An Exploration of Binary Classification Models Using MCMC

Introduction to Relevant Methodology

Markov Chain Monte Carlo (MCMC) are a class of Bayesian algorithms first developed in Los Alamos National Laboratory in the early 1950s by a team led by physicist and mathematician Nicholas Metropolis [5]. These algorithms skyrocketed in popularity in the 1990s as computing power began to match the capabilities required to run these algorithms. The "Monte Carlo" aspect to MCMC is a broad term used to describe the Monte Carlo method, credited to John Von Neumann and Stanislaw Ulam [5], applying a deterministic "rule" to a sample from a random probability distribution in order to approximate an estimate of a target numerical result. This was then extended to MCMC, firstly (and the case which will be focused on in this report) in the form of the Metropolis-Hastings algorithm.

The Metropolis-Hastings algorithm works on the following logic:

- Define a candidate distribution q and a target distribution π.
- Define an initial state X_i.
  For $i \in \{i, \cdots, n-1\}$:
- Generate a proposed move $Y \sim Q(X_i, \cdot)$. In the random walk case: $Y \sim \mathcal{N}(X_i, hI_{d\times d})$
- Define $\alpha(X_i, Y) = \min\Big(1, \frac{\pi(Y)q(Y,X_i)}{\pi(X_i)q(X_i,Y)}\Big)$.
  Consider $(U \sim \mathcal{U}[0,1])$, if $U \leq \alpha(X_i, Y)$:
- $X_{i+1} = Y$
  Else:
- $X_{i+1} = X_i$

Task Outset

With this foundational knowledge of the Metropolis-Hastings algorithm, this knowledge can be applied to the task at hand.

Consider:

$$Y_i|\beta, x_i \overset{\text{iid}}{\sim} \text{Bernoulli}\left(F(x_i^T\beta)\right)$$

Where F is a symmetric cdf around 0.

Let's consider two possible forms of F, the standard logistic distribution (logistic regression) and the standard Cauchy distribution (Cauchit regression).

Let X be a $n\times 10$ matrix, where $\forall i, X_{i,1} = 1$ and for $j \in \{2, \cdots, 10\} X_{i,j}$ are a linear combination of standard normal variables.

Y can also be represented with its latent variable interpretation $Y_i = \mathbb{1}(Y^* > 0)$ where

$$Y^* = x_i^T\beta + \epsilon_i$$

With ϵ_i being randomly drawn from f.

n values of Y^{C} will be generated using this method, with $\epsilon_i \sim t_1$

Now considering the logistic case, rejection sampling can be applied on the latent ϵ_i space to draw n ϵ_i values from the logistic distribution.

Rejection Sampling

Rejection sampling works with a similar logic to the Metropolis-Hastings algorithm, with a candidate distribution $q(x)$ and a target distribution $\pi(x)$ required, and acceptance rates based on their ratios. The constant $1 \leq M < \infty$ and the specified number of samples n are also required. It works as follows: Draw an $X \sim q(\cdot)$ and a $U \sim \mathcal{U}[0,1]$ and if $U \leq \frac{\pi(x)}{Mq(x)}$ then X can be accepted as a sample. If not, try again with a new X and U. Repeating this until n samples are drawn allows direct sampling of n random variables from π, assuming that $\forall x\, \pi(x) \leq Mq(x)$. Therefore, an M value must be chosen such that it is large enough to satisfy this equality, but there is incentive to minimise this, as the expected value of iterations in order to achieve n samples is Mn, so from an efficiency standpoint a low M is desirable.

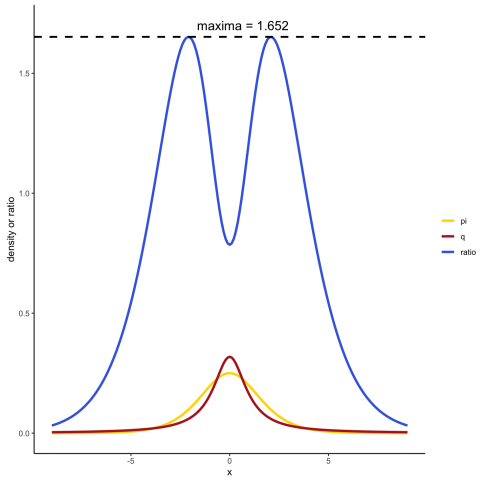

Figure 1: Finding the optimal value of M

An M can therefore be chosen such that $M = \inf\{M : \forall x, \pi(x)/q(x) \leq M\}$. So, take M to be the maxima of the ratio $\pi(x)/q(x)$. Let's consider the case where q is a standard Cauchy distribution and π is a standard logistic distribution. Since the Cauchy distribution is heavier tailed than the logistic distribution, as x deviates from 0 the ratio will tend to 0. The maxima of the ratio in this case is ≈ 1.652, which can be seen in Figure 1. With this optimal M value, the rejection sampling procedure can be performed to achieve n samples of Y^{L} which will be carried forwards into the next section.

Implementing an MCMC Algorithm - Preliminary Considerations

First assuming a prior $\beta_j \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$ for $j \in \{1, \cdots, 10\}$, a Markov Chain Monte Carlo method can be considered in order to estimate β. First, the posterior distribution must be attained, which is achieved by the following formula:

$$\pi(\beta \mid y) \propto \prod_{i=1}^{n} F(x_i^T\beta)^{y_i}(1 - F(x_i^T\beta))^{1-y_i} \prod_{j=1}^{10} \exp\left(-\frac{\beta_j^2}{2}\right)$$

$$\propto \exp\left(\sum_{i=1}^{n}\left[y_i \log\left(F(x_i^T\beta)\right) + (1-y_i)\log\left(1 - F(x_i^T\beta)\right)\right] - \frac{1}{2}\sum_{j=1}^{10}\beta_j^2\right)$$

The log-posterior can therefore be found by removing the exponent. Before considering an algorithm in order to estimate the true β values, a few characteristics of the data must be considered. First, the tails of the posterior can be analysed, as a heavier tailed target distribution may require different approaches for MCMC interpretation. The issue with these heavy tailed distributions, is if the chain is far out in the tails it may struggle to leave these regions, as the gradient of these regions are ≈ 1 (depending on how heavy the tails are, usually if $\pi(x) \to 0$ slower than $e^{-|x|} \to 0$ as $x \to \infty$) so the chain may exhibit random walk behaviour in the extreme regions of the distribution as $a \to 1$, not correctly exploring all regions of the distribution. Whilst the likelihood will be heavy

tailed in the Cauchy case, this won't be an issue, since the prior distribution that will be considered (both here and later with the UIP) will be Gaussian, the overall tails of the posterior distribution will be lighter than exponential.

The next consideration is that of correlation of the X columns. From looking at their generation in the R code, it appears that high correlation may occur. While computing the actual correlation would be very cumbersome, a reasonable approximation to this can be achieved using the sample correlation. Firstly, it can be noted that the first and fourth column of X are uncorrelated, as $X_1 = 1$ and X_4 is created solely from halving the third column of Z, where Z is a matrix of standard Gaussian variables. Since no other column in X uses any values from the third column of Z, X_4 is uncorrelated from the other X_j. The sample correlation can therefore be computed, with 0 replacing any correlations involving the first and fourth columns.

Table 1: Sample correlation matrix of columns of X

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
X_1	1	0	0	0	0	0	0	0	0	0
X_2	0	1	0.97	0	0.05	0.01	0.01	0.52	0.053	−0.029
X_3	0	0.97	1	0	0.04	0.02	0.02	0.51	0.04	−0.01
X_4	0	0	0	1	0	0	0	0	0	0
X_5	0	0.05	0.04	0	1	−0.08	−0.08	0.42	1	0.04
X_6	0	0.01	0.02	0	−0.08	1	1	0.02	−0.08	0.19
X_7	0	0.01	0.02	0	−0.08	1	1	0.02	−0.09	0.11
X_8	0	0.52	0.51	0	0.42	0.02	0.02	1	0.47	0
X_9	0	0.05	0.04	0	1	−0.08	−0.09	0.47	1	0.04
X_{10}	0	−0.03	−0.01	0	0.04	0.19	0.11	0	0.04	1

Correlation is most certainly present in Table 1, with plenty of significant correlation values outside of the diagonal. Since the data exhibits a fair amount of correlation, this must be considered in the choice of algorithm.

Implementing an MCMC Algorithm - Algorithm Choice

Given the preliminary findings, an algorithm must be considered which takes into account the correlation of the data.

Pre-conditioning is a concept which alters the Metropolis-Hastings algorithm to account for the covariance of the data. It extends the random walk case of the Metropolis-Hastings algorithm by altering the proposal Y from $Y \sim \mathcal{N}(X_i, hI_{d\times d})$ to $Y \sim \mathcal{N}(X_i, h\Sigma)$. The choice of Σ is rather important, with the ideal choice being $\Sigma = \text{Cov}_\pi[\beta \mid y]$. With a good choice of Σ, the algorithm should converge faster than it would in the random walk case [3], so it is important to consider a correct form. Adaptive MCMC seeks to alter Σ based on the chain's output as a "plug and play" solution to this problem, and this is a very good method for when the true covariance cannot be found, which in practical cases is almost all of the time. In more detail, this is done by running the chain with a proposed Σ, and re-adjusting this Σ by taking the sample covariance of the chain and applying an optimal scaling factor: $\frac{(2.38)^2}{d}$ found by Rosenthal [4]. In this case there is a differentiable posterior distribution, so optimisation techniques can be performed in order to achieve $\Sigma \approx \text{Cov}_\pi[\beta \mid y]$. This should give a very good starting point for adaptive MCMC to tune the covariance matrix towards the true covariance.

To find this initial value of Σ, the posterior distribution was optimised using a pre-built numerical algorithm to calculate the sample modes of β_j, and then the Hessian matrix of the distribution at the modes was calculated, and it was positive definite, making the next step possible. This positive definite attribute means two things essentially, firstly that the optimisation found a local maxima instead of a saddle point or minima, and it allows the random walk algorithm to run, as a non-symmetric covariance matrix would result from a non-positive definite Hessian, which is not possible for a valid covariance matrix. The negative inverted Hessian is then used as an approximation of the covariance matrix of the posterior distribution at the MAP [2].

Whether or not this initial value of Σ can be significantly improved upon using adaptive MCMC can be evaluated. By running three chains with a set Σ at the negative inverted Hessian, and one with a constantly adapting Σ, the PSRF value given by calculating the Gelman-Rubin statistic can give an indication of the speed of convergence to the target distribution. Taking this approach, PSRF values of ≤ 1.06 were found for both the adaptive and non-adaptive case (A value of less than 1.10 is typically seen as desirable [7]), with no clear favourite. This result very much favours the use of using the negative inverse Hessian as a simple pre-conditioner, as it seems that complicating the algorithm with the introduction of an adaptable Σ value is providing no tangible improvement to the convergence. Therefore, a pre-conditioned random walk Metropolis-Hastings algorithm will be that carried forwards to the diagnostic testing phase. To re-iterate, a proposal will be found with the following form: $\beta_{\text{prop}} = \beta_{\text{curr}} + h\mathcal{N}(1, \Sigma)$, where Σ is the negative inverted Hessian found. This model will also be applied to a different prior distribution, the *Unit Information Prior* (UIP) of the form: $\beta \sim \mathcal{N}_d\left(0, n(X^TX)^{-1}\right)$, and the posterior distribution when incorporating the UIP will become:

$$\pi(\beta|y) \propto \exp\left(\sum_{i=1}^{n}\left[y_i \log(F(x_i^T\beta)) + (1-y_i)\log(1-F(x_i^T\beta))\right] - \frac{1}{2}\beta^T(n(X^TX))\beta\right)$$

Diagnostic Testing

A good starting point for diagnostic testing is to view the mixing. The step size has already been tuned to match the acceptance rate of 23% found to be optimal in the dimension setting relevant to this problem [1, 6], so to view how this theoretically optimally scaled algorithm looks, Figure 2 provides some insight. The mixing looks very good, reminiscent of the "hairy caterpillar" that is often looked for, with no visible trends or drifting.

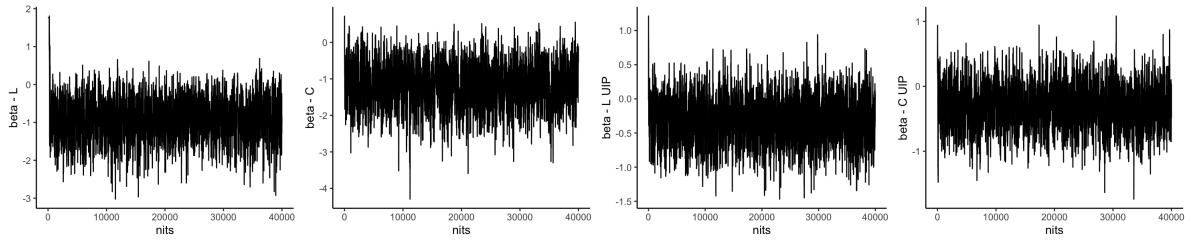

Figure 2: Mixing of β_1: Logit data, Cauchit data, logit data with UIP, Cauchit data with UIP

But how about the actual accuracy of the estimates to the real β values? This of course depends a lot on the sample of X values achieved with the relatively small n value of 150, but this is still a vital criteria to test the chosen algorithm on. By plotting the density of the MCMC estimates in Figure 3, there is a lot to dissect. Firstly, the MCMC estimates seem zero-centered, as when the β values are closer to $|2|$, the estimates seem somewhat worse. Secondly, the standard prior seems to

perform better than the UIP, which is an interesting observation that taking the sample covariance into account seems to make the estimate worse.

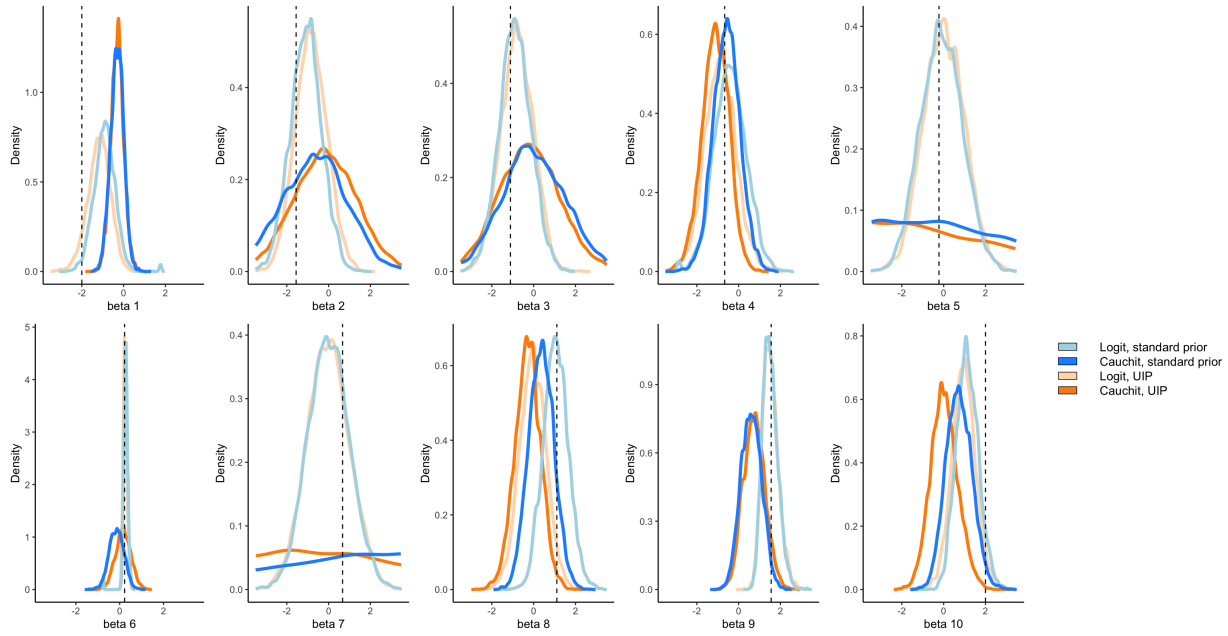

Figure 3: β estimates. Dashed line is true β_i value, blue is logit data, orange/pink is Cauchit, light is standard prior and dark is UIP

To better quantify this bias, the Brier score can be used in these four instances. The Brier score is of the form: $S(\mathcal{M}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - m_i)^2$, where $m_i = F(x_i^T\beta)$. A Brier score of 0 represents perfect accuracy, whilst a score of 1 represents perfect inaccuracy. The scores can be seen in Figure 4, and as expected from the density plots, the standard prior performs best. What wasn't so evident was the positive performance of the logit data which is interesting.

	Logit Data	Cauchit Data
Standard Prior	0.014	0.023
UIP	0.031	0.043

Figure 4: Brier scores

A final diagnostic to consider is the Gelman-Rubin test. This was previously discussed in the decision making for dropping adaptive MCMC, and is ubiquitous in MCMC analysis. The test essentially compares inter-chain variance with inter-chain variance and the statistic should tend to one as nits increase if the chain demonstrates good mixing and explores the parameter space. This convergence to one does indeed occur for all four chains, a positive sign for the consistency of the four chains.

Final Remarks

To conclude, this report has explored the binary classification model, using both logistic regression and Cauchit regression to create a pre-conditioned random walk Metropolis-Hastings algorithm to estimate β. The main two forms of quantifying the accuracy of the model is mixing and bias, with the model performing very well in mixing and less well in bias, however given the small nature of the original sample of X values this could either be sample-related or to do with an oversight in the approach chosen.

References

[1] Justin Ellis. A practical guide to mcmc part 1: Mcmc basics, 01 2018.

[2] J.-L. Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, 2:291–298, 04 1994.

[3] Max Hird and Samuel Livingstone. Quantifying the effectiveness of linear preconditioning in markov chain monte carlo. *arXiv*, 12 2023.

[4] Rosenthal Jeffery. Optimal proposal distributions and adaptive mcmc, handbook of markov chain monte carlo, 2008.

[5] Christian Robert and George Casella. A short history of markov chain monte carlo: Subjective recollections from incomplete data. *Statistical Science*, 26:102–115, 02 2011.

[6] Sebastian M. Schmon and Philippe Gagnon. Optimal scaling of random walk metropolis algorithms using bayesian large-sample asymptotics. *Statistics and Computing*, 32, 02 2022.

[7] Dootika Vats and Christina Knudson. Revisiting the gelman–rubin diagnostic. *Statistical Science*, 36, 11 2021.