

Problem 3

Bayesian Estimation in Logistic Regression

The Binary Regression Model

The logistic regression model is a model for binary outcome data formed using predictor values. Suppose that, for

$i = 1, \dots, n$,

$$Y_i \sim \text{Bernoulli}(\mu_i(\beta))$$

are an independent sample of binary outcomes, where

$$\mu_i(\beta) = \mathbb{E}[Y_i]$$

and β is a $(p \times 1)$ parameter vector. The log-likelihood is then

$$\begin{aligned}\ell(\beta) &= \sum_{i=1}^n [y_i \log \mu_i(\beta) + (1 - y_i) \log(1 - \mu_i(\beta))] \\ &= \sum_{i=1}^n \left[y_i \log \left(\frac{\mu_i(\beta)}{1 - \mu_i(\beta)} \right) + \log(1 - \mu_i(\beta)) \right]\end{aligned}$$

The Binary Regression Model

If

$$V(\mu_i(\beta)) = \mu_i(\beta)(1 - \mu_i(\beta))$$

is the variance of Y_i , and,

$$\dot{\mu}_i(\beta) \qquad \ddot{\mu}_i(\beta)$$

are a $(p \times 1)$ vector and a $(p \times p)$ matrix respectively, then

$$\begin{aligned} \ddot{\ell}(\beta) = & \sum_{i=1}^n \left[\frac{(y_i - \mu_i(\beta))}{V(\mu_i(\beta))} \ddot{\mu}_i(\beta) \right. \\ & \left. - \left(\frac{(y_i - \mu_i(\beta))(1 - 2\mu_i(\beta)) + V(\mu_i(\beta))}{\{V(\mu_i(\beta))\}^2} \right) \dot{\mu}_i(\beta) \dot{\mu}_i(\beta)^\top \right] \end{aligned}$$

The Binary Regression Model

We use link function l to capture the influence of predictors $\mathbf{x}_i$ on response Y_i . We write

$$l(\mu_i(\beta)) = \mathbf{x}_i\beta$$

for $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})$. Hence

$$\mu_i(\beta) = l^{-1}(\mathbf{x}_i\beta) = h(\mathbf{x}_i\beta)$$

say, and

$$\dot{\mu}_i(\beta) = \dot{h}(\mathbf{x}_i\beta)\mathbf{x}_i^\top \quad (p \times 1)$$

$$\ddot{\mu}_i(\beta) = \ddot{h}(\mathbf{x}_i\beta)\mathbf{x}_i^\top \mathbf{x}_i \quad (p \times p)$$

Here, $h(\cdot)$ is regarded as a function of a **scalar** argument, so $\dot{h}(\cdot)$ and $\ddot{h}(\cdot)$ are also scalar quantities

$$\dot{h}(t) = \frac{dh(t)}{dt} \qquad \ddot{h}(t) = \frac{d^2h(t)}{dt^2}.$$

The Binary Regression Model

In the logistic regression model,

$$h(t) = \frac{e^t}{1 + e^t}$$

we have

$$\dot{h}(t) = h(t)(1 - h(t))$$

$$\ddot{h}(t) = \dot{h}(t)(1 - 2h(t))$$

The Binary Regression Model: Bayesian analysis

For a Bayesian analysis of this model we perform the usual calculation for the posterior distribution $\pi(\beta)$

$$\pi(\beta) \propto \exp\{\ell(\beta)\}\pi_0(\beta)$$

where π_0 is the prior distribution for β .

Note: choosing

$$\pi_0(\beta) \propto 1$$

leads to an improper posterior (that is, $\pi(\beta)$ is not a proper distribution: it is not an integrable function of β).

For illustration, we may choose

$$\pi_0(\beta) \equiv \text{Normal}(0, \Sigma_0)$$

for some prior covariance matrix Σ_0 .

The Binary Regression Model: Bayesian analysis

We may use MCMC to produce a sample from $\pi(\beta)$. Here

$$\beta = (\beta_1, \dots, \beta_p)^\top$$

and we will assume that in the covariate vector

$$\mathbf{x}_i = (x_{i1}, \dots, x_{ip})$$

we have $x_{i1} \equiv 1$ to yield an intercept term.

We investigate three alternative algorithms:

1. Metropolis algorithm on β ;
2. Independence Metropolis-Hastings on β ;
3. Metropolis-within-Gibbs sampler using elements $\beta_j, j = 1, \dots, p$.

The Metropolis-Hastings Algorithm

Recall that to generate a Markov chain with stationary distribution π , we

- ▶ initialize $\beta^{(0)}$
- ▶ at iteration r , choose a proposal density $q(.,.)$ to generate candidate value β^* based on current point $\beta^{(r)}$. That is, we choose

$$q(u, v) = q(v|u)$$

and then sample β^* from $q(.|\beta^{(r)})$.

- ▶ **accept** β^* as the next value, and set $\beta^{(r+1)} = \beta^*$ with probability

$$\alpha(\beta^{(r)}, \beta^*) = \min \left\{ 1, \frac{\pi(\beta^*)}{\pi(\beta^{(r)})} \frac{q(\beta^{(r)}, \beta^*)}{q(\beta^*, \beta^{(r)})} \right\}$$

otherwise, **reject** β^* and set $\beta^{(r+1)} = \beta^{(r)}$.

Algorithms

1. Metropolis algorithm on β ;

- ▶ choose $q(u, v) = q(v, u)$
- ▶ acceptance probability reduces to

$$\alpha(\beta^{(r)}, \beta^*) = \min \left\{ 1, \frac{\pi(\beta^*)}{\pi(\beta^{(r)})} \right\}$$

- ▶ we choose

$$q(u, v) \equiv \text{Normal}(u, \sigma_q^2 \mathbf{I})$$

for some $\sigma_q > 0$. Results shown below are for $\sigma_q = 0.6528$.

Algorithms (cont.)

2. Independence Metropolis-Hastings on β ;

- we choose

$$q(u, v) \equiv q(v) \equiv \text{Normal}(\hat{\beta}, V_n(\hat{\beta}))$$

where $\hat{\beta}$ is the MLE, and

$$V_n(\hat{\beta}) = \left(-\ddot{\ell}(\hat{\beta}) \right)^{-1}$$

Algorithms (cont.)

3. Metropolis-within-Gibbs sampler using elements

$\beta_j, j = 1, \dots, p.$

- ▶ we consider the univariate full conditional distributions

$$\pi_j(\beta_j | \beta_{(-j)}) \propto \pi(\beta_1, \dots, \beta_j, \dots, \beta_p)$$

and use cyclic updating: for $j = 1, \dots, p$, to generate $\beta^{(r+1)}$, we use a Metropolis step on the densities

$$\pi_j(\cdot | \beta_1^{(r+1)}, \dots, \beta_{j-1}^{(r+1)}, \beta_{j+1}^{(r)}, \dots, \beta_p^{(r)})$$

for $j = 1, \dots, p.$

- ▶ choose $q(u, v) = q(v, u)$ componentwise; we use a Normal proposal

$$q(\cdot | \beta_j) \equiv \text{Normal}(\beta_j, \sigma_{qj}^2)$$

with σ_{qj} taken to be 0.3570, 0.6604, 0.2228, 1.0447 for $j = 1, 2, 3, 4.$

Simulation Settings

```
set.seed(3122)
library(MASS)
n<-200
p<-3
Sig<-rWishart(1,p,diag(1,p))[, , 1]
X<-cbind(rep(1,n),mvrnorm(n,mu=rep(0,p),Sigma=Sig))
p<-ncol(X)

be<-c(-2,1,0.5,-2.5)

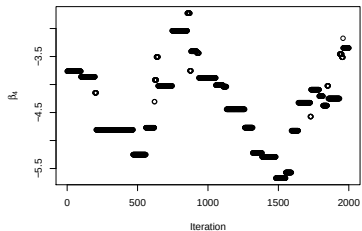
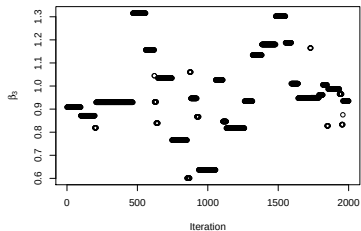
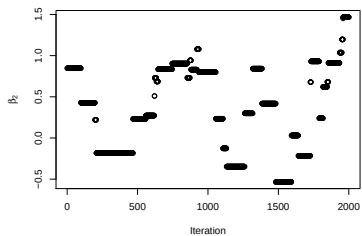
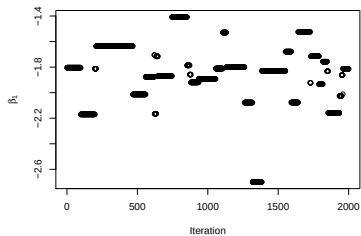
muvec<-1/(1+exp(-X %*%be))

Y<- rbinom(n,1,muvec)

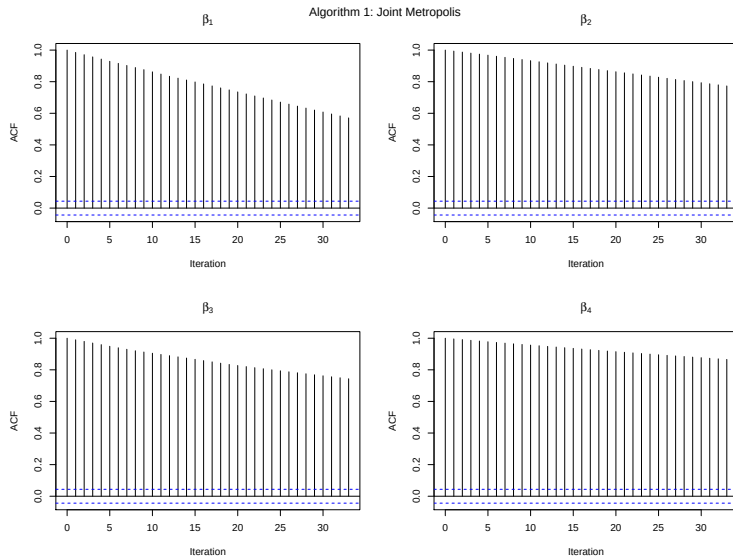
#Prior variance
prior.sd<-10;prior.var<-prior.sd^2
```

Algorithm 1: trace plots

Algorithm 1: Joint Metropolis

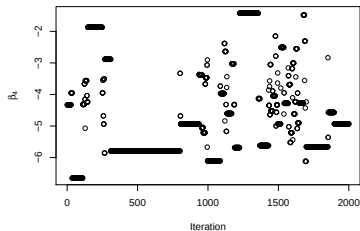
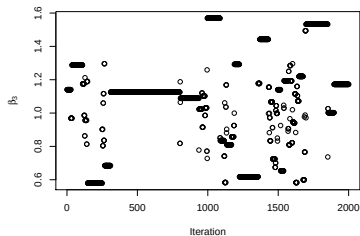
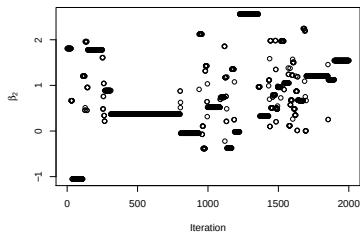
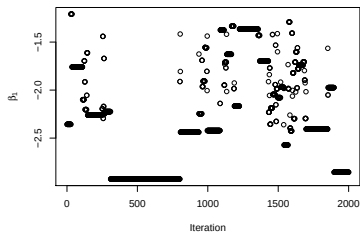


Algorithm 1: autocorrelation

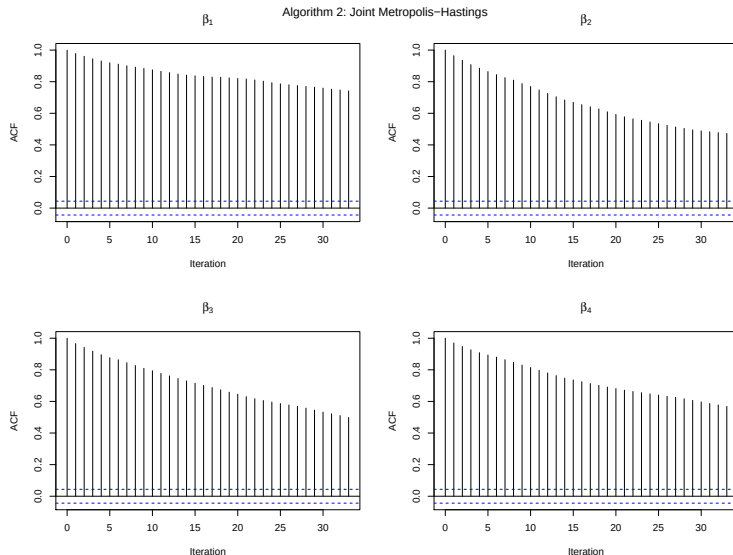


Algorithm 2: trace plots

Algorithm 2: Joint Metropolis-Hastings

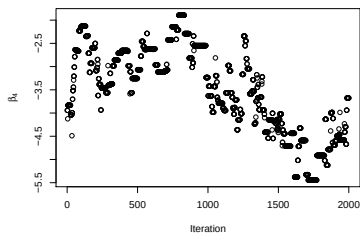
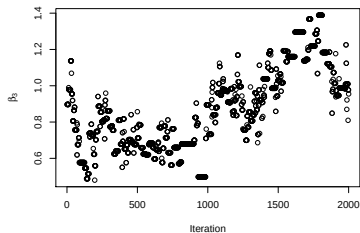
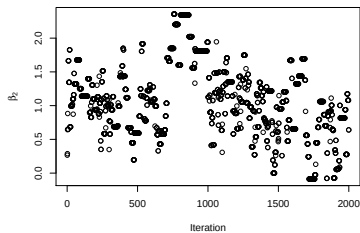
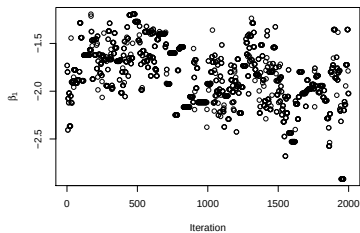


Algorithm 2: autocorrelation

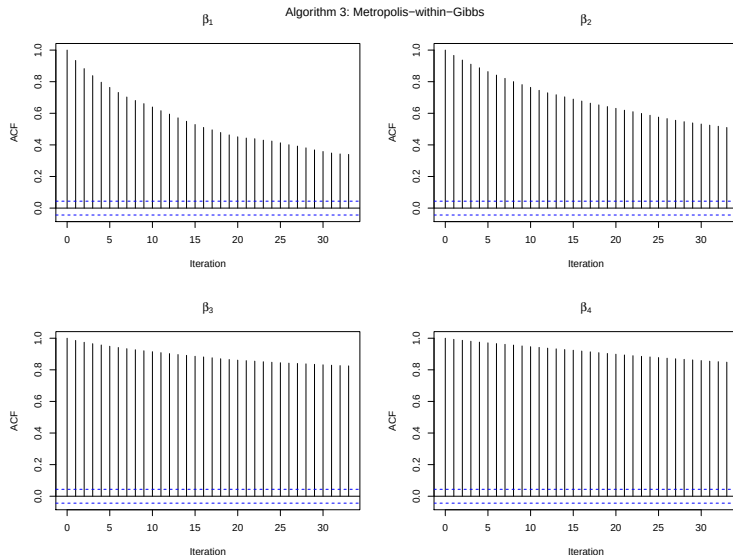


Algorithm 3: trace plots

Algorithm 3: Metropolis-within-Gibbs



Algorithm 3: autocorrelation



Comparison using the Effective Sample Size

```
> library(coda)
> effectiveSize(beta.samp1)
      var1      var2      var3      var4
14.860154  6.377238 10.072063  4.400050
> effectiveSize(beta.samp2)
      var1      var2      var3      var4
 8.854457 26.688114 21.187473 18.429503
> effectiveSize(beta.samp3)
      var1      var2      var3      var4
46.670453 26.420358  9.157468  4.910500
```

Algorithms 2 and 3 perform a little better than Algorithm 1.

Improving the Algorithms

- ▶ could adjust the Metropolis proposal variances σ_q^2
- ▶ could combine joint Metropolis-Hastings proposals with Gibbs sampler steps in a single algorithm
 - ▶ could randomly choose to make a joint or a Gibbs sampler update
 - ▶ could alternate choices at successive iterations.

Improving the Algorithms

- ▶ could use a better Independence Metropolis-Hastings proposal
 - ▶ standard MLE theory suggests

$$\exp\{\ell(\beta)\} \simeq \exp\left\{-\frac{1}{2}(\beta - \hat{\beta})^\top \left(-\ddot{\ell}(\hat{\beta})\right) (\beta - \hat{\beta})\right\}$$

- ▶ standard Bayesian calculation reveals that

$$\pi(\beta) \simeq \text{Normal}\left(\tilde{\mathbf{v}}_n, \tilde{V}_n\right)$$

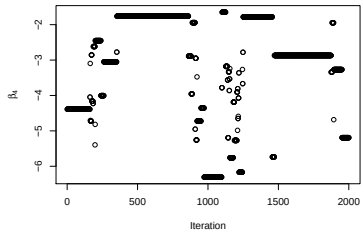
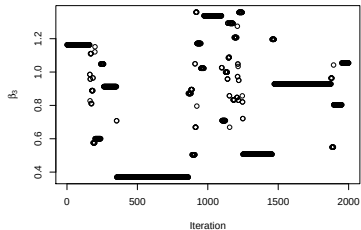
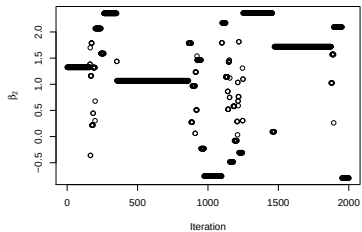
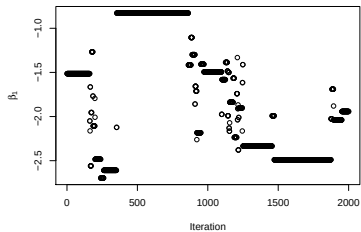
where

$$\begin{aligned}\tilde{\mathbf{v}}_n &= \left(-\ddot{\ell}(\hat{\beta}) + \Sigma_0^{-1}\right) \hat{\beta} \\ \tilde{V}_n &= \left(-\ddot{\ell}(\hat{\beta}) + \Sigma_0^{-1}\right)^{-1}\end{aligned}$$

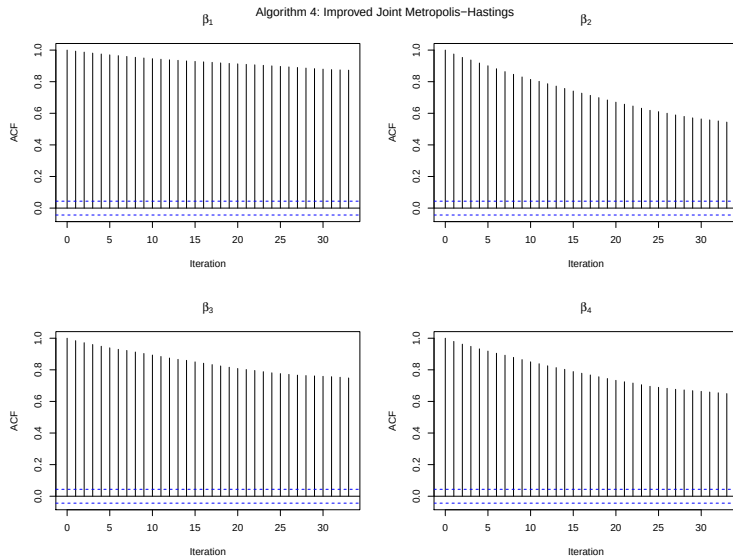
- ▶ little difference from Algorithm 2 when n is large

Algorithm 4: trace plots

Algorithm 4: Improved Joint Metropolis-Hastings



Algorithm 4: autocorrelation



Improving the Algorithms

- ▶ could reparameterize $\beta \longrightarrow \psi$ so that the ψ are less correlated in the posterior.
 - ▶ Normal approximation to posterior suggests that a linear reparameterization may be sufficient, that is

$$\psi = M\beta$$

for some $p \times p$ non-singular matrix M .

- ▶ note that if $\eta_i = \mathbf{x}_i\beta$, then

$$\eta = \mathbf{X}\beta = \mathbf{X}M^{-1}\psi = \mathbf{X}^*\psi$$

say, so in the ψ parameterization we have an identical GLM but defined with **transformed** predictors $\mathbf{x}_i^* = \mathbf{x}_iM^{-1}$.

- ▶ as

$$-\ddot{\ell}(\widehat{\beta}) \simeq \text{constant} \cdot \mathbf{X}^\top \mathbf{X}$$

choose M such that

$$\mathbf{X}^{*\top} \mathbf{X}^* = M^{-\top} \mathbf{X}^\top \mathbf{X} M^{-1}$$

is diagonal (we need to orthogonalize the design matrix).

Improving the Algorithms

Choosing M using the **Cholesky decomposition** of $\mathbf{X}^\top \mathbf{X}$ suffices: we define upper triangular matrix U by

$$\mathbf{X}^\top \mathbf{X} = U^\top U$$

so that

$$M^{-\top} \mathbf{X}^\top \mathbf{X} M^{-1} = M^{-\top} U^\top U M^{-1}$$

and then choose

$$M = U \qquad M^{-1} = U^{-1}$$

which results in the required orthogonalization

$$\mathbf{X}^* = \mathbf{X} U^{-1}$$

Improving the Algorithms

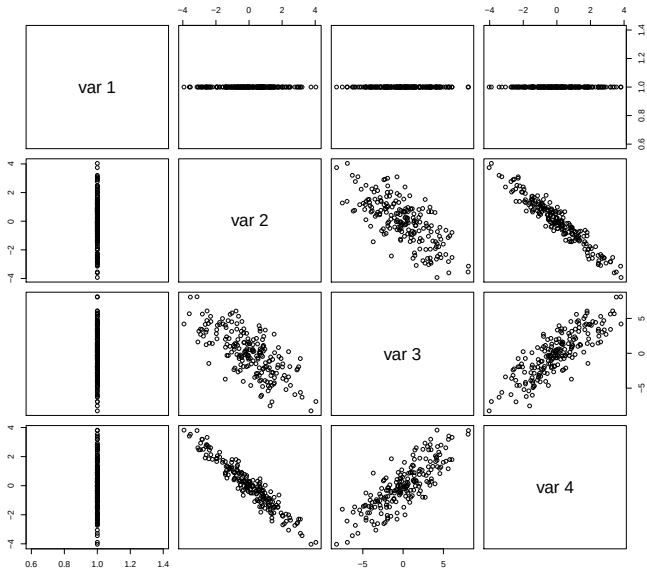
```
> V<-t(X) %*% X
> (round(V,6))
```

	[,1]	[,2]	[,3]	[,4]
[1,]	200.000000	1.464983	3.374561	3.187453
[2,]	1.464983	476.787858	-690.319392	-463.562858
[3,]	3.374561	-690.319392	1999.171086	815.570132
[4,]	3.187453	-463.562858	815.570132	489.511297

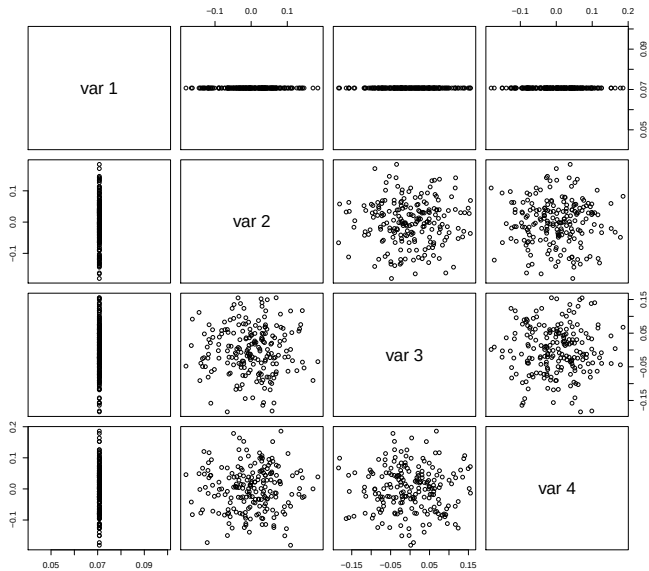
```
> U<-chol(V)
> Xstar<-X %*% solve(U)
> round(t(Xstar) %*% Xstar,6)
```

	[,1]	[,2]	[,3]	[,4]
[1,]	1	0	0	0
[2,]	0	1	0	0
[3,]	0	0	1	0
[4,]	0	0	0	1

Original X



Transformed $\mathbf{X}^*$



Prior in new parameterization

Note that if

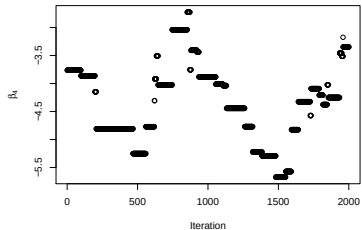
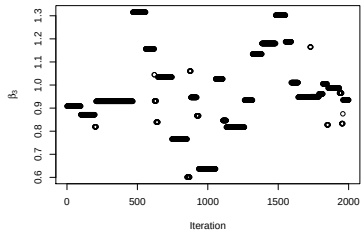
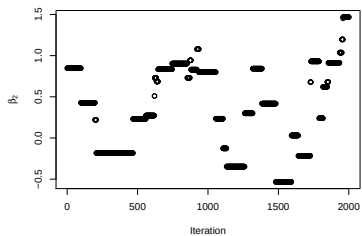
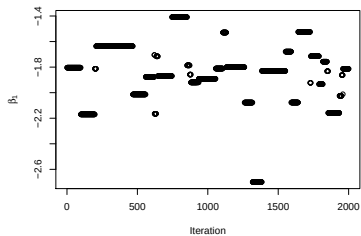
$$\pi_0(\beta) \equiv \text{Normal}(0, \Sigma_0)$$

then the equivalent prior in the new parameterization is

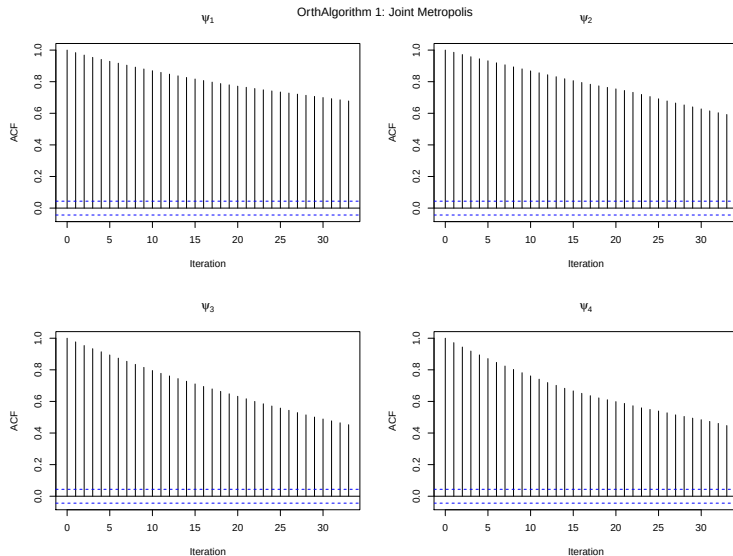
$$\text{Normal}(0, M\Sigma_0M^\top)$$

Algorithm 1: trace plots

Algorithm 1: Joint Metropolis

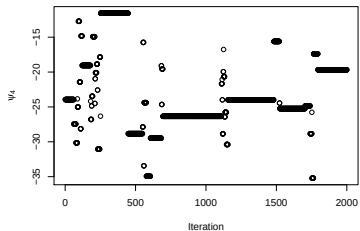
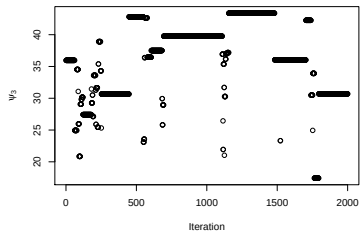
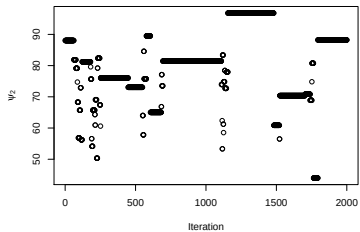
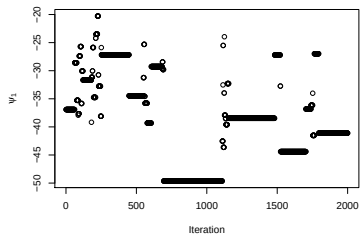


Algorithm 1: autocorrelation

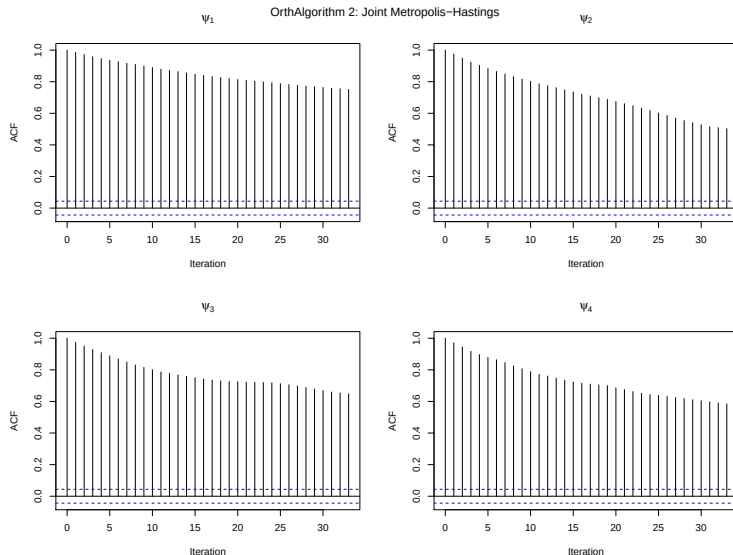


Algorithm 2: trace plots

OrthAlgorithm 2: Joint Metropolis-Hastings

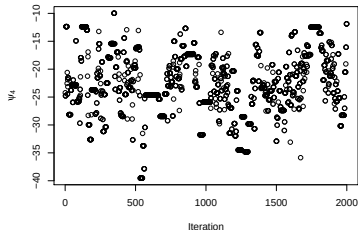
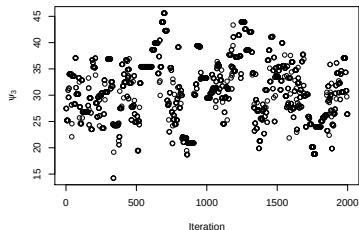
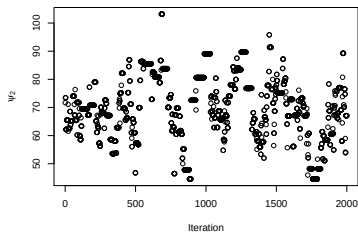
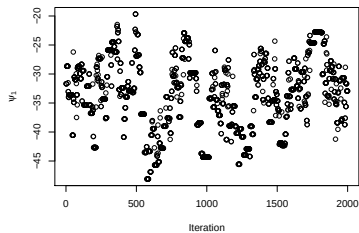


Algorithm 2: autocorrelation

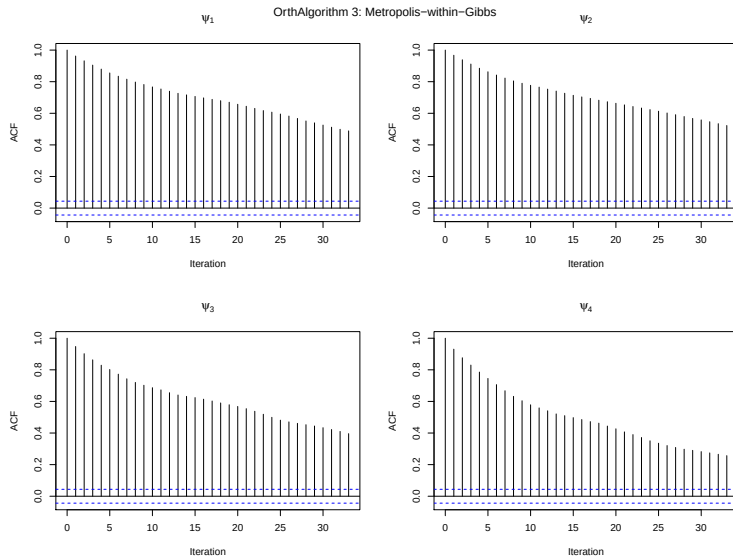


Algorithm 3: trace plots

OrthAlgorithm 3: Metropolis-within-Gibbs

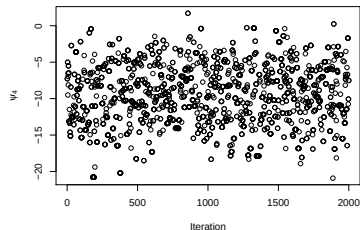
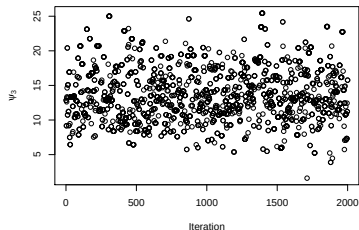
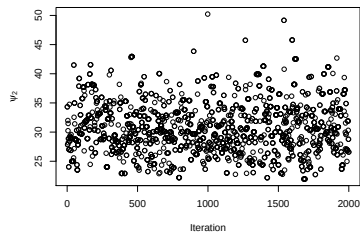
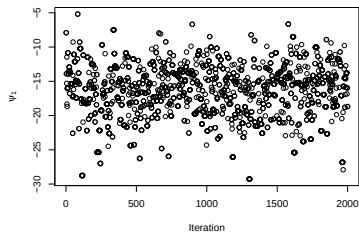


Algorithm 3: autocorrelation



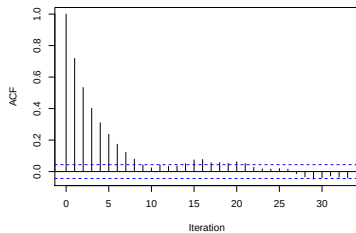
Algorithm 4: trace plots

OrthAlgorithm 4: Improved Joint Metropolis–Hastings

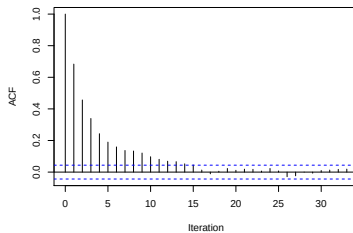


Algorithm 4: autocorrelation

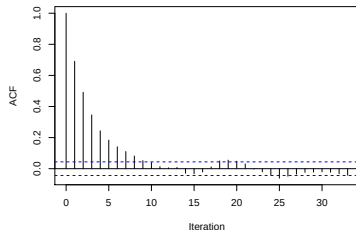
Ψ_1 OrthAlgorithm 4: Improved Joint Metropolis-Hastings



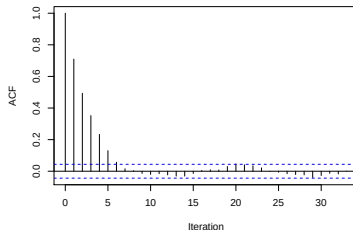
Ψ_2



Ψ_3



Ψ_4



Comparison using the Effective Sample Size

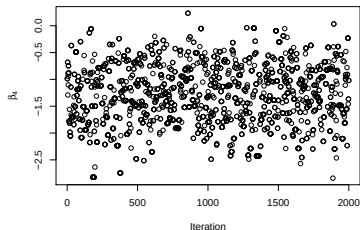
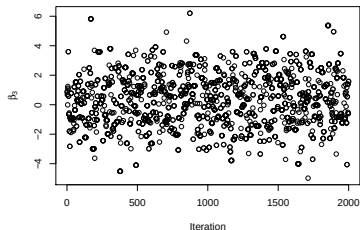
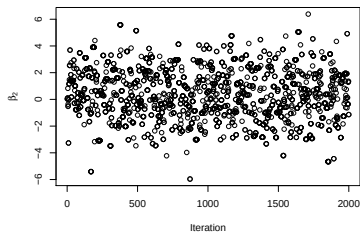
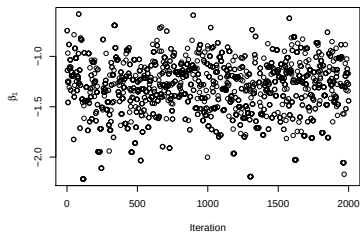
```
> effectiveSize(orth.beta.samp1)
      var1      var2      var3      var4
17.719751  9.688564 26.907225 21.299532
> effectiveSize(orth.beta.samp2)
      var1      var2      var3      var4
 6.789724 37.145172 11.286162 22.284763
> effectiveSize(orth.beta.samp3)
      var1      var2      var3      var4
47.21871 47.92092 70.75486 59.06712
> effectiveSize(orth.beta.samp4)
      var1      var2      var3      var4
357.1096 343.1607 307.6202 413.6570
```

All algorithms perform better than in the non-orthogonal parameterization. We can obtain samples of β by back-transforming the samples for ψ , that is

$$\beta = M^{-1}\psi.$$

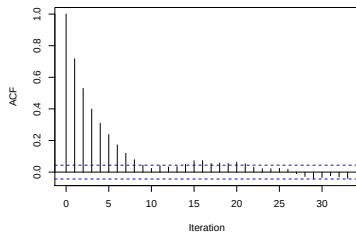
Algorithm 4: trace plots for β

OrthAlgorithm 4: Improved Joint Metropolis–Hastings

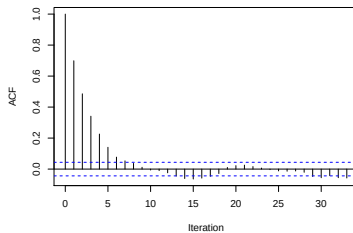


Algorithm 4: autocorrelation for β

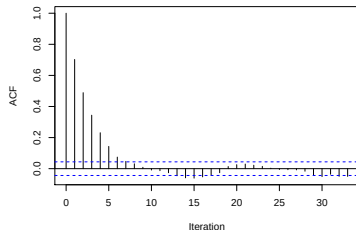
β_1 OrthAlgorithm 4: Improved Joint Metropolis-Hastings



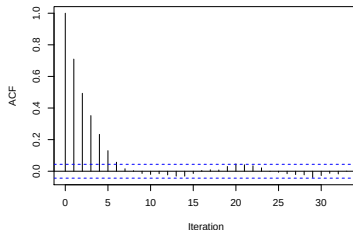
β_2



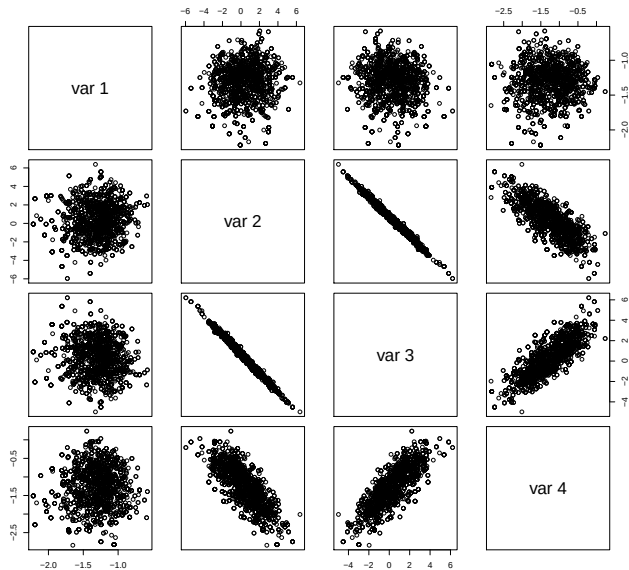
β_3



β_4



Algorithm 4: posterior scatter plots for β



Algorithm 4: posterior scatter plots for ψ

