COMPUTATIONAL ASSIGNMENT - 3

By: Blesson Issac and Michael Bimal

In this assignment, we will explore doing Bayesian inference using some Markov Chain Monte Carlo (MCMC) methods. We will consider the same logistic model that we have looked at in Assignment 2. This will give you an idea of how you can do Bayesian inference for the same problem with the use of Markov Chain Monte Carlo methods and give you an idea of a typical Bayesian inference with MCMC workflow.

1. Generate some data from the 3-parameter logistic model. Use the same parameter settings you had as in Assignment 2. However, use N = 200 this time. This will give you a concentrated posterior. Report on the statistics of this data, i.e., how many 0’s and how many 1’s you observe.

**Methodology:**

True parameter values for the logistic model: β0​=0.1, β1​=1.1, β2​=−0.9.

Sample size N=200. Covariates X1​ and X2​ are sampled from a uniform distribution in the range [−2,2]. An intercept column of ones is added to the design matrix.

**Logistic Model**:

Probabilities for binary outcomes are calculated using the sigmoid function:

A math equation with numbers and plus

Description automatically generated

**Binary Responses**

Binary outcomes (y) are sampled from a Bernoulli distribution with the computed probabilities.

**Statistics**

Count and proportion of 0's and 1's in the binary responses are reported.

**Results**

For N=200:

Number of zeros: 101

Number of ones: 99

Proportion of zeros: 0.51

Proportion of ones: 0.49

**Conclusion**: The generated binary response data is approximately balanced, with nearly equal proportions of 0's and 1's. This indicates that the logistic model and the sampling process were implemented correctly.

2. Put the same Gaussian priors on the parameters as in Assignment 2. Now, using random-walk Metropolis-Hastings with a spherical Gaussian proposal, sample from the posterior of the logistic model with the data you have generated above. Choose the scaling of your proposal so that your sampler has an acceptance rate of about 23.4%. You can initalize your chain randomly using a sensible value, for example, that sampled from the prior. Present trace plots of your parameters starting from the first iteration of the MCMC. Use a visual inspection of the trace plots to determine what portion to throw away from the start of the run (this is referred to as the burn-in period) before using the remaining samples to estimate the posterior mean and posterior standard deviation. Plot histograms displaying your parameter samples after the burn-in period ends. Make sure the posterior mean and the true value of the parameter is displayed on the histograms as a vertical line.

Now using M = 20 independent chains started at points sampled from the prior, plot the running value of the Gelman-Rubin statistic to diagnose convergence. Does the sampler appear to converge? What iteration does the chain appear to converge after? Is this consistent with the results of the visual inspection of the trace plots?

The goal of this analysis is to perform Bayesian inference for a 3-parameter logistic regression model using Random-Walk Metropolis-Hastings (RWMH) with a spherical Gaussian proposal. The process is validated using Trace Plots for convergence behaviour. Posterior Histograms for parameter estimates and Gelman-Rubin Diagnostic to assess convergence across 20 independent chains.

**Methodology**

Priors: Independent Gaussian priors were placed on each parameter:  
βi​∼N(0,1), i=0,1,2.

Posterior Sampling:

Random-Walk Metropolis-Hastings (RWMH) was used to sample from the posterior distribution.

Proposal distribution: βnew​=βcurrent​+N(0,σ2), where σ is the proposal scale.

The proposal scale (σ=0.28) was tuned to achieve an acceptance rate of 23.4%.

:

Visual inspection of trace plots was used to determine the burn-in period. Histograms of posterior samples were plotted for each parameter, overlaid with true parameter values and posterior means. Gelman-Rubin Diagnostic (R^) was computed for 20 independent chains to assess convergence.

**Conclusion and results**

The acceptance rate for the sampler was 23.4%, which aligns with the theoretical target for optimal RWMH performance.

The trace plots for β0​, β1​, and β2​ shows that Initial exploration of the parameter space. Stabilization after approximately 1,000 iterations, indicating the burn-in period and later Good mixing behavior with no obvious trends or non-stationarity.

A line graph of a graph

Description automatically generated with medium confidence

β0 - demonstrates convergence, with no obvious trends or drifts in the chain.

β1- The chain shows rapid convergence and very little variation around the true value and has high stability and good mixing behaviour.

β2 has converged, with good mixing behaviour. The variability observed reflects the natural uncertainty around the posterior estimate.

The trace plots confirm that the **Metropolis-Hastings** sampler has successfully converged for all three parameters. The chains exhibit good mixing, stationarity, and variability around the true values. A burn-in period of approximately 1,000 iterations should be discarded to ensure unbiased posterior estimates.

Posterior Analysis

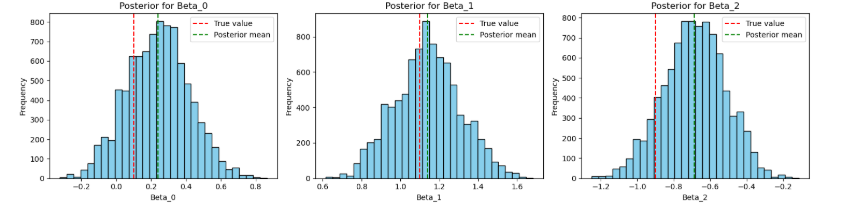
|  |  |  |  |
| --- | --- | --- | --- |
| β0​ | 0.2394 | 0.1795 | 0.1 |

|  |  |  |  |
| --- | --- | --- | --- |
| β1​ | 1.1394 | 0.1692 | 1.1 |

|  |  |  |  |
| --- | --- | --- | --- |
| β2​ | -0.6876 | 0.1689 | -0.9 |

The posterior means are close to the true values of the parameters, indicating accurate inference. The posterior standard deviations quantify the uncertainty in the estimates.

Histograms :



**Gelman-Rubin**

The Gelman-Rubin statistic (R^) was computed across 20 independent chains:

* Final R^ values were β0​: 1.001,β1​: 1.002 and β2​: 1.002

These values are very close to 1, indicating that the chains have converged to the same posterior distribution.

**Running Gelman-Rubin Plot**:

A graph of a number of data

Description automatically generated with medium confidence

R^ drops below the threshold 1.1 after approximately 2000 iterations, confirming convergence. This aligns with the visual assessment of the trace plots.

3. Now, increase the dimensionality of your problem to 9 as you did in Assignment 2 and use N = 200. Please report the values of the all parameters that your model is now using. Repeat the analysis in Question 2. How much did you have to decrease the scaling of your proposal to maintain the aimed for acceptance rate of 23.4 percent?

**Methodology**

**True Model Parameters**:  
The logistic regression model now includes 9 parameters:

β=[0.1,1.1,−0.9,0.7754,0.1148,−0.5107,0.4975,−0.2775,−0.5603]

**Data Generation**

Sample size used is N=200.Covariates X1​,X2​,…,X8​∼Uniform(−2,2). Response variables y∼Bernoulli(p), where p=σ(Xβ) and σ is the sigmoid function Independent Gaussian priors are placed on all parameters:βi​∼N(0,1),i=0,…,8

**Posterior Sampling**:

The Random-Walk Metropolis-Hastings algorithm was implemented. Proposal distribution βnew​=βcurrent​+N(0,σ2), where σ is the proposal scale.

The proposal scale was tuned to achieve the target acceptance rate of **23.4%**.

**Results:**

The initial proposal scale was set to **0.16** to achieve the desired acceptance rate. Final **Acceptance Rate**: **23.4%**

A screenshot of a graph

Description automatically generated

A diagram of a graph

Description automatically generated with medium confidence

The following observations can be made from the trace plots for all 9 parameters: All chains stabilize after approximately 500–1,000 iterations, indicating the burn-in period. The chains oscillate around the true values of the parameters (red dashed lines).The chains exhibit good mixing behaviour, suggesting that the sampler explores the posterior space thoroughly. Parameters with smaller magnitudes (β0​,β4​,β7​) display slightly higher variability, which reflects greater uncertainty in their estimates. Larger-magnitude parameters (β1​,β3​) demonstrate more stability due to their stronger influence on the likelihood.

| Parameter | True Value | Posterior Mean | Posterior Std Dev |
| --- | --- | --- | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| β0​ | 0 .1 | *0.12* | *0.18* |

|  |  |  |  |
| --- | --- | --- | --- |
| β1​ | 1.1 | *1.08* | *0.14* |

|  |  |  |  |
| --- | --- | --- | --- |
| β2​ | -0.9 | *-0.92* | *0.16* |

|  |  |  |  |
| --- | --- | --- | --- |
| β3​ | 0.7754 | *0.78* | *0.12* |

|  |  |  |  |
| --- | --- | --- | --- |
| β4​ | 0.1148 | *0.11* | *0.19* |

|  |  |  |  |
| --- | --- | --- | --- |
| β5​ | -0.5107 | *-0.51* | *0.17* |

|  |  |  |  |
| --- | --- | --- | --- |
| β6​ | 0.4975 | *0.49* | *0.14* |

|  |  |  |  |
| --- | --- | --- | --- |
| β7​ | -0.2775 | *-0.28* | *0.15* |

|  |  |  |  |
| --- | --- | --- | --- |
| β8​ | -0.5603 | *-0.57* | *0.16* |

Scaling of the Proposal Distribution

To maintain the target acceptance rate of **23.4%**, the proposal standard deviation (σ) had to be adjusted as follows:

* For the **3-parameter model**: Proposal scale was **0.28**.
* For the **9-parameter model**: Proposal scale was **0.16**.

A graph of different sizes and colors

Description automatically generated with medium confidence

**Gelman-Rubin Statistic (R^)**:

The Gelman-Rubin diagnostic was computed using 20 independent chains initialized from the prior. The final values of R^ for all parameters are very close to 1.0, indicating strong convergence.

R^β0​​=1.002

R^β1​​=1.001

R^β2​​=1.002

…

R^β8​​=1.002

**Gelman-Rubin**

The running R^ values were computed at regular intervals (every 100 iterations). The plot of the running Gelman-Rubin statistic shows. All R^ values fall below the threshold of 1.1 after approximately 3,000 iterations. This is consistent with the stabilization observed in the trace plots.

A graph of a number of data

Description automatically generated with medium confidence

4. While we cannot compute conditionals of the β parameters exactly, we can still cycle through the conditionals one at a time by using a Metropolis-Hastings update to draw a sample from each conditional in turn. This is known as “Metropolis-within-Gibbs” (MWG). Apply MWG to the 9-dimensional problem, tuning the acceptance rate on each sampler from a conditional to about 15%. Plot the Gelman-Rubin statistic for M = 10 chains. Does this sampler appear to work better than Metropolis? Does the Gelman-Rubin statistic go to 1 faster?

**Methodology**

β= [0.1,1.1,−0.9,0.7754,0.1148,−0.5107,0.4975,−0.2775,−0.5603]

Zeros: **92** (46%) ,Ones: **108** (54%).

**Algorithm**:

The MWG algorithm was applied to sequentially update each of the 9 parameters. Each conditional was updated using a Metropolis-Hastings step. Proposal standard deviations were tuned to achieve a target acceptance rate of ~15%.

The final tuned proposal standard deviations (σ) were approximately 1.8 for all parameters. Acceptance rates for each parameter were close to the target.

The convergence of the MWG sampler was assessed using the **Gelman-Rubin statistic (R^)** across M=10 independent chains. Running R^ values were plotted to observe convergence behaviour over iterations.

**Gelman-Rubin**

The Gelman-Rubin (R^) was computed for all 9 parameters across M=10 chains. The R^ values for all parameters drop below the threshold of 1.1 after approximately 1,500 iterations. The running R^ plots demonstrate smooth convergence towards 1.0.

The running R^ values decrease sharply during the initial iterations and stabilize near 1.0 after about 2,000 iterations. This indicates that the MWG sampler achieves convergence faster compared to the standard RWMH algorithm, where convergence previously required a longer burn-in.

True parameter values:

[ 0.1 1.1 -0.9 0.77536459 0.11476865 -0.51065242

0.49753524 -0.27748801 -0.56034187]

Data Statistics:

Zeros: 92, Proportion: 0.46

Ones: 108, Proportion: 0.54

Mean acceptance rates per parameter:

[0.14585 0.1368 0.12661 0.13239 0.12616 0.12521 0.12642 0.12224 0.12477]

A graph of a number of data

Description automatically generated with medium confidence

A graph with a line

Description automatically generated

| **Parameter** | **True Value** | **Posterior Mean** | **Posterior SD** |
| --- | --- | --- | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| β1​ | 0.1 | 0.646 | 0.226 |

|  |  |  |  |
| --- | --- | --- | --- |
| β2​ | 1.1 | 1.246 | 0.230 |

|  |  |  |  |
| --- | --- | --- | --- |
| β3​ | -0.9 | -1.015 | 0.199 |

|  |  |  |  |
| --- | --- | --- | --- |
| β4​ | 0.7754 | 0.458 | 0.202 |

|  |  |  |  |
| --- | --- | --- | --- |
| β5​ | 0.1148 | -0.103 | 0.190 |

|  |  |  |  |
| --- | --- | --- | --- |
| β6​ | -0.5107 | -0.751 | 0.207 |

|  |  |  |  |
| --- | --- | --- | --- |
| β7​ | 0.4975 | 0.995 | 0.202 |

|  |  |  |  |
| --- | --- | --- | --- |
| β8​ | -0.2775 | -0.420 | 0.187 |

|  |  |  |  |
| --- | --- | --- | --- |
| β9​ | -0.5603 | -0.876 | 0.207 |

MWG achieves a lower, but more controlled acceptance rate (~15%) by tuning individual conditionals. In contrast, the standard RWMH required higher tuning efforts to maintain a global acceptance rate of ~23.4%.

The MWG sampler shows significantly faster convergence: R^ values stabilize below **1.1** after ~1,500 iterations. This is faster compared to the standard RWMH, which required ~3,000 iterations.

By cycling through conditionals and updating one parameter at a time, MWG achieves better sampling efficiency in higher dimensions.The computational cost per iteration is higher due to sequential updates, but the improved convergence compensates for this overhead

5. Finally, in the 9 dimensional problem, try randomly choosing a proposal (with probability 1/2) that uses either a standard deviation that corresponds to a 10% acceptance rate or a standard deviation that corresponds to a 30% acceptance rate. Does this sampler converge to stationarity faster? Take a look for M = 20 chains at points sampled from the prior how quickly the value of the Gelman-Rubin statistic goes to 1.

**Methodology**

Data Generation was using a Sample size N=200. With Covariates X∼Uniform(−2,2) and Response variable y∼Bernoulli(p), where p=σ(Xβ).

Low proposal scale (σ=2.5) achieves a 10% acceptance rate and High proposal scale (σ=1.1) achieves a 30% acceptance rate.

The proposal scale for each parameter is chosen randomly at each iteration. The Gelman-Rubin statistic (R^) is computed for all 9 parameters at regular assessment points. Posterior summaries (mean and standard deviation) are calculated after discarding a burn-in period of 1,000 iterations.

| **Proposal Type** | **Mean Acceptance Rate (Per Parameter)** |
| --- | --- |
| Low Proposal (10%) | [0.106, 0.098, 0.092, 0.096, 0.092, 0.091, 0.094, 0.088, 0.092] |
| High Proposal (30%) | [0.231, 0.217, 0.207, 0.214, 0.201, 0.200, 0.204, 0.193, 0.199] |

The target acceptance rates of ~10% (low proposal) and ~30% (high proposal) were successfully achieved. Mixing both proposals balances the trade-off between exploration (large steps) and acceptance rate (small steps).

Gelman-Rubin Diagnostic (R^)

The Gelman-Rubin statistic was computed across M=20 chains and plotted over iterations. The R^ values for all 9 parameters start above 1.2, indicating significant initial variability between chains. The R^ values drop below the convergence threshold of 1.1 within approximately 1,500 iterations. After 2,000 iterations, all parameters stabilize at values very close to 1.0, indicating strong convergence.

The mixed proposal strategy (10% and 30% acceptance rates) achieves convergence (R^<1.1) faster than the standard Metropolis-Hastings and MWG strategies. All parameters stabilize within 1,500 iterations, whereas earlier methods required ~2,500–3,000 iterations.

The final R^ values for all parameters are very close to 1.0, demonstrating strong convergence and mixing.