbol{\gamma}^{(t+1)}, \mathbf{y}\right) \sim N\left(V_2\left(\frac{y_{ij}}{\sigma_\epsilon^2} + \frac{\gamma_i^{(t+1)}}{\sigma_\beta^2}\right), V_2\right),$$

where

$$V_3 = \left(\frac{J_i}{\sigma_\beta^2} + \frac{1}{\sigma_\alpha^2}\right)^{-1}.$$

**Ans.**

a. Let $\boldsymbol{\phi} = (\mu, \boldsymbol{\alpha}, \boldsymbol{\beta})$ be the set of unknown parameters. Then the posterior is

$$
\begin{aligned}
\pi(\boldsymbol{\phi} \mid \mathbf{y}) \propto & L(\mathbf{y} \mid \boldsymbol{\phi}) f(\mu) \prod_i p(\alpha_i) \prod_{j(i)} p(\beta_{j(i)}) \\
\propto & \prod_i \prod_{j(i)} \exp\left\{-\frac{(y_{ij} - \mu - \alpha_i - \beta_{j(i)})^2}{2\sigma_\epsilon^2}\right\} \exp\left\{-\frac{\alpha_i^2}{2\sigma_\alpha^2}\right\} \exp\left\{-\frac{\beta_{j(i)}^2}{2\sigma_\beta^2}\right\} \\
= & \exp\left\{-\frac{1}{2\sigma_\epsilon^2} \sum_i \sum_{j(i)} \left(y_{ij} - \alpha_i - \beta_{j(i)} - \mu\right)^2 - \frac{1}{2\sigma_\alpha^2} \sum_i \alpha_i^2 - \frac{1}{2\sigma_\beta^2} \sum_i \sum_{j(i)} \beta_{j(i)}^2\right\}.
\end{aligned}
$$

Thus, we have

$$
\begin{aligned}
\pi(\boldsymbol{\phi} \mid \mathbf{y}) &\propto \exp\left\{-\frac{1}{2\sigma_\epsilon^2}\left[\mu - \left(y_{..} - \frac{1}{n}\sum_i J_i \alpha_i - \frac{1}{n}\sum_{j(i)} \beta_{j(i)}\right)\right]^2\right\} f(\boldsymbol{\alpha}, \boldsymbol{\beta} \mid \mu), \\
\pi(\boldsymbol{\phi} \mid \mathbf{y}) &\propto \exp\left\{-\frac{1}{2\sigma_\epsilon^2}\sum_i J_i \left[\alpha_i - \left(y_{i.} - \mu - \frac{1}{J_i}\sum_{j(i)} \beta_{j(i)}\right)\right]^2 - \frac{\sum_i \alpha_i^2}{2\sigma_\alpha^2}\right\} f(\mu, \boldsymbol{\beta} \mid \boldsymbol{\alpha}) \\
&\propto \prod_i \exp\left\{-\frac{1}{2V_1}\left[\alpha_i - \frac{J_i V_1}{\sigma_\epsilon^2}\left(y_{i.} - \mu - \frac{1}{J_i}\sum_{j(i)} \beta_{j(i)}\right)\right]^2\right\} f(\mu, \boldsymbol{\beta} \mid \boldsymbol{\alpha}), \\
\pi(\boldsymbol{\phi} \mid \mathbf{y}) &\propto \exp\left\{-\frac{1}{2}\sum_i \sum_{j(i)} \left[\frac{1}{\sigma_\epsilon^2}\left[\beta_{j(i)} - (y_{ij} - \mu - \alpha_i)\right]^2 + \frac{\beta_{j(i)}^2}{\sigma_\beta^2}\right]\right\} f(\mu, \boldsymbol{\alpha} \mid \boldsymbol{\beta}) \\
&\propto \prod_i \prod_{j(i)} \exp\left\{-\frac{1}{2V_2}\left[\beta_{j(i)} - \frac{V_2}{\sigma_\epsilon^2}(y_{ij} - \mu - \alpha_i)\right]^2\right\} f(\mu, \boldsymbol{\alpha} \mid \boldsymbol{\beta}).
\end{aligned}
$$

□

b. The translation could be rewritten as

$$
\begin{aligned}
y_{ij} &= \eta_{ij} + \epsilon_{ij}, \\
\alpha_i &= \gamma_i - \mu, \\
\beta_{j(i)} &= \eta_{ij} - \gamma_i.
\end{aligned}
$$

Let $\boldsymbol{\psi} = (\mu, \gamma, \eta)$ be the set of unknown parameters. Then the posterior is

$$
\begin{aligned}
\pi(\boldsymbol{\psi} \mid \mathbf{y}) \propto & L(\mathbf{y} \mid \boldsymbol{\psi}) f(\mu) \prod_i \prod_{j(i)} p(\eta_{ij} \mid \gamma_i) p(\gamma_i) \\
\propto & \prod_i \prod_{j(i)} \exp\left\{-\frac{(y_{ij} - \eta_{ij})^2}{2\sigma_\epsilon^2}\right\} \exp\left\{-\frac{(\eta_{ij} - \gamma_i)^2}{2\sigma_\beta^2}\right\} \exp\left\{-\frac{(\gamma_i - \mu)^2}{2\sigma_\alpha^2}\right\} \\
= & \exp\left\{-\frac{1}{2}\sum_{i, j(i)} \left[\frac{(y_{ij} - \eta_{ij})^2}{\sigma_\epsilon^2} + \frac{(\eta_{ij} - \gamma_i)^2}{\sigma_\beta^2} + \frac{(\gamma_i - \mu)^2}{\sigma_\alpha^2}\right]\right\}.
\end{aligned}
$$

Thus, we have

$$
\begin{aligned}
\pi(\boldsymbol{\psi} \mid \mathbf{y}) &\propto \exp\left\{-\frac{I}{2\sigma_\alpha^2}\left[\mu - \frac{1}{I}\sum_i \gamma_i\right]^2\right\} f(\boldsymbol{\gamma}, \boldsymbol{\eta} \mid \mu), \\
\pi(\boldsymbol{\psi} \mid \mathbf{y}) &\propto \prod_i \exp\left\{-\frac{1}{2}\left[\sum_{j(i)} \frac{(\eta_{ij}-\gamma_i)^2}{\sigma_\beta^2} + \frac{(\gamma_i-\mu)^2}{\sigma_\alpha^2}\right]\right\} f(\mu, \boldsymbol{\eta} \mid \boldsymbol{\gamma}) \\
&\propto \prod_i \exp\left\{-\frac{1}{2V_3}\left[\gamma_i - V_3\left(\frac{1}{\sigma_\beta^2}\sum_j \eta_{ij} + \frac{\mu}{\sigma_\alpha^2}\right)\right]^2\right\} f(\mu, \boldsymbol{\eta} \mid \boldsymbol{\gamma}), \\
\pi(\boldsymbol{\psi} \mid \mathbf{y}) &\propto \prod_{i,j(i)} \exp\left\{-\frac{1}{2}\left[\frac{(y_{ij}-\eta_{ij})^2}{\sigma_\epsilon^2} + \frac{(\eta_{ij}-\gamma_i)^2}{\sigma_\beta^2}\right]\right\} f(\mu, \boldsymbol{\gamma} \mid \boldsymbol{\eta}) \\
&\propto \prod_{i,j(i)} \exp\left\{-\frac{1}{2V_2}\left[\eta_{ij} - V_2\left(\frac{y_{ij}}{\sigma_\epsilon^2} + \frac{\gamma_i}{\sigma_\beta^2}\right)\right]^2\right\} f(\mu, \boldsymbol{\gamma} \mid \boldsymbol{\eta}).
\end{aligned}
$$

□

## 4 LU 分解

**Li 5.5** 列主元的高斯消元法得到三角形矩阵及相应方程组的过程为:

$$
\begin{aligned}
A^{(n-1)} &= M^{(n-1)} P\left(n-1, s_{n-1}\right) \ldots M^{(2)} P\left(2, s_2\right) M^{(1)} P\left(1, s_1\right) A, \\
A^{(n-1)} \boldsymbol{x} &= M^{(n-1)} P\left(n-1, s_{n-1}\right) \ldots M^{(2)} P\left(2, s_2\right) M^{(1)} P\left(1, s_1\right) \boldsymbol{b}.
\end{aligned}
$$

相当于把原始矩阵 $A$ 做了如下 LU 分解:

$$PA = LU.$$

其中 $P$ 是一个置换矩阵，$P(i,j)$ 交换第 $i,j$ 行:

$$P = P\left(n-1, s_{n-1}\right) \ldots P\left(2, s_2\right) P\left(1, s_1\right).$$

定义 $n$ 维向量 $\boldsymbol{m}^{(j)}$ 为

$$m_i^{(j)} = \begin{cases} 0, & i \leq j \\ a_{ij}^{(j-1)} / a_{jj}^{(j-1)}, & i > j \end{cases}$$

则初等变换矩阵 $M^{(j)}$ 可以表示为

$$M^{(j)} = I_n - \boldsymbol{m}^{(j)} \boldsymbol{e}_j^T.$$

将行置换打乱次序的 $\boldsymbol{m}^{(j)}$ 记为 $\boldsymbol{m}_*^{(j)}$ 则

$$\boldsymbol{m}_*^{(j)} = P\left(n-1, s_{n-1}\right) \ldots P\left(j+1, s_{j+1}\right) \boldsymbol{m}^{(j)},$$

所以 $L$ 可以表示为

$$L = I_n + \boldsymbol{m}_*^{(1)} \boldsymbol{e}_1^T + \cdots + \boldsymbol{m}_*^{(n-1)} \boldsymbol{e}_{n-1}^T.$$

设 $A^{(0)} = A$,

$$A^{(j)} = M^{(j)} A^{(j-1)}, j = 1, 2, \ldots, n-1,$$

而

$$U = A^{(n-1)}.$$

证明用列主元法进行矩阵 LU 分解的算法正确。

a. 对 $M^{(i)} = I_n - \boldsymbol{m}^{(i)}\boldsymbol{e}_i^T$ 矩阵，证明当 $k, j > i$ 时有

$$P(k,j)M^{(i)} = \left[I_n - P(k,j)\boldsymbol{m}^{(i)}\boldsymbol{e}_i^T\right] P(k,j).$$

b. 令 $M_*^{(k)} = I_n - \boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T$, 证明

$$\begin{aligned}&M^{(n-1)}P\left(n-1,s_{n-1}\right)\dots M^{(2)}P\left(2,s_2\right)M^{(1)}P\left(1,s_1\right)\\=&M_*^{(n-1)}\dots M_*^{(1)}P\left(n-1,s_{n-1}\right)\dots P\left(1,s_1\right).\end{aligned}$$

c. 证明

$$\left(M_*^{(k+1)}M_*^{(k)}\right)^{-1} = I_n + \boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T + \boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_{k+1}^T.$$

d. 证明 $PA = LU$ 成立。

**证明.**

a. From the definition of $\boldsymbol{m}^{(i)}$ we know that the $k, j$ th column of $\boldsymbol{m}^{(i)}\boldsymbol{e}_i^T$ equal to zero. Thus, we have

$$P(k,j)\boldsymbol{m}^{(i)}\boldsymbol{e}_i^T = P(k,j)\boldsymbol{m}^{(i)}\boldsymbol{e}_i^T P(k,j).$$

Therefore,

$$P(k,j)M^{(i)} = P(k,j)\left[I_n - \boldsymbol{m}^{(i)}\boldsymbol{e}_i^T\right] = \left[I_n - P(k,j)\boldsymbol{m}^{(i)}\boldsymbol{e}_i^T\right] P(k,j), \forall k \neq i, j \neq i.$$

□

b. Since $M_*^{(k)} = I_n - \boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T = I_n - P\left(n-1,s_{n-1}\right)\dots P\left(k+1,s_{k+1}\right)\boldsymbol{m}^{(k)}\boldsymbol{e}_k^T$ and $M_*^{(n-1)} = M^{(n-1)}$, we have

$$\begin{aligned}&M^{(n-1)}P\left(n-1,s_{n-1}\right)\dots M^{(2)}P\left(2,s_2\right)M^{(1)}P\left(1,s_1\right)\\=&M_*^{(n-1)}\left[I_n - P\left(n-1,s_{n-1}\right)\boldsymbol{m}^{(n-2)}\boldsymbol{e}_{n-2}^T\right]P\left(n-1,s_{n-1}\right)P\left(n-2,s_{n-2}\right)\dots M^{(1)}P\left(1,s_1\right)\\=&M_*^{(n-1)}\left[I_n - \boldsymbol{m}_*^{(n-2)}\boldsymbol{e}_{n-2}^T\right]P\left(n-1,s_{n-1}\right)P\left(n-2,s_{n-2}\right)M^{(n-3)}\dots M^{(1)}P\left(1,s_1\right)\\=&M_*^{(n-1)}M_*^{(n-2)}P\left(n-1,s_{n-1}\right)P\left(n-2,s_{n-2}\right)M^{(n-3)}\dots M^{(1)}P\left(1,s_1\right)\\=&\dots\\=&M_*^{(n-1)}\dots M_*^{(1)}P\left(n-1,s_{n-1}\right)\dots P\left(1,s_1\right).\end{aligned}$$

□

c. From the definition, we can obtain:

$$\begin{aligned}\boldsymbol{m}_*^{(k)} &= \boldsymbol{m}_*^{(k+1)}P(k+1,s_{k+1})\\\boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T &= \boldsymbol{m}_*^{(k+1)}P(k+1,s_{k+1})\boldsymbol{e}_k^T = \boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_k^T\\\boldsymbol{e}_k^T\boldsymbol{m}_*^{(k+1)} &= \boldsymbol{e}_{k+1}^T\boldsymbol{m}_*^{(k+1)} = 0.\end{aligned}$$

Thus,

$$\begin{aligned}&\left(M_*^{(k+1)}M_*^{(k)}\right)^{-1}\\=&\left[\left(I_n - \boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_{k+1}^T\right)\left(I_n - \boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T\right)\right]^{-1}\\=&\left[I_n - \boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_{k+1}^T - \boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T + \boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_{k+1}^T\boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T\right]^{-1}\\=&\left[I_n - \boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_{k+1}^T - \boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T + \boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_{k+1}^T\boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_k^T\right]^{-1}\\=&\left[I_n - \boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_{k+1}^T - \boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T\right]^{-1}\\\overset{(*)}{=}&I_n + \boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T + \boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_{k+1}^T.\end{aligned}$$

(*) Note that

$$
\begin{aligned}
&\left[I_n-\boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_{k+1}^T-\boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T\right]\left[I_n+\boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T+\boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_{k+1}^T\right]\\
=&\left[I_n-\boldsymbol{m}_*^{(k+1)}\left(\boldsymbol{e}_{k+1}^T+\boldsymbol{e}_k^T\right)\right]\left[I_n+\boldsymbol{m}_*^{(k+1)}\left(\boldsymbol{e}_{k+1}^T+\boldsymbol{e}_k^T\right)\right]\\
=&I_n-\boldsymbol{m}_*^{(k+1)}\left(\boldsymbol{e}_{k+1}^T+\boldsymbol{e}_k^T\right)\boldsymbol{m}_*^{(k+1)}\left(\boldsymbol{e}_{k+1}^T+\boldsymbol{e}_k^T\right)\\
=&I_n-0\\
=&I_n.
\end{aligned}
$$

□

d. Similar to c,

$$
\begin{aligned}
&M_*^{(k+2)}M_*^{(k+1)}M_*^{(k)}\\
=&\left(I_n-\boldsymbol{m}_*^{(k+2)}\boldsymbol{e}_{k+2}^T\right)\left[I_n-\boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_{k+1}^T-\boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T\right]\\
=&\left[I_n-\boldsymbol{m}_*^{(k+2)}\boldsymbol{e}_{k+2}^T-\boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_{k+1}^T-\boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T+\boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_{k+1}^T\boldsymbol{m}_*^{(k+2)}\boldsymbol{e}_{k+2}^T+\boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T\boldsymbol{m}_*^{(k+2)}\boldsymbol{e}_{k+2}^T\right]\\
=&I_n-\boldsymbol{m}_*^{(k+2)}\boldsymbol{e}_{k+2}^T-\boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_{k+1}^T-\boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T\\
=&\left[I_n-\boldsymbol{m}_*^{(k+2)}\boldsymbol{e}_{k+2}^T-\boldsymbol{m}_*^{(k+1)}\boldsymbol{e}_{k+1}^T-\boldsymbol{m}_*^{(k)}\boldsymbol{e}_k^T\right]^{-1}.
\end{aligned}
$$

Thus, generalizing the result in c, we can obtain that

$$
L=I_n+\boldsymbol{m}_*^{(1)}\boldsymbol{e}_1^T+\cdots+\boldsymbol{m}_*^{(n-1)}\boldsymbol{e}_{n-1}^T=\left(M_*^{(n-1)}\dots M_*^{(1)}\right)^{-1}.
$$

Hence,

$$
\begin{aligned}
LU=&LA^{(n-1)}\\
=&\left(M_*^{(n-1)}\dots M_*^{(1)}\right)^{-1}M^{(n-1)}P\left(n-1,s_{n-1}\right)\dots M^{(2)}P\left(2,s_2\right)M^{(1)}P\left(1,s_1\right)A\\
=&\left(M_*^{(n-1)}\dots M_*^{(1)}\right)^{-1}M_*^{(n-1)}\dots M_*^{(1)}P\left(n-1,s_{n-1}\right)\dots P\left(1,s_1\right)A\\
=&PA.
\end{aligned}
$$

□

# 5 证明矩阵范数公式

**Li 5.12** 定义 $\|A\|_p=\sup_{\|\boldsymbol{x}\|_p=1}\|A\boldsymbol{x}\|_p$ 称为 $A$ 的 $p$ 范数。证明，

$$
\begin{aligned}
\|A\|_1&=\max_{1\le j\le n}\sum_{i=1}^n|a_{ij}|\\
\|A\|_2&=\sqrt{A^TA\text{ 的最大特征值}},\\
\|A\|_\infty&=\max_{1\le i\le n}\sum_{j=1}^n|a_{ij}|
\end{aligned}
$$

**证明.**

a. **1-norm:**

$$\begin{aligned}
\|\mathbf{A}\mathbf{x}\|_1 &= \sum_{i=1}^{n} \left| \sum_{j=1}^{n} a_{ij} x_j \right| \\
&\le \sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}| |x_j| = \sum_{j=1}^{n} \left( \sum_{i=1}^{n} |a_{ij}| \right) |x_j| \\
&\le \sum_{j=1}^{n} \left( \max_{i<k<n} \sum_{i=1}^{n} |a_{ik}| \right) |x_j| = \left( \max_{1<j<n} \sum_{i=1}^{n} |a_{ij}| \right) \left( \sum_{j=1}^{n} |x_j| \right) \\
&= \left( \max_{1<j<n} \sum_{i=1}^{n} |a_{ij}| \right) \|\mathbf{x}\|_1
\end{aligned}$$

$$\|\mathbf{A}\|_1 = \sup_{\|\mathbf{x}\|_1=1} \|\mathbf{A}\mathbf{x}\|_1 \le \max_{1<j<n} \sum_{i=1}^{n} |a_{ij}|$$

Suppose the maximum in the last expression is attained for $j = p$. We now choose $\mathbf{x}$ such that

$$x_p = 1 \quad \text{and} \quad x_j = 0, \quad j \ne p$$

when $\|\mathbf{x}\|_1 = 1$ and

$$\|\mathbf{A}\mathbf{x}\|_1 = \sum_{i=1}^{n} |a_{ip}| .$$

Therefore,

$$\|\mathbf{A}\|_1 = \max_{1\le j\le n} \sum_{i=1}^{n} |a_{ij}| .$$

□

b. **2-norm:** From the 2-norm definition of the vector, we know that

$$\|\mathbf{A}\mathbf{x}\|_2^2 = (\mathbf{A}\mathbf{x})^T \mathbf{A}\mathbf{x} = \mathbf{x}^T \mathbf{A}^T \mathbf{A}\mathbf{x} = \mathbf{x}^T \mathbf{B}\mathbf{x},$$

where $\mathbf{B} = \mathbf{A}^T\mathbf{A}$ is a symmetric matrix. Thus, there exists an orthogonal matrix T such that

$$\mathbf{x}^T\mathbf{B}\mathbf{x} = \mathbf{x}^T\mathbf{T}\boldsymbol{\Lambda}\mathbf{T}^T\mathbf{x} = \mathbf{y}^T\boldsymbol{\Lambda}\mathbf{y},$$

where $\Lambda$ is the diagonal matrix formed from the eigenvalues of $\mathbf{B}$ and $\mathbf{y} = \mathbf{T}^T\mathbf{x}$. Since $\mathbf{T}$ is orthogonal, $\mathbf{T}^T$ is orthogonal and $\|\mathbf{x}\|_2 = \|\mathbf{y}\|_2$. Thus $\|\mathbf{y}\|_2 = 1$, so that

$$\sum_{i=1}^{n} y_i^2 = 1,$$

and

$$\|\mathbf{A}\mathbf{x}\|_2^2 = \mathbf{y}^T\boldsymbol{\Lambda}\mathbf{y} = \lambda_1 y_1^2 + \lambda_2 y_2^2 + \cdots + \lambda_n y_n^2.$$

As $\mathbf{B} = \mathbf{A}^T\mathbf{A}$ is semi-positive definite all its eigenvalues are non-negative. We assume that they are arranged in the order $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n \ge 0$. Hence,

$$\|\mathbf{A}\mathbf{x}\|_2^2 \le \lambda_1 \left(y_1^2 + y_2^2 + \ldots + y_n^2\right) = \lambda_1.$$

Now choose $x$ such that $y = [1, 0, 0, \ldots, 0]^T$, when $\|\mathbf{x}\|_2 = \|\mathbf{y}\|_2 = 1$. We deduce that

$$\|\mathbf{A}\|_2 = \sqrt{\lambda_1} = \sqrt{A^T A \text{ 的最大特征值}}.$$

□

c. $\infty$ **-norm:** Note that

$$\begin{aligned}\|\mathbf{A}\|_\infty &= \sup_{\|\mathbf{x}\|_\infty=1}\|\mathbf{A}\mathbf{x}\|_\infty \\ &\leq \sup_{\|\mathbf{x}\|_\infty=1}\left[\max_i\left|\sum_{j=1}^n a_{ij}x_j\right|\right] \\ &\leq \max_i\sum_{j=1}^n |a_{ij}|\end{aligned}$$

as $|x_j|\leq 1$ for $\|\mathbf{x}\|_\infty=1$. When $i^*=\arg\max_i|\sum_{j=1}^n a_{ij}|$ and $x_j=\operatorname{sign}a_{i^*j}$, we have

$$\max_i\left|\sum_{j=1}^n a_{ij}x_j\right| = \max_i\sum_{j=1}^n|a_{ij}|.$$

Thus,

$$\|A\|_\infty = \max_{1\leq i\leq n}\sum_{j=1}^n|a_{ij}|\,.$$

$\square$

Reference: Theory and applications of numerical analysis[M]. Elsevier, 1996. Chapter 10: MATRIX NORMS AND APPLICATIONS.

# 6 Cholesky 分解求解广义特征值问题

**Li 5.29** 设 $A$ 为 $n$ 阶实对称方阵，$B$ 为 $n$ 阶正定阵。写出用 Cholesky 分解的方法，求解广义特征值问题

$$A\alpha=\lambda B\alpha \tag{1}$$

的算法，并用编写 R 程序实现该算法。

**解.** 公式 (1) 等价于 $B^{-1}A\boldsymbol{\alpha}=\lambda\boldsymbol{\alpha}$。设 $B$ 有 Cholesky 分解

$$B=LL^T,$$

则由

$$A\boldsymbol{\alpha}=\lambda LL^T\boldsymbol{\alpha}$$

得

$$L^{-1}A\left(L^T\right)^{-1}\left(L^T\boldsymbol{\alpha}\right)=\lambda\left(L^T\boldsymbol{\alpha}\right),$$

求解普通特征值问题

$$\left(L^{-1}A\left(L^T\right)^{-1}\right)\boldsymbol{\beta}=\lambda\boldsymbol{\beta}$$

得 $\lambda$ 和 $\boldsymbol{\beta}$。再求解

$$L^T\boldsymbol{\alpha}=\boldsymbol{\beta}$$

即可得广义特征值和广义特征向量。

R 代码如下

```
# Cholesky 分解求解广义特征值问题
A = ...
B = ...
L = t(chol(B)) # R输出答案是L的转置
C = solve(L) %*% A %*% solve(t(L))
eig = eigen(C)
beta = eig$vectors
alpha = solve(t(L)) %*% beta
lambda = eig$values
```

# 7 R Code

## 7.1 Metropolis-Hastings Algorithm

### 7.1.1 Generate Mixture Normal Data

```
generate_normal_data = function(n, delta, miu, sigma) {
    # generate observed data from mixture normal distribution
    n_component = length(delta)
    n_sample_each_component = round(n * delta)
    # make sure sum equals n
    n_sample_each_component[-1] = n_sample_each_component[-1] + n
    - sum(n_sample_each_component)
    x = NULL
    for (i in 1:n_component) {
        xi = rnorm(n_sample_each_component[i], miu[i], sigma[i])
        x = c(x, xi)
    }
    x
}
# generate data ----
set.seed(0)
n = 200
delta = c(.7, .3)
miu = c(7, 10)
sigma = c(.5, .5)
# hist of generate data
par(mfrow = c(1, 1))
y = generate_normal_data(n, delta, miu, sigma)
hist(
y,
breaks = 30,
freq = FALSE,
main = "Histogram of Mixture Data",
ylab = "Density"
```

```
)
density_y = seq(5, 14, by = .01)
points(density_y,
    .7 * dnorm(density_y, 7, .5) + .3 * dnorm(density_y, 10, .5),
    type = "l")
```

### 7.1.2 Indepedence Chain MCMC Procedure

```
# general settings
set.seed(0)
MAX_ITER = 10000
burnin = 100
L = function(delta, y) {
    prod(delta * dnorm(y, 7, 0.5) + (1 - delta) * dnorm(y, 10, 0.5))
}

# Indepedence Chain MCMC Procedure ----
IndepedenceChain = function(y, R, init.value, max_iter = MAX_ITER) {
    est_delta = vector(length = max_iter + 1)
    est_delta[1] = init.value
    for (i in 1:max_iter) {
    xt = est_delta[i] # value in current iter
    x = runif(1, 0, 1) # value generate by proposal distribution
    r = min(R(xt, x, y, L), 1) # MH rate
    d = rbinom(1, 1, r)
    est_delta[i + 1] = x * d + xt * (1 - d)
    }
    est_delta
}
R.ic = function(xt, x, y, L) {
    L(x, y) / L(xt, y)
}
est_delta = IndepedenceChain(y, R.ic, .4)
indepchain.mean = mean(est_delta[(burnin + 1):(max_iter + 1)])
par(mfrow = c(2, 1))
plot(est_delta,
      ylim = c(.4, 1),
      type = "l",
      ylab = "delta")
abline(h = indepchain.mean, lwd = 2, col = "red")
text(8000, 0.8, paste0("mean=",
    round(indepchain.mean, 4)), col = "red")
title("Sample path for Indepedence Chain")
hist(est_delta[(burnin + 1):(max_iter + 1)],
      breaks = 20,
```

```
        xlab = "delta",
        main = "Hist. for Indepedence Chain")
```

#### 7.1.3 Random Walk MCMC Procedure

```
set.seed(0)
# in delta space ----
R.rw.delta = function(xt, x, y, L) {
    L(x, y) / L(xt, y)
}
RW.Delta = function(y, R, init.value, max_iter = MAX_ITER) {
    est_delta = vector(length = max_iter + 1)
    est_delta[1] = init.value
    for (i in 1:max_iter) {
    xt = est_delta[i] # value in current iter
    u = logit(xt) + runif(1, -1, 1)
    x = logit.inv(u) # value generate by random walk
    r = min(R(xt, x, y, L), 1) # MH rate
    d = rbinom(1, 1, r)
    est_delta[i + 1] = x * d + xt * (1 - d)
    }
    est_delta
}
est_delta = RW.Delta(y, R.rw.delta, .4)
rwchain.delta.mean = mean(est_delta[(burnin + 1):(max_iter + 1)])
par(mfrow = c(2, 1))
plot(est_delta,
        ylim = c(.4, .9),
        type = "l",
        ylab = "delta")
abline(h = rwchain.delta.mean, lwd = 2, col = "red")
text(8000,.8,paste0("mean=",round(rwchain.delta.mean,4)), col="red")
title("Sample path for Random Walk in delta Space")

# in U space ----
set.seed(0)
J_u = function(u) {
    exp(u) / (1 + exp(u)) ^ 2
}
logit = function(x) {
    log(x / (1 - x))
}
logit.inv = function(x) {
    exp(x) / (1 + exp(x))
}
```

```
R.rw.u = function(xt, x, y, L, J_u) {
    L(logit.inv(x), y) * abs(J_u(x)) /
        L(logit.inv(xt), y) / abs(J_u(xt))
}
RW.U = function(y, R, init.value, max_iter = MAX_ITER) {
    est_delta = vector(length = max_iter + 1)
    est_delta[1] = init.value
    for (i in 1:max_iter) {
    xt = est_delta[i] # value in current iter
    xt_u = logit(xt)
    x_u = xt_u + runif(1, -1, 1)# value generate by random walk
    r = min(R(xt_u, x_u, y, L, J_u), 1) # MH rate
    d = rbinom(1, 1, r)
    est_delta[i + 1] = logit.inv(x_u) * d + xt * (1 - d)
    }
    est_delta
}
est_delta = RW.U(y, R.rw.u, .4)
rwchain.u.mean = mean(est_delta[(burnin + 1):(max_iter + 1)])
plot(est_delta,
        ylim = c(.4, .9),
        type = "l",
        ylab = "delta")
abline(h = rwchain.u.mean, lwd = 2, col = "red")
text(8000, 0.8, paste0("mean=",
    round(rwchain.u.mean, 4)), col = "red")
title("Sample path for Random Walk in U Space")
```

### 7.2 Gibbs Sampler: Poisson Process with Change Point

```
set.seed(0)
MAX_ITER = 800
library(readr)
data = read_table2("coal.dat")
x = data$disasters
n = 112
burnin = 200
# initial
theta.init = 20
alpha.init = 5
lambda.init = 2
theta = vector(length = MAX_ITER + 1)
alpha = vector(length = MAX_ITER + 1)
lambda1 = vector(length = MAX_ITER + 1)
lambda2 = vector(length = MAX_ITER + 1)
```

```r
theta[1] = theta.init
alpha[1] = alpha.init
lambda1[1] = lambda.init
lambda2[1] = lambda.init

# gibbs sampler
for (i in 1:MAX_ITER) {
  lambda1[i + 1] = rgamma(1, sum(x[1:theta[i]]) + 3,
                          rate  = alpha[i + 1] + theta[i])
  lambda2[i + 1] = rgamma(1, sum(x[(1 + theta[i]):n]) + 3,
                          rate = n + alpha[i + 1] - theta[i])
  alpha[i + 1] = rgamma(1, 16,
                        rate =  10 + lambda1[i + 1] + lambda2[i + 1])
  ptheta = exp((lambda2[i + 1] - lambda1[i + 1])
               * (1:n)) * (lambda1[i +1] / lambda2[i + 1]) ^ cumsum(x)
  ptheta = ptheta / sum(ptheta) # 归一化
  theta[i + 1] = min((1:n)[runif(1) < cumsum(ptheta)])
}

# path
par(mfrow = c(2, 1))
plot(lambda1, type = "l")
plot(lambda2, type = "l")
plot(alpha, type = "l")
plot(theta, type = "l")

#cusum
l1.est = mean(lambda1[burnin:(MAX_ITER + 1)])
l2.est = mean(lambda2[burnin:(MAX_ITER + 1)])
a.est = mean(alpha[burnin:(MAX_ITER + 1)])
t.est = mean(theta[burnin:(MAX_ITER + 1)])
tmp = 1:(MAX_ITER - burnin + 1)
par(mfrow = c(2, 1))
plot(cumsum(lambda1[burnin:MAX_ITER] - l1.est) / tmp,
     ylab = "cusum lambda1",
     type = "l")
plot(cumsum(lambda2[burnin:MAX_ITER] - l2.est) / tmp,
     ylab = "cusum lambda2",
     type = "l")
plot(cumsum(alpha[burnin:MAX_ITER] - a.est) / tmp,
            ylab = "cusum alpha", type = "l")
plot(cumsum(theta[burnin:MAX_ITER] - t.est) / tmp,
            ylab = "cusum theta", type = "l")

# autocorrelation plot
```

```r
acf.max = 50
acf(lambda1, lag.max = acf.max)
acf(lambda2, lag.max = acf.max)
acf(theta, lag.max = acf.max)
acf(alpha, lag.max = acf.max)
ccf(lambda1, lambda2, lag.max = acf.max)
ccf(lambda1, theta, lag.max = acf.max)
ccf(lambda1, alpha, lag.max = acf.max)
ccf(lambda2, theta, lag.max = acf.max)
ccf(lambda2, alpha, lag.max = acf.max)
ccf(theta, alpha, lag.max = acf.max)

# hist density
l1d = density(lambda1[-(1:burnin)])
l2d = density(lambda2[-(1:burnin)])
ad = density(alpha[-(1:burnin)])
td = density(theta[-(1:burnin)])
par(mfrow = c(2, 1))
hist(
  lambda1[-(1:burnin)],
  breaks = 40,
  freq = FALSE,
  main = "Density Histogram of lambda1",
  ylab = "Density",
  xlab = "lambda1"
)
points(l1d$x, l1d$y, type = "l")
hist(
  lambda2[-(1:burnin)],
  breaks = 40,
  freq = FALSE,
  main = "Density Histogram of lambda2",
  ylab = "Density",
  xlab = "lambda2"
)
points(l2d$x, l2d$y, type = "l")
hist(
  alpha[-(1:burnin)],
  breaks = 20,
  freq = FALSE,
  main = "Density Histogram of alpha",
  ylab = "Density",
  xlab = "alpha"
)
points(ad$x, ad$y, type = "l")
```

```r
hist(
  theta[-(1:burnin)],
  breaks = 20,
  freq = FALSE,
  main = "Density Histogram of theta",
  ylab = "Density",
  xlab = "theta"
)
points(td$x, td$y, type = "l")
#summary stat
summary(lambda1[-(1:burnin)])
summary(lambda2[-(1:burnin)])
summary(alpha[-(1:burnin)])
summary(theta[-(1:burnin)])
```