

Exercise 1: Let the observed data $\{y_i\}_{i=1}^n$ be binary with each y_i being an independent realization of a Bernoulli random variable Y_i with parameter $\mu_i = \Pr(Y_i = 1)$. Each $\mu_i = \mathbb{E}[Y_i]$ is modeled with a logistic function, that is,

$$\mu_i = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \Leftrightarrow \log \left\{ \frac{\mu_i}{1 - \mu_i} \right\} = \beta_0 + \beta_1 x_i,$$

where $\mathbf{x} = [x_1, \dots, x_n]^T$ is a vector of predictor values and β_0 and β_1 are parameters.

1. Write down the likelihood $p(\mathbf{y} | \beta_0, \beta_1)$ for the logistic regression model.
2. Find the posterior $p(\beta_0, \beta_1 | \mathbf{y})$ assuming independent Normal priors with $\mu_{\beta_j} = 0$ and variance $\sigma_{\beta_j}^2$, where $j = 1, 2$.
3. Demonstrate that marginalizing \mathbf{u} out of the joint posterior

$$p(\mathbf{u}, \beta_0, \beta_1 | \mathbf{y}) \propto \prod_{i=1}^n 1 \left(u_i < \frac{e^{\beta_0 y_i + \beta_1 x_i y_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \right) \exp \left\{ -\frac{(\beta_0 - \mu_{\beta_0})^2}{2\sigma_{\beta_0}^2} - \frac{(\beta_1 - \mu_{\beta_1})^2}{2\sigma_{\beta_1}^2} \right\}$$

yields the posterior $p(\beta_0, \beta_1 | \mathbf{y})$.

4. Implement slice sampling by generating

$$\begin{aligned} u_i | \mathbf{u}_{-i}, \beta_0, \beta_1, \mathbf{y} &\sim \text{Unif} \left[0, \frac{e^{\beta_0 y_i + \beta_1 x_i y_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \right] \\ \beta_0 | \mathbf{u}, \beta_1, \mathbf{y} &\sim \text{Normal}(\beta_0 | \mu_{\beta_0}, \sigma_{\beta_0}) \prod_{i=1}^n 1 \left(u_i < \frac{e^{\beta_0 y_i + \beta_1 x_i y_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \right) \\ \beta_1 | \mathbf{u}, \beta_0, \mathbf{y} &\sim \text{Normal}(\beta_1 | \mu_{\beta_1}, \sigma_{\beta_1}) \prod_{i=1}^n 1 \left(u_i < \frac{e^{\beta_0 y_i + \beta_1 x_i y_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \right) \end{aligned}$$

Note that the full conditionals $p(\beta_0 | \mathbf{u}, \beta_1, \mathbf{y})$ and $p(\beta_1 | \mathbf{u}, \beta_0, \mathbf{y})$ are truncated Normal distributions.

5. Apply the slice sampler to the following simulated data:

```
n <- 100
beta0 <- 2 ; beta1 <- 0.5
set.seed( 100 ) ; x <- abs( rnorm( n ) )
eta <- beta0 + beta1 * x ; mu <- exp( eta ) / ( 1 + exp( eta ) )
y <- rbinom( n, 1, mu )
```

Solution: The likelihood is

$$p(\mathbf{y} | \beta_0, \beta_1) = \prod_{i=1}^n \left(\frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \right)^{y_i} \left(1 - \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \right)^{1 - y_i} = \prod_{i=1}^n \frac{e^{\beta_0 y_i + \beta_1 x_i y_i}}{1 + e^{\beta_0 + \beta_1 x_i}}.$$

The posterior assuming independent Normal priors is

$$p(\beta_0, \beta_1 | \mathbf{y}) \propto \left[\prod_{i=1}^n \frac{e^{\beta_0 y_i + \beta_1 x_i y_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \right] \exp \left\{ -\frac{\beta_0^2}{2\sigma_{\beta_0}} - \frac{\beta_1^2}{2\sigma_{\beta_1}} \right\}.$$

Marginalizing \mathbf{u} out of the joint posterior $p(\mathbf{u}, \beta_0, \beta_1 | \mathbf{y})$ yields the original posterior density:

$$\begin{aligned} p(\beta_0, \beta_1 | \mathbf{y}) &= \int_0^{z_1} \cdots \int_0^{z_n} p(\mathbf{u}, \beta_0, \beta_1 | \mathbf{y}) d\mathbf{u} \quad \text{where } z_i = \frac{e^{\beta_0 y_i + \beta_1 x_i y_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \\ &= \left[\prod_{i=1}^n \int_0^{z_i} du_i \right] \exp \left\{ -\frac{\beta_0^2}{2\sigma_{\beta_0}} - \frac{\beta_1^2}{2\sigma_{\beta_1}} \right\} \\ &= \left[\prod_{i=1}^n \frac{e^{\beta_0 y_i + \beta_1 x_i y_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \right] \exp \left\{ -\frac{\beta_0^2}{2\sigma_{\beta_0}} - \frac{\beta_1^2}{2\sigma_{\beta_1}} \right\}. \end{aligned}$$

The truncation points of the full conditionals $p(\beta_0 | \mathbf{u}, \beta_1, \mathbf{y})$ and $p(\beta_1 | \mathbf{u}, \beta_0, \mathbf{y})$ are found by noting that

$$\begin{aligned} u_i < \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} &\Rightarrow \log \left\{ \frac{u_i}{1 - u_i} \right\} < \beta_0 + \beta_1 x_i & \text{if } y_i = 1 \\ u_i < \frac{1}{1 + e^{\beta_0 + \beta_1 x_i}} &\Rightarrow \log \left\{ \frac{1 - u_i}{u_i} \right\} < \beta_0 + \beta_1 x_i & \text{if } y_i = 0 \end{aligned}$$

so that

$$\begin{aligned} \beta_0 &\in \left(\max_{y_i=1} \left\{ \log \frac{u_i}{1 - u_i} - \beta_1 x_i \right\}, \min_{y_i=0} \left\{ \log \frac{1 - u_i}{u_i} - \beta_1 x_i \right\} \right) \\ \beta_1 &\in \left(\max_{y_i=1} \left\{ x_i^{-1} \left[\log \frac{u_i}{1 - u_i} - \beta_0 \right] \right\}, \min_{y_i=0} \left\{ x_i^{-1} \left[\log \frac{1 - u_i}{u_i} - \beta_0 \right] \right\} \right). \end{aligned}$$

An implementation of the slice sampler in R is:

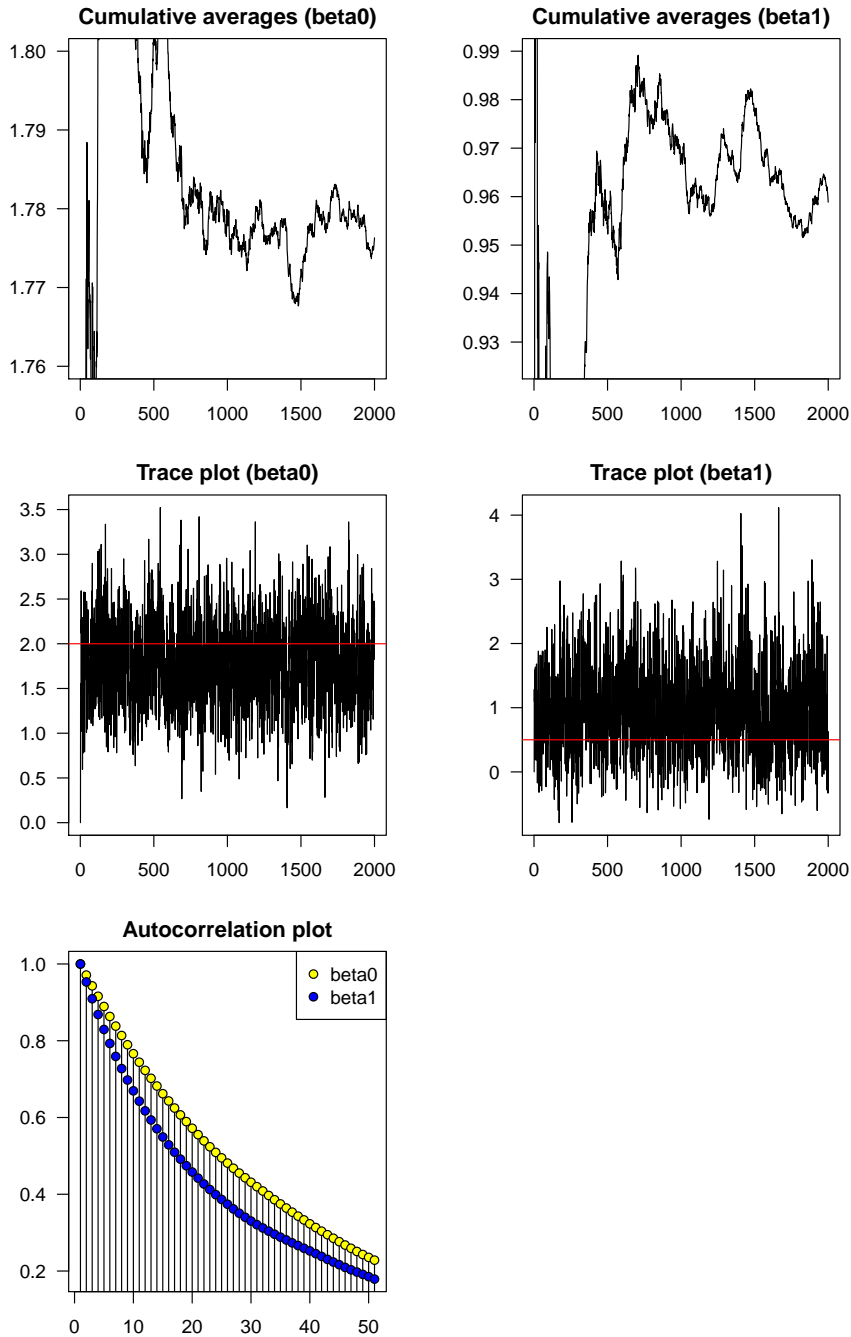
```
nSamples <- 100000 ; sigmaBeta <- 100
yOnes <- y == 1 ; yZeros <- !yOnes
result <- matrix( 0, nSamples, 2 )
for( ii in seq( 2, nSamples ) ) {
  eta <- result[ ii - 1, 1 ] + result[ ii - 1, 2 ] * x
  u <- runif( n, rep( 0, n ), exp( y * eta ) / ( 1 + exp( eta ) ) )
  logit <- log( u / ( 1 - u ) )
  aBeta <- max( logit[ yOnes ] - result[ ii - 1, 2 ] * x[ yOnes ] )
  bBeta <- min( -logit[ yZeros ] - result[ ii - 1, 2 ] * x[ yZeros ] )
  aF <- pnorm( aBeta, sd = sigmaBeta )
  bF <- pnorm( bBeta, sd = sigmaBeta )
  result[ ii, 1 ] <- qnorm( aF + runif( 1 ) * ( bF - aF ), sd = sigmaBeta )
  aBeta <- max( ( logit[ yOnes ] - result[ ii, 1 ] ) / x[ yOnes ] )
  bBeta <- min( ( -logit[ yZeros ] - result[ ii, 1 ] ) / x[ yZeros ] )
  aF <- pnorm( aBeta, sd = sigmaBeta )
}
```

```

bF <- pnorm( bBeta, sd = sigmaBeta )
result[ ii, 2 ] <- qnorm( aF + runif( 1 ) * ( bF - aF ), sd = sigmaBeta )
}

```

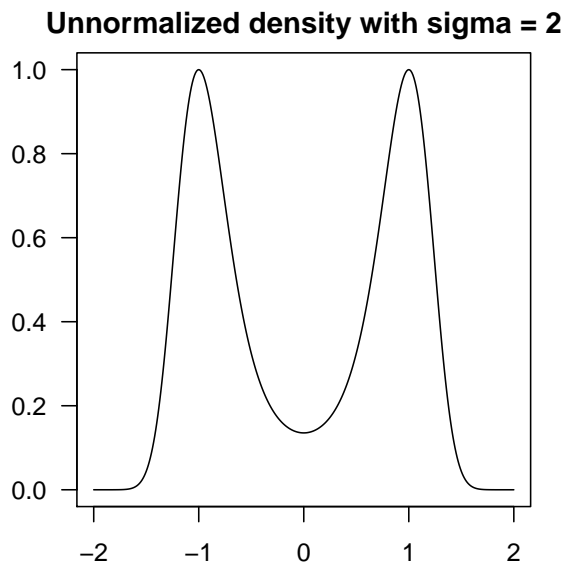
Generating 100 000 samples from the posterior using $\sigma_{\beta_0}^2 = \sigma_{\beta_1}^2 = 10000$ yields $\mathbb{E}[\beta_0 | \mathbf{y}] = 1.78$ and $\mathbb{E}[\beta_1 | \mathbf{y}] = 0.96$. Note the difference between the posterior mean and maximum likelihood estimate for β_1 . Also note that the slice sampler suffers from large autocorrelations and slow convergences as shown in the following plots.



Exercise 2: Let X be a random variable with unnormalized density

$$p(x|\sigma) \propto \exp\{-\sigma(x^2 - 1)^2\}.$$

The density of X is bimodal as can be seen from the following figure:



1. Implement a random walk Metropolis–Hastings sampler based on $\text{Normal}(0, \sigma^2)$ noise.
2. Implement a random walk Metropolis–Hastings sampler that runs 4 chains in parallel with different values of σ . That is, the first Markov Chain has the target $p_1(x|\sigma_1)$, the second Markov chain has the target $p_2(x|\sigma_2)$ and so forth. In each iteration, accept a swap between the states x_i and x_j of two randomly chosen chains i and j with probability

$$\alpha = \min\left\{1, \frac{p_i(x_j|\sigma_i)p_j(x_i|\sigma_j)}{p_i(x_i|\sigma_i)p_j(x_j|\sigma_j)}\right\}.$$

3. Run both methods for $\sigma = 1, 2, 4, 8$ and inspect the histograms for $\sigma = 8$ to see if both methods can approximate the bivariate distribution.

Solution: An implementation of a random walk Metropolis–Hastings sampler in R is:

```
target <- function( x, sigma ) { -sigma * ( x ^ 2 - 1 ) ^ 2 }
nSamples <- 25000 ; nAccepted <- 0 ; sigma <- 2
samples <- numeric( nSamples )
for( ii in seq( 2, nSamples ) ) {
  samples[ ii ] <- rnorm( 1, samples[ ii - 1 ], 1 )
  alpha <- target( samples[ ii ], sigma ) - target( samples[ ii - 1 ], sigma )
  if( runif( 1 ) > exp( alpha ) ) {
    samples[ ii ] <- samples[ ii - 1 ]
  }
}
```

```

    } else {
      nAccepted <- nAccepted + 1
    }
  }
}

```

An implementation of a parallel random walk Metropolis–Hastings sampler with swapping in R is:

```

target <- function( x, sigma ) { -sigma * ( x ^ 2 - 1 ) ^ 2 }
nSamples <- 25000 ; nChains <- 4 ; nAccepted <- numeric( nChains ) ; nSwaps <- 0
sigma <- c( 1, 2, 4, 8 )
samples <- matrix( 0, nSamples, nChains )
for( ii in seq( 2, nSamples ) ) {
  samples[ ii, ] <- rnorm( nChains, samples[ ii - 1, ], c( 2, 1, 0.5, 0.375 ) )
  alpha <- target( samples[ ii, ], sigma ) - target( samples[ ii - 1, ], sigma )
  rejectIndex <- runif( nChains ) > exp( alpha )
  nAccepted[ !rejectIndex ] <- nAccepted[ !rejectIndex ] + 1
  samples[ ii, rejectIndex ] <- samples[ ii - 1, rejectIndex ]
  swapIndex <- sample( seq_len( nChains ), 2 )
  alpha <- target( samples[ ii, swapIndex[ 2 ] ], sigma[ swapIndex[ 1 ] ] ) +
    target( samples[ ii, swapIndex[ 1 ] ], sigma[ swapIndex[ 2 ] ] ) -
    target( samples[ ii, swapIndex[ 1 ] ], sigma[ swapIndex[ 1 ] ] ) -
    target( samples[ ii, swapIndex[ 2 ] ], sigma[ swapIndex[ 2 ] ] )
  if( runif( 1 ) < exp( alpha ) ) {
    samples[ ii, swapIndex ] <- samples[ ii, rev( swapIndex ) ]
    nSwaps <- nSwaps + 1
  }
}
}

```

Generating 25 000 samples demonstrates that the simple random walk Metropolis–Hasting sampler does not adequately explore the entire distribution, although a larger number of samples improves the exploration of the distribution. In contrast, the parallel random walk Metropolis–Hastings sampler is able to easily jump between the modes and approximate the bivariate distribution. This is due to the fact that the chains with lower values of σ jump between the modes easily and occasionally the chain with $\sigma = 8$ exchanges states with those.

