Bayesian Multinomial Logistic Regression for Numerous Categories [Work in Progress]

Jared D. Fisher **Kyle R. McEvoy***
University of California, Berkeley
Department of Statistics

December 11, 2020

Abstract

While multinomial logistic regression is a useful tool for classification among multiple categories, the posterior sampling of Bayesian implementations is computationally burdensome when working with a large number of categories. In this paper, we show that the appropriate data augmentation technique provides faster posterior sampling than alternatives in the literature. We demonstrate that the computation time of our posterior sampling approach only increases linearly as the number of categories increases. A forthcoming R library contains our sampling algorithm as well as other important algorithms in the literature. In the near future, more comparative methods will be added to our simulations studies, as well as comparisons of effective sample size and effective sampling rate.

Keywords: polychotomous, multiclass, classification, data augmentation

1 Introduction

Standard logistic regression is one of the most popular approaches to binary classification: the problem of assigning a probability of being in one of two categories to an observation. Compared to popular black-box machine learning approaches, logistic regression has the added bonus of interpretability: we can clearly state, or even plot, the model's functional relationship between any input variable and the probability of interest. This is possible as the core of the model is additive, namely a linear model of the log odds.

Multinomial logistic regression is the natural extension when considering more than two categories. To do this, we estimate the log odds between multiple potential outcomes using a linear function of covariates. Bayesian approaches to coefficient estimation in multinomial logistic regression are made more difficult compared to some other common methods, such as the probit regression of Albert and Chib (1993), because the resulting posterior distribution of the coefficients is not in a convenient analytical form. Thus, in order to estimate these coefficients' posterior distributions, it is often necessary to pursue Markov Chain Monte Carlo strategies that can be extremely computationally intensive. Our paper aims to propose a sampling method that can increase speed and reduce this computational intensity particularly in the case of classification problems with many categories.

2 Multinomial Logistic Regression and its Computational Challenge

We first define the multinomial logistic regression model. For $i \in \{1, \ldots, N\}$, let $y_i \in N_0^C$ to be a length-C vector of non-negative integers y_{ij}, $j = 1, ..., C$, which represent the counts of categories observed from n_i

*Listed alphabetically. Contact us at jared.fisher@berkeley.edu and/or kylemcevoy@berkeley.edu. We thank Li (Kelly) Kang, David W. Puelz, and Carlos M. Carvalho for their early work and helpful comments on this project.

independent observations pertaining to subject i, with each observation classified in one of C total categories. Let π_{ij} be the probability that one of the n_i observations in row i falls into category $j \in \{1, \ldots, C\}$. Thus, for each i, y_i will be distributed with a multinomial distribution:

$$y_i|\cdot \sim MN(n_i, \pi_{i1}, \ldots, \pi_{iC}). \tag{1}$$

Now suppose we were to fit a logistic regression model to each of the categorical probabilities. For each $j \in 1, \ldots, C$, let β_j be a vector of coefficients, and we model the probability that an observation from y_i falls into one each of the C different categories with:

$$\pi_{ij} = \pi_j(x_i) = \frac{\exp x_i^T \beta_j}{\sum_{k=1}^{C} \exp x_k^T \beta_k} \tag{2}$$

where each vector $x_i \in R^P$ contains the observed values of the P explanatory variables associated with observation y_i. From here, the Bayesian approach is to put a prior on the β_j.

However, while the π_{ij} are identifiable as presently defined, the β_j are not. If identification and interpretability of the β_j are desirable properties, the common practice is to assume that $\beta_j = 0$ for one of the j. In this paper, when this assumption is made, we assume it is for the last category: without loss of generality, let $\beta_C = 0$. The implication here though is that these coefficients β_j now represent the effects of x_i on the log odds between categories j and C. Of course, there are some cases where this is not a desirable property.

2.1 Approaches in the Literature

Building upon the ideas of Albert and Chib (1993), Holmes and Held (2006) pursued multiple data augmentation strategies for the logistic regression model. The strategy most relevant to our paper represents the logistic regression model as a set of binary observations that are determined by the sign of an auxiliary variable. These auxiliary variables are distributed according to a linear model with logistic errors. They then introduced a set of auxiliary variables following a Kolmogorov-Smirnov distribution, which after squaring and scaling determine the variance of the logistic errors. Using a multivariate normal prior on the cofficents and conditioning on these auxiliary variables and the data, the coefficients will still have a multivariate normal distribution. Furthermore, this method allows for sampling from the full conditional distributions of all of the parameters, so a Gibbs sampling strategy can be adopted without any Metropolis steps. However, implementing this sampling strategy relies on sampling from a truncated logistic distribution.

Holmes and Held (2006) further extend these ideas to multinomial regression by fixing the coefficients for a category at 0, such that the posterior sampling reduces to the same Gibbs sampling procedure, but now looping over a sequence of $C - 1$ logistic regression models one at a time, holding the others fixed. We will call their sampling algorithm "HH". In programming this sampling algorithm, we relied on the pseudo-code included in Holmes and Held (2006) with the corrections made by van der Lans (2011).

Frühwirth-Schnatter and Frühwirth (2010, 2012) expanded upon these ideas, representing the multinomial logistic regression model as a difference random utility model. Auxiliary variables are introduced as linear functions of the predictors with an additive multivariate logistic distributed error term. One consequence of this representation is that the error terms across categories are not independent, and the correlation of the errors between categories must be dealt with, regardless of the sampling technique (Frühwirth-Schnatter and Frühwirth, 2012). Despite this challenge, they were able to implement efficient samplers that performed better in computation time and effective sample size compared to HH on some common binary and categorical data sets (Frühwirth-Schnatter and Frühwirth, 2010). We will call their sampling algorithm "FSF".

While their paper does not address a multinomial/polychotomous version of their logistic regression approach Polson et al. (2013)'s data-augmentation strategy has become the go-to approach in some circles for Bayesian logistic regression. Using the Polya-Gamma latent variables, they construct a posterior sampler that is both simple to write down and does not need Metropolis-Hastings. Though the paper does not extend to the multiclass case, early versions of their R package *BayesLogit* contained code and documentation for

running the multinomial logistic, which we have built into our package.

2.2 MCMC Issue and Murray's solution

Holmes and Held (2006) and Frühwirth-Schnatter and Frühwirth (2010) developed data augmentation methods that allowed for full Gibbs sampling as the Markov chain Monte Carlo (MCMC) method for posterior estimation, as opposed to using the usually-less-efficient Metropolis algorithms. We proceed with an approach that still relies on Metropolis sampling, but that achieves speed improvements by removing the aforementioned denominator from the posterior distribution of the coefficients β_j. We now describe both this problem and our solution.

The aforementioned papers consider logistic regression where the information in covariates x_i enter into the model as parametric linear combinations, i.e. $x_i^T \beta_j$. However, these could be any category-specific function of the data, $f_j(x_i)$, such that

$$\pi_{ij} = \pi_j(x_i) = \frac{f_j(x_i)}{\sum_{k=1}^{C} f_k(x_i)}. \tag{3}$$

This is the case in Murray (2020), who fits these functions with Bayesian additive regression trees (Chipman et al., 2010). However, the denominator above presents a challenge with MCMC: the conditional posterior of f_j depends on f_k for all $k \neq j$. This yields two computational issues. First, you cannot parallelize the sampling of f_j for all j. Second, and perhaps more importantly, the calculation of the sum in the denominator above grows in complexity as C grows. However, Murray (2020) augments the data with the auxiliary variables ϕ_i:

$$(\phi_i | y_i, \cdot) \sim \text{Gamma}\left(n_i, \sum_{j=1}^{C} f_j(x_i)\right). \tag{4}$$

Now, given ϕ_i, the posterior distribution of regression function f_j does not depend on the posterior distribution of f_k, $k \neq j$. This opens new avenues to computational efficiency.

3 Our Approach

3.1 Our Model and Data Augmentation

We assume multinomial likelihood in Equation 1 with probabilities from Equation 2, and any standard prior on each β_j, denoted $p(\beta_j)$. These assumptions yield the following posterior on the vector of coefficients β_j.

$$p(\beta_j | \cdot) \propto \left[\prod_{i=1}^{N} \left(\frac{\exp\{x_i^T \beta_j\}}{\sum_{k=1}^{C} \exp\{x_i^T \beta_k\}}\right)^{y_{ij}}\right] p(\beta_j) \tag{5}$$

Note that the denominator $\sum_{k=1}^{C} \exp\{x_k^T \beta_k\}$ implies that the posterior of β_j is conditional upon the coefficients β_k from all other categories $k \neq j$. This is the same issue that the aforementioned general regression function case of Murray (2020), but now specifically with $f_j(x_i) = exp(x_i^T \beta_j)$, there are still the two issues for posterior sampling as the number of categories grows. First, this denominator grows in complexity as the number of categories grows. Second, the posterior samples must be drawn in sequence, whereas if that denominator was removed, then the coefficient vectors β_j could be drawn in parallel for all j. Thus we likewise introduce auxiliary variables ϕ_i as

$$(\phi_i | y_i, \cdot) \sim \text{Gamma}\left(n_i, \sum_{k=1}^{C} \exp(x_k^T \beta_k)\right). \tag{6}$$

Now, conditioning the full joint likelihood on the data and the set of ϕ_i, we get a posterior distribution of β_j that is no longer proportional in β_j to any of the other categories coefficients, β_k (for $k \neq j$), as seen in Equation 7.

$$p(\beta_j|\cdot) \propto exp\left\{\sum_{i=1}^{N} y_{ij}x_i^T\beta_j - \phi_i exp\left(x_i^T\beta_j\right)\right\} p(\beta_j) \tag{7}$$

Thus we can sample from this conditional distribution independently of the other categories. The implementation of this is the key contribution of our paper, allowing for parallelized solutions to classification problems with many categories.

However, this is without a free lunch. The posterior distribution in Equation 7 does not have an obvious connection to any distribution that we've found, which takes the fastest Gibbs samplers off the table. Yet, we find that simple Metropolis within Gibbs strategies as incredibly fast for our purposes.

3.2 Our Sampler

We proceeded with a Gibbs sampler that samples in two main steps:

1. For each $i = 1, ..., N$, ϕ_i is sampled from the distribution in Equation 6 conditioning on the data and the current draws/values of β_j, $j = 1, ..., C$. These can be drawn in parallel across each i.
2. For each $j = 1, ..., C$, β_j is sampled using a type of Metropolis algorithm conditioning on the data and the current draws/values of ϕ_i, $i = 1, ..., N$. These can be drawn in parallel across each j.

To sample each β_j we use a sequence of univariate random walk Metropolis samplers for each coefficient in the β_j vector. This simple approach preserves flexibility and can generalize to many scenarios. In contrast, a multivariate normal proposal distribution would require a case-specific covariance matrix to sample efficiently. Thus, we set up simple, univariate normal proposal distributions for each individual coefficient.

Let $\beta_{jp}^{(t-1)}$ be the coefficient in category $j = 1, ..., C$ for predictor $p = 1, ..., P$, drawn at MCMC iteration $t-1$. The proposal distribution for generating $\beta_{jp}^{(t)}$ is $q(\beta_{jp}^{(t-1)}) = \beta_{jp}^{(t-1)} + W$ where $W \sim N(0, \sigma_{jp}^2)$. While using univariate samplers mean that there is no covariance matrix to tune, there is a variance term for each coefficient, PC in total. While these can be pre-specified by the user, we have automated their tuning. During the burn-in phase of MCMC, the value of σ_{jp} is adjusted if the acceptance rate is outside of an acceptable range (currently 20%-40%), akin to Fellingham and Fisher (2018). Specifically, for some predetermined number of MCMC iterations η, which is much less than the number of burn-in iterations, we check the number of values accepted during those η iterations. If there were more than 0.4η new values accepted in these η iterations, we double the value of σ_{jp}. If there were fewer than 0.2η new values accepted, we reduce σ_{jp} by 10%, i.e. set its value to $0.9\sigma_{jp}$. In this way, as long as the initially provided values of both η and σ_{jp} are reasonable, the σ_{jp} will be automatically tuned to optimal values. For example, if performing 2000 burn-in iterations, then $\eta = 100$ provides $2000/\eta = 20$ windows with which to tune σ_{jp} for all j, p.

Presently, our code allows for both normal and flat priors on the coefficients for the categories, β_j. In our program we assumed that these priors were an i.i.d. sample across the different categories. While the full i.i.d. assumption is not necessary for the sampling procedure we are using, we do rely on independence between the categories. We would not be able to sample the posterior independently if we chose a prior that had correlation between the categories. The program allows for multivariate normal vector means and covariance matrices to be specified by the user.

4 Simulation Study

In order to compare the performance of these algorithms as the number of categories, C, in the classification problem increased, we generated simulated test data. For each test of the algorithms, we fixed a number of

categories C. Then we generated 999 new observations of simulated data.

First, we independently generated a $N \times P$ matrix X of predictor variables from standard normal distributions. Then a $P \times C$ matrix of random normal weights, B was generated in the same fashion. In order to generate categorical probabilities we took $p = \text{softmax}(XB)$. Finally, using these probabilities we generated the categorical observation with $y_i \sim \text{categorical}(p)$ for $i = 1, \ldots N$.

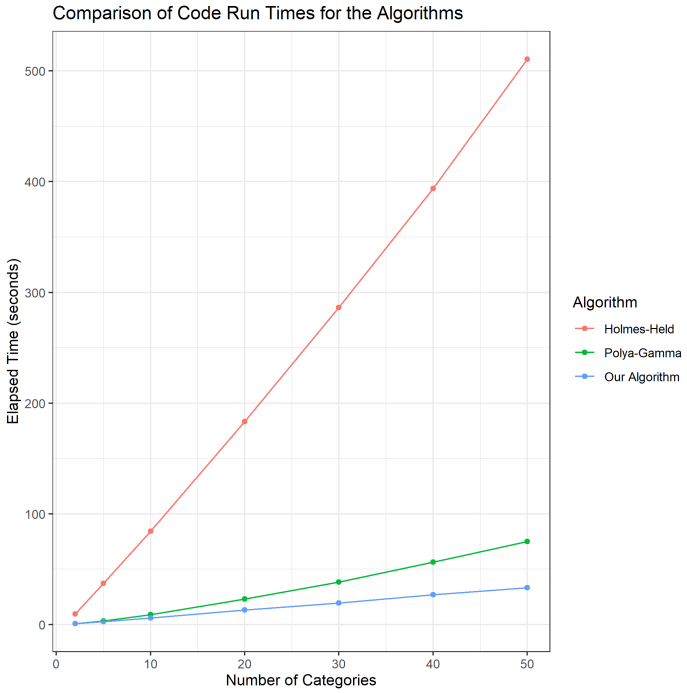

Figure 1: Plot of the run times for 1000 MCMC iterations using the different algorithms on 999 simulated observations.

Then the algorithm was run for a total of 1000 MCMC iterations. We measured the elapsed time it took the code to complete execution with the *system.time()* function in R. In subsequent work we will compare the Effective Sample Size (ESS) of the algorithms to compare the degree to which each algorithm is limiting auto-correlation. We will also add the Frühwirth-Schnatter and Frühwirth (2010) algorithm to the comparison.

Algorithm	Elapsed Time (s)						
	Number of Categories						
	2	5	10	20	30	40	50
Holmes-Held	9.600	37.250	84.280	183.280	286.340	393.660	510.680
Polya-Gamma	0.870	3.320	8.940	23.070	38.300	56.590	75.150
Our Algorithm	0.910	2.780	6.130	13.140	19.660	27.020	33.480

Table 1: Table of simulation study run times for the algorithms at different numbers of categories.

5 Conclusion and Future Work

By combining the ideas of Murray (2020) and the case of additive logistic regression with the simplicity of univariate Metropolis sampling, we have developed a multinomial logistic regression posterior sampling method that is faster than others in the literature. We see several avenues that we will pursue in the near future to complete this paper.

We plan to extend our complement of competing algorithms to include the multinomial version of Gramacy and Polson (2012) logistic regression, mentioned in their discussion and extensions section. We also recently became aware of a different approach in Scott (2011) that would be worth exploring. We also plan to expand future simulation studies to include Frühwirth-Schnatter and Frühwirth (2010)'s algorithm.

Of course, the main desire is for an MCMC algorithm that minimizes the computation time needed to obtain the desired effective sample size, not necessarily the number of iterations. In future comparisons, we will explore the effective sample size and effective sample size rate by examining auto-correlation of the resulting chains.

Finally, we are currently developing an R package that implements our data augmentation sampler for multinomial logistic regression. We are also including the other algorithms discussed in this paper, as some do not have readily-available implementations C++ that can be called in R. We use the RcppArmadillo package, giving us access to the Armadillo C++ library for linear algebra within an R framework. In addition, we could improve our current code by taking advantage of the opportunity for parallelization across the categories.

References

Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669 – 679.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). BART: Bayesian additive regression trees. *Ann. Appl. Stat.*, 4(1):266–298.

Fellingham, G. and Fisher, J. (2018). Predicting home run production in Major League Baseball using a Bayesian semiparametric model. *The American Statistician*, 72:253 – 264.

Frühwirth-Schnatter, S. and Frühwirth, R. (2010). *Data Augmentation and MCMC for Binary and Multinomial Logit Models*, pages 111–132.

Frühwirth-Schnatter, S. and Frühwirth, R. (2012). Bayesian inference in the multinomial logit model. *Austrian Journal of Statistics*, 41:27–43.

Gramacy, R. B. and Polson, N. G. (2012). Simulation-based regularized logistic regression. *Bayesian Analysis*, 7(3):567–590.

Holmes, C. C. and Held, L. (2006). Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis*, 1(1):145–168.

Murray, J. S. (2020+). Log-linear Bayesian additive regression trees for multinomial logistic and count regression models. *To appear in the Journal of the American Statistical Association.*

Polson, N. G., Scott, J. G., and Windle, J. (2013). Bayesian inference for logistic models using Pólya–gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349.

Scott, S. L. (2011). Data augmentation, frequentist estimation, and the Bayesian analysis of multinomial logit models. *Statistical Papers*, 52(1):87–109.

van der Lans, R. (2011). Bayesian estimation of the multinomial logit model: A comment on Holmes and Held, "Bayesian auxiliary variable models for binary and multinomial regression". *Bayesian Analysis*, 6(2):353 – 355.