Journal of Machine Learning Research 1 (2017) xxx Submitted xxx; Published xxx

Scaling up Data Augmentation MCMC via Calibration

Leo L. Duan[*], **James E. Johndrow**[†], **David B. Dunson**[‡]

Editor:

Abstract: There has been considerable interest in making Bayesian inference more scalable. In big data settings, most literature focuses on reducing the computing time per iteration, with less focused on reducing the number of iterations needed in Markov chain Monte Carlo (MCMC). This article focuses on data augmentation MCMC (DA-MCMC), a widely used technique. DA-MCMC samples tend to become highly autocorrelated in large samples, due to a miscalibration problem in which conditional posterior distributions given augmented data are too concentrated. This makes it necessary to collect very long MCMC paths to obtain acceptably low MC error. To combat this inefficiency, we propose a family of calibrated data augmentation algorithms, which appropriately adjust the variance of conditional posterior distributions. A Metropolis-Hastings step is used to eliminate bias in the stationary distribution of the resulting sampler. Compared to existing alternatives, this approach can dramatically reduce MC error by reducing autocorrelation and increasing the effective number of DA-MCMC samples per computing time. The approach is simple and applicable to a broad variety of existing data augmentation algorithms, and we focus on three popular models: probit, logistic and Poisson log-linear. Dramatic gains in computational efficiency are shown in applications.

KEY WORDS: Bayesian Probit; Bayesian Logit; Big n; Data Augmentation; Maximal Correlation; Polya-Gamma for Poisson.

∗. Department of Statistical Science, Duke University, Durham, NC, email: leo.duan@duke.edu
†. Department of Statistics, Stanford University, Stanford, CA, email: johndrow@stanford.edu
‡. Department of Statistical Science, Duke University, Durham, NC, email: dunson@duke.edu

1. Introduction

With the deluge of data in many modern application areas, there is pressing need for scalable computational algorithms for inference from such data, including uncertainty quantification (UQ). Somewhat surprisingly, even as the volume of data increases, uncertainty often remains sizable. Examples in which this phenomenon occurs include financial fraud detection (Ngai et al., 2011), disease mapping (Wakefield, 2007) and online click-through tracking (Wang et al., 2010). Bayesian approaches provide a useful paradigm for quantifying uncertainty in inferences and predictions in these and other settings.

The standard approach to Bayesian posterior computation is Markov chain Monte Carlo (MCMC) and related sampling algorithms. However, conventional MCMC algorithms often scale poorly in problem size and complexity. Due to its sequential nature, the computational cost of MCMC is the product of two factors: the evaluation cost at each sampling iteration and the total number of iterations needed to obtain an acceptably low Monte Carlo (MC) error. While a substantial literature has developed focusing on decreasing computational cost per iteration (Minsker et al. (2014); Maclaurin and Adams (2015); Srivastava et al. (2015); Conrad et al. (2015) among others), very little has been done to reduce the number of iterations needed to produce a desired MC error in posterior summaries as the problem size grows.

A major concern in applying MCMC algorithms in big data problems is that the level of autocorrelation in the MCMC path may increase with the size of the data. Markov chains with high autocorrelation tend to produce a low *effective sample size (ESS)* per unit computational time, which is informally known as the *slow mixing* problem. The ESS is designed to compare the information content in the sampling iterations relative to a gold standard Monte Carlo algorithm that collects independent samples. If the number of effective samples in 1,000 iterations is only 10, then the MCMC algorithm will need to be run 100 times as long as the gold standard algorithm to obtain the same MC error in posterior summaries. Such a scenario is not unusual in big data problems, leading MCMC algorithms to face a *double burden*, with the time per iteration increasing and it becoming necessary to collect more and more iterations.

This double burden has led many members of the machine learning community to abandon MCMC in favor of more easily scalable alternatives, such as variational approximations. Unfortunately these approaches lack theoretical guarantees and often badly under-estimate posterior uncertainty. Hence, there has been substantial interest in recent years in designing scalable MCMC algorithms. The particular focus of this paper is on a popular and broad class of Data Augmentation (DA)-MCMC algorithms. DA-MCMC algorithms are used routinely in many classes of models, with the algorithms of Albert and Chib (1993) for probit models and Polson et al. (2013) for logistic models particularly popular. Our focus is on improving the performance of such algorithms in big data settings in which issues can arise in terms of both the time per iteration and the mixing. The former problem can be addressed using any of a broad variety of existing approaches, and we focus here on the slow mixing problem.

Johndrow et al. (2016) discovered that popular DA-MCMC algorithms have small effective sample sizes in large data settings involving imbalanced data. For example, data may be binary with a high proportion of zeros. A key insight is that the reason for this problem

is a discrepancy in the rates at which Gibbs step sizes and the width of the high-probability region of the posterior converge to zero as n increases. In particular, the conditional posterior given the augmented data may simply be too concentrated relative to the marginal posterior, with this problem amplified as the data sample size increases. There is a rich literature on methods for accelerating mixing in DA-MCMC algorithms using tricks ranging from reparameterization to parameter-expansion (Liu and Wu, 1999; Meng and Van Dyk, 1999; Papaspiliopoulos et al., 2007). However, we find that such approaches fail to address the miscalibration problem and have no impact on the worsening mixing rate with increasing data sample size n.

The focus of this article is on proposing a general new class of algorithms for addressing the fundamental miscalibration problem that leads to worsening mixing of DA-MCMC with n. In particular, the key idea underlying our proposed class of *calibrated* DA (CDA) algorithms is to introduce auxiliary parameters that change the variance of full conditional distributions for one or more parameters. These auxiliary parameters can adapt with the data sample size n to fundamentally address the key problem causing the worsening mixing with n. In general, the invariant measure of CDA-MCMC does not correspond exactly to the true joint posterior distribution of interest. Instead, we can view CDA-MCMC as representing a computationally more efficient perturbation of the original Markov chain. The perturbation error can be eliminated using Metropolis-Hastings. Compared to other adaptive Metropolis-Hastings algorithms, which often require carefully chosen multivariate proposals and complicated adaptation with multiple chains (Tran et al., 2016), CDA-MCMC only requires a simple modification to Gibbs sampling steps. We show the auxiliary parameters can be efficiently adapted for each type of data augmentation, via minimizing the difference between Fisher information of conditional and marginal distributions.

2. Calibrated Data Augmentation

Data augmentation Gibbs samplers alternate between sampling latent data z from their conditional posterior distribution given model parameters θ and observed data y, and sampling parameters θ given z and y; either of these steps can be further broken down into a series of full conditional sampling steps but we focus for simplicity on algorithms of the form:

$$\begin{aligned} z \mid \theta, y &\sim \pi(z; \theta, y) \\ \theta \mid z, y &\sim f(\theta; z, y), \end{aligned} \tag{1}$$

where f belongs to a location-scale family, such as the Gaussian. Popular data augmentation algorithms are designed so that both of these sampling steps can be conducted easily and efficiently; e.g., sampling the latent data for each subject independently and then drawing θ simultaneously (or at least in blocks) from a multivariate Gaussian or other standard distribution. This effectively avoids the need for tuning, which is a major issue for Metropolis-Hastings algorithms, particularly when θ is high-dimensional. Data augmentation algorithms are particularly common for generalized linear models (GLMs), with $\mathbb{E}(y_i \mid x_i, \theta) = g^{-1}(x_i\theta)$ and a conditionally Gaussian prior distribution chosen for θ. We focus in particular on Poisson log-linear, binomial logistic, and binomial probit as motivating examples.

Consider a Markov kernel $K((\theta, z); \cdot)$ with invariant measure Π and update rule of the form (1), and a Markov chain (θ_t, z_t) on a state space $\Theta \times \mathcal{Z}$ evolving according to K. We will abuse notation in writing $\Pi(d\theta) = \int_{z \in \mathcal{Z}} \Pi(d\theta, dz)$. The lag-1 autocorrelation for a function $g : \Theta \to \mathbb{R}$ at stationarity can be expressed as the Bayesian fraction of missing information (Papaspiliopoulos et al. (2007), Rubin (2004), Liu (1994b))

$$\gamma_g = 1 - \frac{\mathbb{E}[\text{var}(g(\theta) \mid z)]}{\text{var}(g(\theta))}, \tag{2}$$

where the integrals in the numerator are with respect to $\Pi(d\theta, dz)$ and in the denominator with respect to $\Pi(d\theta)$. Let

$$L_2(\Pi) = \left\{ g : \Theta \to \mathbb{R}, \int_{\theta \in \Theta} \{g(\theta)\}^2 \Pi(d\theta) < \infty \right\}$$

be the set of real-valued, Π square-integrable functions. The *maximal autocorrelation*

$$\gamma = \sup_{g \in L^2(\Pi)} \gamma_g = 1 - \inf_{g \in L^2(\Pi)} \frac{\mathbb{E}[\text{var}(g(\theta) \mid z)]}{\text{var}(g(\theta))}$$

is equal to the geometric convergence rate of the data augmentation Gibbs sampler (Liu (1994b)). For $g(\theta) = \theta_j$ a coordinate projection, the numerator of the last term of (2) is, informally, the average squared step size for the augmentation algorithm at stationarity in direction j, while the denominator is the squared width of the bulk of the posterior in direction j. Consequently, γ will be close to 1 whenever the average step size at stationarity is small relative to the width of the bulk of the posterior.

The purpose of CDA is to introduce additional parameters that allow us to control the step size relative to the posterior width – roughly speaking, the ratio in (2) – with greater flexibility than reparametrization or parameter expansion. The flexibility gains are achieved by allowing the invariant measure to change as a result of the introduced parameters. The additional parameters, which we denote (r, b), correspond to a collection of reparametrizations, each of which defines a proper (but distinct) likelihood $L_{r,b}(\theta; y)$, and for which there exists a Gibbs update rule of the form (1). In general, b will correspond to a location parameter and r a scale parameter that are tuned to increase $\mathbb{E}[\text{var}(g(\theta) \mid z)]\{\text{var}(g(\theta))\}^{-1}$, although the exact way in which they enter the likelihood and corresponding Gibbs update depend on the application. The reparametrization also has the property that $L_{1,0}(\theta; y) = L(\theta; y)$, the original likelihood. The resulting Gibbs sampler, which we refer to as CDA Gibbs, has θ-marginal invariant measure $\Pi_{r,b}(\theta; y) \propto L_{r,b}(\theta; y)\Pi^0(\theta)$, where $\Pi^0(\theta)$ is the prior. Ultimately, we are interested in $\Pi_{1,0}(\theta; y)$, so we use CDA Gibbs as an efficient proposal for Metropolis-Hastings. That is, we propose θ^* from $Q(\theta; \cdot)$ where

$$Q_{r,b}(\theta; A) = \int_{(\theta^*, z) \in A \times \mathcal{Z}} \pi_{r,b}(z; \theta, y) f_{r,b}(\theta^*; z, y) dz d\theta^* \tag{3}$$

for $A \subseteq \Theta$, where $\pi_{r,b}$ and $f_{r,b}$ denote the conditional densities of z and θ in the Gibbs sampler with invariant measure $\Pi_{r,b}$. By tuning working parameters during an adaptation phase to reduce the autocorrelations and increase the Metropolis-Hastings acceptance rate,

we can select values of the working parameters that yield a computationally efficient algorithm. Tuning is facilitated by the fact that the M-H acceptance ratios using this proposal kernel have a convenient form, which is a nice feature of using Gibbs to generate M-H proposals.

Remark 1 *The CDA M-H acceptance ratio is given by*

$$1 \wedge \frac{L(\theta';y)\Pi^0(\theta')Q_{r,b}(\theta;\theta')}{L(\theta;y)\Pi^0(\theta)Q_{r,b}(\theta';\theta)} = 1 \wedge \frac{L(\theta';y)L_{r,b}(\theta;y)}{L(\theta;y)L_{r,b}(\theta';y)} \tag{4}$$

A general strategy for tuning is given in Section 3.3.

We give a basic convergence guarantee that holds for the CDA M-H under weak assumptions on $L_{r,b}$, which is based on (Roberts and Smith, 1994, Theorem 3, also pp. 214). Basically, one needs $\Pi(\cdot) \ll \Pi_{r,b}(\cdot)$ for all r, b, where for two probability measures μ, ν, $\mu(\cdot) \ll \nu(\cdot)$ means μ is absolutely continuous with respect to ν.

Remark 2 (Ergodicity) *Assume that $\Pi(d\theta)$ and $\Pi_{r,b}(d\theta)$ have densities with respect to Lebesgue measure on $\mathbb{R}^p$, and that $K_{r,b}((\theta,z);(\theta',z')) > 0 \,\forall\, ((\theta,z),(\theta',z')) \in (\Theta \times \mathcal{Z}) \times (\Theta \times \mathcal{Z})$. Then,*

- *For fixed r, b, CDA Gibbs is ergodic with invariant measure $\Pi_{r,b}(d\theta, dz)$.*
- *A Metropolis-Hastings algorithm with proposal kernel $Q_{r,b}(\theta';\theta)$ as defined in (3) with fixed r, b is ergodic with invariant measure $\Pi(d\theta)$.*

Proofs are located in the Appendix.

2.1 Initial Example: Probit with Intercept Only

We use a simple example to illustrate CDA. Consider an intercept-only probit

$$y_i \sim \text{Bernoulli}(p_i), \quad p_i = \Phi(\theta) \quad i = 1, \ldots, n$$

and improper prior $\Pi^0(\theta) \propto 1$. The basic data augmentation algorithm (Tanner and Wong, 1987; Albert and Chib, 1993) has the update rule

$$z_i \mid \theta, y_i \sim \begin{cases} \text{No}_{[0,\infty)}(\theta, 1) & \text{if } y_i = 1 \\ \text{No}_{(-\infty,0]}(\theta, 1) & \text{if } y_i = 0 \end{cases} \qquad i = 1, \ldots, n$$

$$\theta \mid z, y \sim \text{No}\left(n^{-1}\sum_i z_i, n^{-1}\right),$$

where $\text{No}_{[a,b]}(\mu, \sigma^2)$ is the normal distribution with mean μ and variance σ^2 truncated to the interval $[a, b]$. Johndrow et al. (2016) show that when $\sum_i y_i = 1$, $\text{var}(\theta_t \mid \theta_{t-1})$ is approximately $n^{-1}\log n$, while the width of the high probability region of the posterior is order $(\log n)^{-1}$, leading to slow mixing.

As the conditional variance $\text{var}(\theta \mid z, y)$ is independent of z, we introduce a scale parameter r in the update for z, then adjust the conditional mean by a location parameter

b. This is equivalent to changing the scale of $z_i \mid \theta, y_i$ from 1 to r and the mean from θ to $\theta + b$. These adjustments yield

$$\text{pr}(y_i = 1|\theta, r, b) = \int_0^\infty \frac{1}{\sqrt{2\pi r}} \exp\left(-\frac{(z_i - \theta - b)^2}{2r^2}\right) dz_i = \Phi\left(\frac{\theta + b}{\sqrt{r}}\right), \tag{5}$$

leading to the modified data augmentation algorithm

$$z_i \mid \theta, y_i \sim \begin{cases} \text{No}_{[0,\infty)}(\theta + b, r) & \text{if } y_i = 1 \\ \text{No}_{(-\infty,0]}(\theta + b, r) & \text{if } y_i = 0 \end{cases} \qquad i = 1, \ldots, n \tag{6}$$

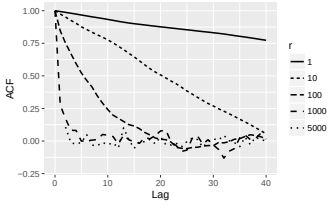
$$\theta \mid z, y \sim \text{No}\left(n^{-1}\sum_i (z_i - b), n^{-1}r\right).$$

To achieve step sizes consistent with the width of the high posterior probability region, we need $n^{-1}r \approx (\log n)^{-1}$, so $r \approx n/\log n$. To preserve the original target, we use (6) to generate an M-H proposal θ^*. By Remark 1, the M-H acceptance probability is given by (4) with $L_{r,b}(\theta; y_i) = \Phi\big((\theta + b)r^{-1/2}\big)^{y_i}\Phi\big(-(\theta + b)r^{-1/2}\big)^{(1-y_i)}$ and $L(\theta; y_i) = L_{1,0}(\theta; y_i)$. Setting $r_i = 1$ and $b_i = 0$ leads to acceptance rate of 1, which corresponds to the original Gibbs sampler.

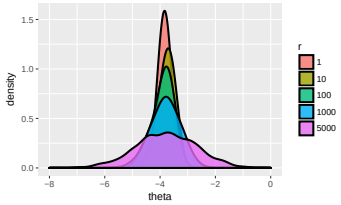
To illustrate, we consider $\sum_i y_i = 1$ and $n = 10^4$. Letting $r = n/\log n$, we then choose the b_i's to increase the acceptance rate in the M-H step. In this simple example, it is easy to compute a "good" value of b_i, since $b_i = -3.7(\sqrt{r} - 1)$ results in $\text{pr}(y_i = 1) = \Phi(-3.7) = n^{-1}\sum_i y_i \approx 10^{-4}$ in the proposal distribution, centering the proposals near the MLE for p_i.

We perform computation for these data with different values of r ranging from $r = 1$ to $r = 5,000$, with $r = 1,000 \approx n/\log n$ corresponding to the theoretically optimal value. Figure 2(a) plots autocorrelation functions (ACFs) for these different samplers without M-H adjustment. Autocorrelation is very high even at lag 40 for $r = 1$, while increasing r leads to dramatic improvements in mixing. There are no further gains in increasing r from the theoretically optimal value of $r = 1,000$ to $r = 5,000$. Figure 2(b) shows kernel-smoothed density estimates of the posterior of θ without M-H adjustment for different values of r and based on long chains to minimize the impact of Monte Carlo error; the posteriors are all centered on the same values but with variance increasing somewhat with r. With M-H adjustment such differences are removed; the M-H step has acceptance probability close to one for $r = 10$ and $r = 100$, about 0.6 for $r = 1,000$, and 0.2 for $r = 5,000$.

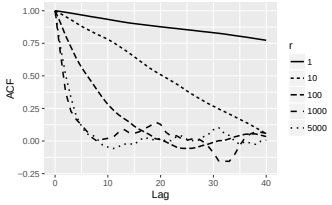
Figure 1: Autocorrelation functions (ACFs) and kernel-smoothed density estimates for different CDA samplers in intercept-only probit model.



(a) ACF for CDA without M-H adjustment.



(b) Posterior density estimates without M-H adjustment.



(c) ACF for CDA with M-H adjustment

3. Specific Algorithms

In this section, we describe CDA algorithms for general probit and logistic regression, and describe a general strategy for tuning r, b.

3.1 Probit Regression

Consider the probit regression:

$$y_i \sim \text{Bernoulli}(p_i), \quad p_i = \Phi(x_i\theta) \quad i = 1, \ldots, n$$

with improper prior $\Pi^0(\theta) \propto 1$. The data augmentation sampler (Tanner and Wong, 1987; Albert and Chib, 1993) has the update rule

$$\begin{aligned} z_i \mid \theta, x_i, y_i &\sim \begin{cases} \text{No}_{[0,\infty)}(x_i\theta, 1) & \text{if } y_i = 1 \\ \text{No}_{(-\infty,0]}(x_i\theta, 1) & \text{if } y_i = 0 \end{cases} \quad i = 1, \ldots, n \\ \theta \mid z, x, y &\sim \text{No}((X'X)^{-1}X'z, (X'X)^{-1}). \end{aligned}$$

Liu and Wu (1999) and Meng and Van Dyk (1999), among others, previously studied this algorithm and proposed to rescale θ through parameter expansion. However, this modification does not impact the conditional variance of θ and thus does not directly increase typical step sizes.

Our approach is fundamentally different, since we directly adjust the conditional variance. Similar to the intercept only model, we modify $\text{var}(\theta|z)$ by changing the scale of each z_i. Since the conditional variance is now a matrix, for flexible tuning, we let r and b vary over index i, yielding update rule

$$\begin{aligned} z_i \mid \theta, x_i, y_i &\sim \begin{cases} \text{No}_{[0,\infty)}(x_i\theta + b_i, r_i) & \text{if } y_i = 1 \\ \text{No}_{(-\infty,0]}(x_i\theta + b_i, r_i) & \text{if } y_i = 0 \end{cases} \quad i = 1, \ldots, n \\ \theta \mid z, X &\sim \text{No}((X'R^{-1}X)^{-1}X'R^{-1}(z-b), (X'R^{-1}X)^{-1}), \end{aligned} \tag{7}$$

where $R = \text{diag}(r_1, \ldots, r_n)$, $b = (b_1, \ldots, b_n)'$, under the Bernoulli likelihood:

$$\text{pr}(y_i = 1|\theta, x_i, r_i, b_i) = \int_0^\infty \frac{1}{\sqrt{2\pi r_i}} \exp\left(-\frac{(z_i - x_i\theta - b_i)^2}{2r_i}\right) dz_i = \Phi\left(\frac{x_i\theta + b_i}{\sqrt{r_i}}\right). \tag{8}$$

For fixed $r = (r_1, \ldots, r_n)$ and $b = (b_1, \ldots, b_n)$, (8) defines a proper Bernoulli likelihood for y_i conditional on parameters, and therefore the transition kernel $K_{r,b}((\theta, z); \cdot)$ defined by the Gibbs update rule in (7) would have a unique invariant measure for fixed r, b, which we denote $\Pi_{r,b}(\theta, z \mid y)$.

For insight into the relationship between r and step size, consider the θ-marginal autocovariance in a Gibbs sampler evolving according to $K_{r,b}$:

$$\begin{aligned} \text{cov}_{r,b}(\theta_t \mid \theta_{t-1}, X, z, y) &= (X'R^{-1}X)^{-1} + (X'R^{-1}X)^{-1}X'R^{-1}\text{cov}(z-b|R)R^{-1}X(X'R^{-1}X)^{-1} \\ &\geq (X'R^{-1}X)^{-1}, \end{aligned} \tag{9}$$

In the special case where $r_i = r_0$ for all i, we have

$$\text{cov}_{r,b}(\theta_t \mid \theta_{t-1}, X, z, y) \geq r_0(X'X)^{-1},$$

so that all of the conditional variances are increased by at least a factor of r_0. This holds uniformly over the entire state space, so it follows that

$$\mathbb{E}_{\Pi_{r,b}}[\mathrm{var}(\theta_j \mid z)] \geq r_0 \mathbb{E}_{\Pi}[\mathrm{var}(\theta_j \mid z)].$$

The key to CDA is to choose r, b to make $\mathbb{E}_{\Pi_{r,b}}[\mathrm{var}(\theta_j \mid z)]$ close to $\mathrm{var}_{\Pi_{r,b}}(\theta_j \mid z)$, while additionally maximizing the M-H acceptance probability. We defer the choice for r, b and their effects to the last subsection.

3.2 Logistic Regression

Calibration was easy to achieve in the probit examples, because $\mathrm{var}(\theta|z, y)$ does not involve the latent variable z. In cases in which the latent variable impacts the variance of the conditional posterior distribution of θ, we propose to stochastically increase $\mathrm{var}(\theta|z, y)$ by modifying the distribution of z. We focus on the logistic regression model with

$$y_i \sim \mathrm{Bernoulli}(p_i), \quad p_i = \frac{\exp(x_i\theta)}{1+\exp(x_i\theta)} \quad i = 1, \ldots, n$$

and improper prior $\Pi^0(\theta) \propto 1$. For this model, Polson et al. (2013) proposed Polya-Gamma data augmentation:

$$\begin{aligned} z_i &\sim \mathrm{PG}(1, |x_i\theta|) \quad i = 1, \ldots, n, \\ \theta &\sim \mathrm{No}\left((X'ZX)^{-1}X'(y - 0.5), (X'ZX)^{-1}\right), \end{aligned}$$

where $Z = \mathrm{diag}(z_1, \ldots, z_n)$. This algorithm relies on expressing the logistic regression likelihood as

$$L(x_i\theta; y_i) = \int \exp\{x_i\theta(y_i - 1/2)\} \exp\left\{ -\frac{z_i(x_i\theta)^2}{2}\right\} \mathrm{PG}(z_i \mid 1, 0) dz_i,$$

where $\mathrm{PG}(a_1, a_2)$ denotes the Polya-Gamma distribution with parameters a_1, a_2, with $\mathbb{E}z_i = a_1/(2a_2)\tanh(a_2/2)$.

We replace $\mathrm{PG}(z_i \mid 1, 0)$ with $\mathrm{PG}(z_i \mid r_i, 0)$ in the step for updating the latent data. Since $\mathbb{E}_z\mathrm{var}(\theta|z, y)$ lacks closed-form, we focus on the precision matrix $\mathbb{E}_z\big(\mathrm{var}(\theta|z, y)\big)^{-1} = X'\mathbb{E}ZX$. Smaller r_i can lead to smaller $\mathbb{E}z_i$, providing a route to calibration. Applying the bias-adjustment term b_i to the linear predictor $\eta_i = x_i\theta$ leads to

$$\begin{aligned} L_{r,b}(x_i\theta; y_i) =& \int_0^\infty \exp\{(x_i\theta + b_i)(y_i - r_i/2)\} \exp\left\{ -\frac{z_i(x_i\theta + b_i)^2}{2}\right\} \mathrm{PG}(z_i \mid r_i, 0) dz_i \\ =& \frac{\exp\{(x_i\theta + b_i)y_i\}}{\{1+\exp(x_i\theta + b_i)\}^{r_i}}, \end{aligned} \tag{10}$$

and the update rule for the CDA Gibbs sampler is then

$$\begin{aligned} z_i &\sim \mathrm{PG}(r_i, |x_i\theta + b_i|) \quad i = 1, \ldots, n, \\ \theta^* &\sim \mathrm{No}\left((X'ZX)^{-1}X'(y - r/2 - Zb), (X'ZX)^{-1}\right), \end{aligned}$$

where $r = (r_1, \ldots, r_n)$. By (4), the M-H acceptance probability is

$$1 \wedge \prod_i \frac{\{1+\exp(x_i\theta)\}\{1+\exp(x_i\theta^* + b_i)\}^{r_i}}{\{1+\exp(x_i\theta^*)\}\{1+\exp(x_i\theta + b_i)\}^{r_i}}.$$

3.3 Choice of Calibration Parameters

As illustrated in the previous subsection, efficiency of CDA is dependent on a good choice of the calibration parameters $r = (r_1, \ldots, r_n)$ and $b = (b_1, \ldots, b_n)$. We propose a simple and efficient algorithm for calculating "good" values of these parameters relying on Fisher information. Although our choice of calibration parameters relies on large data sample arguments, we find that this calibration approach also works well in smaller data samples.

Our goal is to adjust the conditional variance under calibration of (r, b) to approximately match the marginal variance under the exact target distribution. The inverses of the following Fisher information provide useful approximation to the two posterior covariances.

$$\begin{aligned}\left(\mathcal{I}_{y|\theta}(\theta)\right)_{i,j} &= \mathbb{E}_{y|\theta}\left[\left(\frac{\partial}{\partial\theta_i}\log L(y;\theta)\right)\left(\frac{\partial}{\partial\theta_j}\log L(y;\theta)\right)\right],\\ \left(\mathcal{I}_{y|\theta,z}(\theta;r,b)\right)_{i,j} &= \mathbb{E}_{y|\theta,z}\left[\left(\frac{\partial}{\partial\theta_i}\log L_{r,b}(y,z;\theta)\right)\left(\frac{\partial}{\partial\theta_j}\log L_{r,b}(y,z;\theta)\right)\right]\end{aligned}$$

for $i = 1, \ldots, p$, $j = 1, \ldots, p$, with $\mathbb{E}_{y|\theta}$ taken over the distribution of y under the target marginal $L(y;\theta)$ and $\mathbb{E}_{y|\theta,z}$ taken over the conditional distribution of y under the augmented $L_{r,b}(y,z;\theta)$ with the calibration of (r,b). Since $\mathcal{I}_{y|\theta,z}(\theta;r,b)$ depends on random z, we marginalize over the conditional distribution of z under $L_{r,b}(\theta;y)$ and obtain $\mathbb{E}_{z|\theta}\mathcal{I}_{y|\theta,z}(\theta;r,b)$. Via adjusting r, one can then minimize the difference between $\mathcal{I}_{y|\theta}(\theta)$ and $\mathbb{E}_{z|\theta}\mathcal{I}_{y|\theta,z}(\theta;r,b)$.

Often, one can avoid computing the full Fisher information. For each class of models under the same data augmentation, they share the same form of conditional likelihoods for $y \mid \eta(\theta)$ given a mapping $\eta(\theta) : \mathbb{R}^p \to \mathbb{R}^d$. For example, all Bernoulli probit models follow $y_i \mid \eta_i(\theta) \overset{iid}{\sim} \text{Bernoulli}(\Phi(\eta_i(\theta)))$, except with different $\eta(\theta)$. The above full Fisher information can be rewritten as

$$\begin{aligned}\mathcal{I}_{y|\theta}(\theta) &= \dot{\eta}\mathcal{I}_{y|\theta}(\eta(\theta))\dot{\eta}',\\ \mathcal{I}_{y|\theta,z}(\theta;r,b) &= \dot{\eta}\mathcal{I}_{y|\theta,z}(\eta(\theta);r,b)\dot{\eta}',\end{aligned}$$

where $\dot{\eta}$ denotes the p-by-d gradient matrix consisting of the partial derivative $\partial\eta_k(\theta)/\partial\theta_j$ of the kth output $\eta_k(\theta)$ with respect to θ_j. It suffices to reduce the difference between $\mathcal{I}_{y|\theta}(\eta(\theta))$ and $\mathcal{I}_{y|\theta,z}(\eta(\theta);r,b)$ instead of the full Fisher information. The solution is a function of η, with form invariant to models under the same conditional likelihood of $y \mid \eta$.

In all CDA algorithms presented in this article, $\mathcal{I}_{y|\theta}(\eta(\theta))$ and $\mathcal{I}_{y|\theta,z}(\eta(\theta);r,b)$ are simple diagonal matrices, and it can be made exactly $\mathcal{I}_{y|\theta}(\theta) = \mathbb{E}_{z|\theta}\mathcal{I}_{y|\theta,z}(\theta;r,b)$ for given θ with a closed-form solution. As there could be more complicated scenarios, we suggest the following. When the difference cannot be simply eliminated, one could utilize a metric between two matrices, such as Rao's distance $\{\text{tr}[\log(A^{-1/2}BA^{-1/2})^2]\}^{1/2}$ with tr as the trace (Atkinson and Mitchell, 1981), and an optimization algorithm to minimize the difference. When the Fisher information is intractable to compute, one could instead utilize the observed Fisher information (Efron and Hinkley, 1978),

$$\left(\hat{\mathcal{I}}_{y|\theta}(\theta)\right)_{i,j} = \left(\frac{\partial}{\partial\theta_i}\log L(y;\theta)\right)\left(\frac{\partial}{\partial\theta_j}\log L(y;\theta)\right),$$

$$\left(\hat{\mathcal{I}}_{y|\theta,z}(\theta; r, b)\right)_{i,j} = \left(\frac{\partial}{\partial\theta_i} \log L_{r,b}(y, z; \theta)\right)\left(\frac{\partial}{\partial\theta_j} \log L_{r,b}(y, z; \theta)\right),$$

with y the observed data.

As the Fisher information matrices depend on θ, we use an adaptation phase to dynamically update r_t, b_t with posterior sample θ_t. Then we stop adaptation after θ_t enters the high posterior density region. This approach is similar to using the frequentist Fisher information evaluated at the maximum-a-posteriori (MAP) estimate, except it does not require the extra optimization steps for computing the MAP. It works well empirically in examples we have considered.

Specifically, we choose r_{t+1} to minimize the difference between $\mathcal{I}_{y|\theta_t}(\theta_t)$ and $\mathbb{E}_{z|\theta_t}\mathcal{I}_{y|\theta_t,z}(\theta_t; r_{t+1}, b_t)$, or $\mathcal{I}^{-1}_{y|\theta_t}(\theta_t)$ and $\mathbb{E}_{z|\theta_t}\mathcal{I}^{-1}_{y|\theta_t,z}(\theta_t; r_{t+1}, b_t)$. Additionally, we set b_{t+1} to minimize the difference between $L_{1,0}(\theta_t; y)$ and $L_{r_{t+1},b_{t+1}}(\theta_t; y)$. Thus, we use r to adjust the conditional variance based on $L_{r,b}$ to match the marginal variance based on L and b to make $L_{r,b}$ close to $L_{1,0}$ in the neighborhood of θ_t. Intuitively, this will make the target distribution closer to the invariant measure of calibrated Gibbs, and correspondingly increase the MH acceptance rate. Some illustrative results about the adaptation are provided in the appendix. The proposal kernel we describe above is *adaptive*; that is, we have a collection of proposal kernels $\mathcal{Q} = \{Q_{r,b}\}_{(r,b)\in\mathbb{R}_+\times\mathbb{R}}$, and we choose a different member of $\mathcal{Q}$ at each iteration to create the proposal. In general, ergodicity of adaptive algorithms requires a diminishing adaptation condition (Roberts and Rosenthal, 2007). For simplicity, we satisfy this condition by stopping adaptation after a tuning phase.

For a concrete illustration, we first return to the first example of probit regression. Letting $\eta_i = x_i\theta$, we obtain

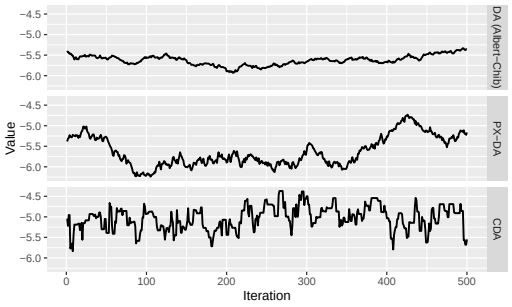
$$\mathcal{I}_{y|\theta}(\theta) = \dot{\eta}\mathrm{diag}\left\{\frac{\phi(\eta_i)^2}{\Phi(\eta_i)(1-\Phi(\eta_i))}\right\}\dot{\eta}', \qquad \mathbb{E}_{z|\theta}\mathcal{I}_{y|\theta,z}(\theta; r, b) = \dot{\eta}R^{-1}\dot{\eta}',$$

where ϕ is the standard normal density, with $\dot{\eta} = X'$. Having $\mathcal{I}_{y|\theta}(\theta) = \mathbb{E}_{z|\theta}\mathcal{I}_{y|\theta,z}(\theta; r, b)$ and $L_{r,b}(\eta_i; y_i) = L(\eta_i; y_i)$ yields

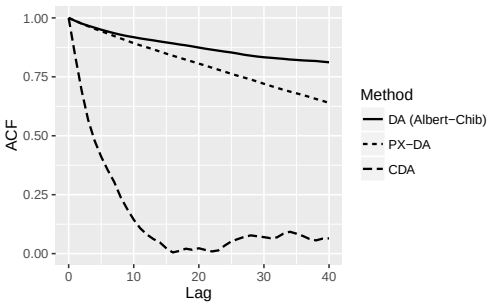
$$\begin{aligned} r_i &= \frac{\Phi(\eta_i)(1-\Phi(\eta_i))}{\phi(\eta_i)^2}, \\ b_i &= \eta_i(\sqrt{r_i} - 1). \end{aligned}$$

For Bernoulli probit models with other forms of η_i, the solution for tuning parameters remains the same.

In simulation, we consider a probit regression with an intercept and two predictors $x_{i,1}, x_{i,2} \sim \mathrm{No}(1, 1)$, with $\theta = (-5, 1, -1)'$, generating $\sum y_i = 20$ among $n = 10,000$. The Albert and Chib (1993) DA algorithm mixes slowly (Figure 2(d) and 2(e)). We also show the results of the parameter expansion algorithm (PX-DA) proposed by Liu and Wu (1999). PX-DA only mildly reduces the correlation, as it does not solve the small step size problem. For CDA, we tuned r and b for 100 steps using the Fisher information, reaching a satisfactory acceptance rate of 0.6 and leading to dramatically better mixing.



(d) Traceplot for the original DA, parameter expanded DA and CDA algorithms.



(e) ACF for original DA, parameter expanded DA and CDA algorithms.

Figure 2: Panel (a) demonstrates in traceplot and panel (b) in autocorrelation the substantial improvement in CDA by correcting the variance mis-match in probit regression with rare event data, compared with the original (Albert and Chib, 1993) and parameter-expanded methods (Liu and Wu, 1999).

For the second example of logistic regression, taking $\eta_i = x_i\theta$, the Fisher information matrices are:

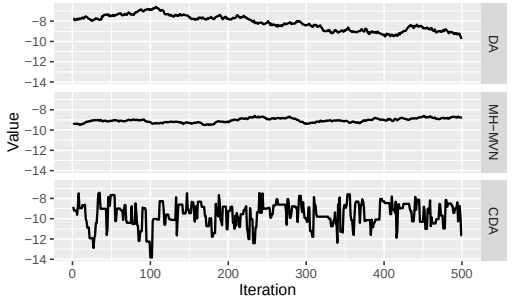
$$\begin{aligned}\mathcal{I}_{y|\theta}(\theta) = &\quad \dot{\eta}\mathrm{diag}\left\{\tfrac{\exp(\eta_i)}{\{1+\exp(\eta_i)\}^2}\right\}\dot{\eta}',\\ \mathbb{E}_{z|\theta}\mathcal{I}_{y|\theta,z}(\theta;r,b) = &\quad \dot{\eta}\mathrm{diag}\left\{\tfrac{r_i}{2|\eta_i+b_i|}\tanh\left(\tfrac{|\eta_i+b_i|}{2}\right)\right\}\dot{\eta}'.\end{aligned}$$

where $\dot{\eta} = X'$. Setting $\mathcal{I}_{y|\theta}(\theta) = \mathbb{E}_{z|\theta}\mathcal{I}_{y|\theta,z}(\theta;r,b)$ and $\{1+\exp(\eta_i)\} = \{1+\exp(\eta_i+b_i)\}^{r_i}$ to locally maximize the M-H acceptance rate yields

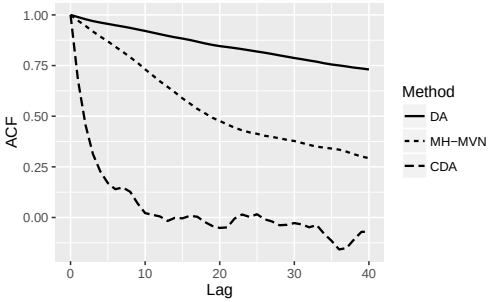
$$\begin{aligned}r_i &= \frac{\exp(\eta_i)}{\{1+\exp(\eta_i)\}^2}2|\eta_i+b_i|/\tanh\left(\frac{|\eta_i+b_i|}{2}\right),\\ b_i &= \log[\{1+\exp(\eta_i)\}^{1/r_i}-1]-\eta_i.\end{aligned}$$

Again, this tuning solution is invariant to the different forms of η under the Bernoulli logistic model.

To illustrate, we use a two parameter intercept-slope model with $x_1 \sim \mathrm{No}(0,1)$ and $\theta = (-9,1)'$. With $n = 10^5$, we obtain rare outcome data with $\sum y_i = 50$. Besides the original DA algorithm (Polson et al., 2013), we also consider an M-H sampler using a multivariate normal proposal $\theta^*|\theta \sim \mathrm{No}(\theta^*|\theta, \mathcal{I}^{-1}(\theta))$ with the inverse Fisher information as the covariance. Similarly, we test an alternative of using DA to generate new θ^*, and scaling to $\theta^{**} = \theta + \alpha(\theta^*-\theta)$, with $\alpha \geq 1$, as an M-H proposal. Both M-H with a normal proposal and with scaled proposal suffer from low acceptance rate, unless $\alpha \approx 1$ in the latter (corresponding to almost no adjustment from DA). For CDA we tuned r and b for 100 steps, reaching an acceptance rate of 0.8. Shown in Figure 3, DA and simple M-H mix slowly, exhibiting strong autocorrelation even at lag 40, while CDA has dramatically better mixing.



(a) Traceplots for DA, CDA and M-H with multivariate normal proposal.



(b) ACF for DA, CDA and M-H with multivariate normal proposal.

Figure 3: Panel (a) demonstrates in traceplot and panel (b) in autocorrelation the substantial improvement of CDA in logistic regression with rare event data, compared with the original DA (Polson et al., 2013) and the M-H algorithm with multivariate normal proposal (MH-MVN).

4. Simulation Study: Scaling to Massive n

As motivated above, two factors are necessary for obtaining usable posterior samples within a practical time: a low computing cost in each iteration and a high effective sample size within a small number of iterations. We first demonstrate that calibration can solve the latter issue.

To start, we consider a simple Bernoulli logistic regression with a common intercept:

$$y_i \overset{iid}{\sim} \text{Bernoulli}\Big(\frac{\exp(\theta)}{1+\exp(\theta)}\Big), \quad i = 1, \ldots, n,$$

with a flat improper prior for θ. As the likelihood is $L(y;\theta) = \exp(\theta \sum y_i)(1+\exp(\theta))^{-n}$, it enjoys efficient computing per iteration that only involves 1 Polya-Gamma latent variable. Alternatively, using calibration parameters (r, b), a proposal can be simulated from

$$\begin{aligned} z &\sim \text{PG}\left(nr, \theta + b\right), \\ \theta^* &\sim \text{No}\left(\frac{\sum y_i - rn/2 - zb}{z}, \frac{1}{z}\right), \end{aligned}$$

and M-H acceptance step as described above using $L_{r,b}(\theta; y) = \exp(\theta+b)^{\sum y_i}\{1+\exp(\theta+b)\}^{-nr}$. To have a proper $L_{r,b}(\theta; y)$, we further require $r \geq (\sum y_i - 1)/n + \epsilon$ with ϵ a small positive constant. Using Fisher information, the parameters are adapted initially for 200 steps, via

$$\begin{aligned} r &= \frac{\exp(\theta)}{\{1+\exp(\theta)\}^2} \Big/ \left(\frac{1}{2|\theta+b|}\tanh\frac{|\theta+b|}{2}\right) \vee \big((\sum y_i - 1)/n + \epsilon\big), \\ b &= \log[\{1+\exp(\theta)\}^{1/r} - 1] - \theta. \end{aligned}$$

To obtain enormous data sample size rare event data, we fixed $\sum y_i = 1$ and increase n from 10^1 to a massive 10^{14}. Figure 4(a) compares the effective sample size per $1,000$ steps

using DA and CDA. Surprisingly, the deterioration of DA shows up as early as $n = 10^2$; its slow-down becomes critical at $n = 10^4$ with effective sample size close to 0. CDA performs exceptionally well, even at massive $n = 10^{14}$ (we stop at 10^{14} as $1/n$ reaches the limit of floating point accuracy).

In more complicated settings, one issue for data augmentation in general is the large number of latent variables to sample in each iteration. A common strategy is to avoid sampling latent variables for every observation by approximating the Markov transition kernel using subsamples (Quiroz et al., 2016; Johndrow et al., 2017). Different from other example algorithms, this approximation changes the invariant measure. Finding a suitable sub-sample size while bounding approximation error requires careful treatment, which is beyond the scope of this article. Instead, our goal is to show sub-sampling alone does not address the burden of low ESS issue; whereas one can trivially couple our proposed CDA strategy with such subsampling to scale DA-MCMC up to enormous data sample sizes. We illustrate such coupling here.

We consider the same two-parameter intercept-slope model in logistic regression as described in the last section, except we now vary data sample size from $n = 10^5$ to 10^8. We simulate Bernoulli outcome $y_i \sim \text{Bernoulli}\big((1 + \exp(-x_i\theta))^{-1}\big)$ based on $x_1 \sim \text{No}(0, 1)$ and $\theta = (-\theta_0, 1)'$. We vary θ_0 and induce $\sum y_i \approx 10$ for each n. We utilize the sub-sampled-Polya-Gamma algorithm described by Johndrow et al. (2017), and apply CDA to calibrate the variance discrepancy. Since y is highly imbalanced in the number of 0 and 1s, we apply biased-sampling by including all data with $y_i = 1$, while sub-sampling 1% of data with $y_i = 0$. Existing work on applying biased subsampling in logistic regression mainly aims to obtain point estimates (King and Zeng, 2001; Wang et al., 2017), in this article we present a simple solution for Bayesian inference.

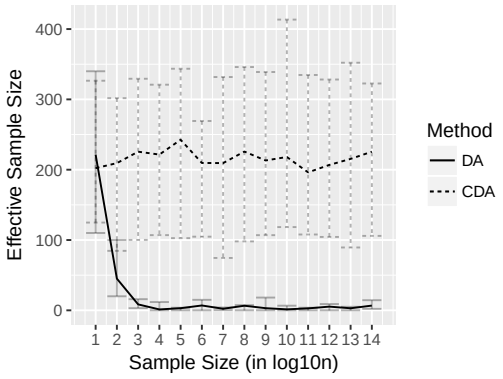
Denoting the set of all data with $y_i = 1$ as V_1 and a random subset with $y_i = 0$ as V_0, it is sensible to keep the likelihood contribution from $y_i = 1$ unchanged, while adjusting the part from $y_i = 0$ via a power of a ratio of $(n - |V_1|)/|V_0|$, leading to an approximate likelihood

$$L(\theta; y) = \prod_{i \in V_1} \frac{\exp(x_i\theta)}{1 + \exp(x_i\theta)} \Big(\prod_{i \in V_0} \frac{1}{1 + \exp(x_i\theta)}\Big)^{\frac{n-|V_1|}{|V_0|}}.$$

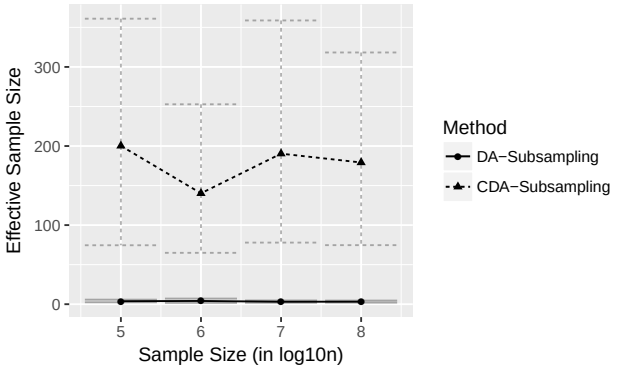
The number of latent variables is reduced to $|V_0| + |V_1|$; since n is still large, the slow mixing would remain and calibration is needed. The algorithmic details and the calibrated form are presented in the appendix.

Figure 4(b) compares the performance of the two approximating algorithms, one combining CDA and sub-sampling, and one using sub-sampling alone. Clearly, only accelerating each step via sub-sampling does not solve the inefficiency of very low effective sample size; while using CDA and sub-sampling together can produce excellent computational performance.

Figure 4: CDA maintains high effective sample size, even when scaling to massive n. Panel(a) shows the performance of DA and CDA when n is scaled up to 10^{14}; Panel(b) shows the performance of CDA and DA, coupled with sub-sampling approximation to reduce the number of sampled latent variables. Only accelerating computing time in each iteration (DA-Subsampling) does not solve the scalability issue.



(a) Effective sample size (with 95% pointwise credible interval) per 1,000 steps with different sample size n from 10 to 10^{14}, using logistic regression model with intercept only.



(b) Effective sample size per (with 95% pointwise credible interval) 1,000 steps with different n from 10^5 to 10^8, in logistic regression with slope and intercept, using subsampling.

5. Co-Browsing Behavior Application

We apply CDA to an online browsing activity dataset. The dataset contains a two-way table of visit count by users who browsed one of 96 client websites of interest, and one of the $n = 59,792$ high-traffic sites during the same browsing session. We refer to visiting more than one site during the same session as co-browsing. For each of the client websites, it is of large commercial interest to find out the high-traffic sites with relatively high co-browsing rates, so that ads can be more effectively placed. For the computational advertising company, it is also useful to understand the co-browsing behavior and predict the traffic pattern of users. We consider two models for these data.

5.1 Hierarchical Binomial Model for Co-Browsing Rates

We initially focus on one client website and analyze co-browsing rates with the high-traffic sites. With the total visit count N_i available for the ith high-traffic site, the count of co-browsing y_i can be considered as the result of a binomial trial. With y_i extremely small relative to N_i (ratio 0.00011 ± 0.00093), the maximum likelihood estimate y_i/N_i can have poor performance. For example, when $y_i = 0$, estimating the rate as exactly 0 is not ideal. Therefore, it is useful to consider a hierarchical model to allow borrowing of information across high-traffic sites:

$$y_i \sim \text{Binomial}\left(N_i, \frac{\exp(\theta_i)}{1+\exp(\theta_i)}\right), \quad \theta_i \overset{iid}{\sim} \text{No}(\theta_0, \sigma_0^2), \quad i = 1 \ldots n$$

$$(\theta_0, \sigma_0^2) \sim \pi(\theta_0, \sigma_0^2)$$

We choose weakly informative priors. Based on expert opinion in quantitative advertising, we use a prior $\theta_0 \sim \text{No}(-12, 49)$ and uniform prior on σ_0^2. Similar to the logistic regression, we calibrate the binomial Polya-Gamma augmentation, leading to the proposal likelihood:

$$L_{r,b}(\theta_i; y_i, N_i, r_i, b_i) = \frac{\exp(\theta_i + b_i)_i^y}{\{1 + \exp(\theta_i + b_i)\}^{N_i r_i}}$$

Conditioned on the latent Polya-Gamma latent variable z_i, each proposal θ_i^* can be sampled from:

$$\begin{aligned} z_i &\sim \text{PG}\left((N_i r_i), \theta_i + b_i\right), \\ \theta_i^* &\sim \text{No}\left(\frac{y_i - r_i N_i/2 - z_i b_i + \theta_0/\sigma_0^2}{z_i + 1/\sigma_0^2}, \frac{1}{z_i + 1/\sigma_0^2}\right), \end{aligned}$$

and accepted or rejected using an M-H step. We further require $r_i \geq (y_i - 1)/N_i + \epsilon$ to have a proper $L_{r,b}(\theta_i; y_i, N_i)$ with ϵ a small constant. Similar to logistic regression, the auxiliary parameters are chosen as

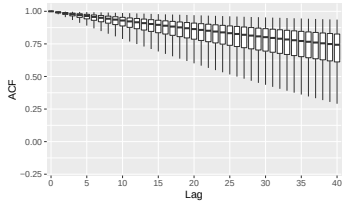
$$\begin{aligned} r_i &= \frac{\exp(\theta_i)}{\{1 + \exp(\theta_i)\}^2} \Big/ \left(\frac{1}{2|\theta_i + b_i|} \tanh \frac{|\theta_i + b_i|}{2}\right) \vee \big((y_i - 1)/N_i + \epsilon\big), \\ b_i &= \log[\{1 + \exp(\theta_i)\}^{1/r_i} - 1] - \theta_i \end{aligned}$$

during adaptation. Since θ_i's are conditionally independent, the calibrated proposal can be individually accepted with high probability for each i. This leads to a high average acceptance of 0.9, despite the high dimensionality of $59,792$ θ_i's. After θ_i's are updated, other parameters are sampled from $\theta_0 \sim \text{No}\big((n/\sigma^2 + 1/49)^{-1}(\sum_i \theta_i/\sigma^2 - 12/49), (n/\sigma^2 + 1/49)^{-1}\big)$, and $\sigma_0^2 \sim \text{Inverse-Gamma}(n/2 - 1, \sum_i (\theta_i - \theta_0)^2/2)$.

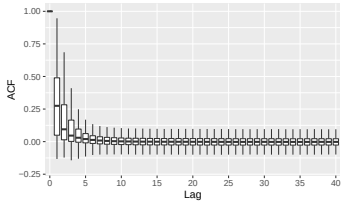
Figure 5 shows the boxplots of the ACFs for all θ_i's. We compare the result with the original DA (Polson et al., 2013) and Hamiltonian Monte Carlo (HMC) provided by the `STAN` software (Carpenter et al., 2016). We run DA for $100,000$ steps, HMC for $2,000$ steps and CDA for $2,000$ steps, so that they have approximately the same effective sample size (calculated with the `CODA` package in `R`). All of the parameters mix poorly in DA; HMC and CDA lead to significant improvement with autocorrelation rapidly decaying to close to zero within 5 lags.

Shown in Table 1, CDA and HMC have very close estimates in posterior means and 95% credible intervals for the parameters, while DA has poor estimates due to critically slow mixing. The difference between HMC and CDA is that, although HMC is slightly more efficient in effective sample size per iteration (T_{eff}/T) for this model, it is much more computationally intensive and generates many fewer iterations than CDA within the same budget of computing time. As the result, CDA has the most efficient computing time per effective sample.

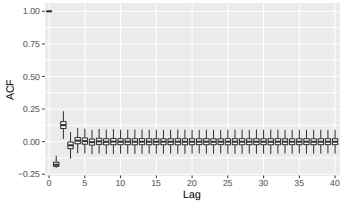
Figure 5: Boxplots of the ACFs show the mixing of the 59,792 parameters in the hierarchical binomial model, for the original DA(Polson et al., 2013), CDA and HMC.



(a) ACFs of the rate parameters θ_i using DA.



(b) ACFs of the rate parameters θ_i using CDA.



(c) ACFs of the rate parameters θ_i using HMC.

	DA	CDA	HMC
$\sum \theta_i/n$	-10.03 (-10.16, -9.87)	-12.05 (-12.09, -12.02)	-12.06 (-12.09, -12.01)
$\sum \theta_i^2/n$	102.25 (98.92, 105.23)	153.04 (152.06, 154.05)	153.17 (152.02, 154.29)
θ_0	-10.03 (-10.17, -9.87)	-12.05 (-12.09,-12.01)	-12.06 (-12.10, -12.01)
σ^2	1.60 (1.36, 1.82)	7.70 (7.49, 7.88)	7.71 (7.51, 7.91)
T_{eff}/T	0.0085 (0.0013 0.0188)	0.5013 (0.1101,1.0084)	0.8404 (0.5149, 1.2470)
Avg Computing Time / T	1.2 sec	1.2 sec	6 sec
Avg Computing Time / T_{eff}	140.4 sec	0.48 sec	1.3 sec

Table 1: Parameter estimates (with 95% credible intervals) and computing speed (ratios among computing time, effective sample sizes T_{eff} and total iterations T) of the DA, CDA and HMC in hierarchical binomial model. CDA provides parameter estimates as accurate as HMC, and is more computationally efficient than HMC.

5.2 Poisson Log-Normal Model for Web Traffic Prediction

The co-browsing on one high-traffic site and one client site is commonly related to the click-through of users from the former to the latter. Therefore, the count of co-browsing is a useful indication of the click-through traffic. For any given client website, predicting the high traffic sites that could generate the most traffic is of high commercial interest. Therefore, we consider a Poisson regression model. We choose the co-browsing count of one client website as the outcome y_i and the log count of the other 95 websites as the predictors $x_{ij} = \log(x_{ij}^* + 1)$ for $i = 1, \ldots, 59792$ and $j = 1, \ldots, 95$. A Gaussian random effect is included to account for over-dispersion relative to the Poisson distribution, leading to a Poisson log-normal regression model:

$$y_i \sim \text{Poisson}\left(\exp(x_i\beta + \tau_i)\right), \quad \tau_i \overset{iid}{\sim} \text{No}(\tau_0, \nu^2), \quad i = 1 \ldots n$$
$$\beta \sim \text{No}(0, I\sigma_\beta^2), \quad \tau_0 \sim \text{No}(0, \sigma_\tau^2) \quad \nu^2 \sim \pi(\nu^2).$$

We assign a weakly informative prior for β and τ_0 with $\sigma_\beta^2 = \sigma_\tau^2 = 100$. For the over-dispersion parameter ν^2, we assign a non-informative uniform prior.

When β and τ are sampled separately, the random effects $\tau = \{\tau_1, \ldots, \tau_n\}$ can cause slow mixing. Instead, we sample β and τ jointly. Using $\tilde{X} = [I_n||X]$ as a $n \times (n+p)$ juxtaposed matrix, and $\eta_i = x_i\beta + \tau_i$ for the linear predictor, the model can be viewed as a linear predictor with $n+p$ coefficients, and $\theta = \{\tau, \beta\}'$ can be sampled jointly in a block. The reason for improved mixing with blocked sampling can be found in Liu (1994a).

We now focus on the mixing behavior of data augmentation. We first review data augmentation for the Poisson log-normal model. Zhou et al. (2012) proposed to treat Poisson(η_i) as the limit of the negative binomial $\mathrm{NB}\big(\lambda, \eta_i/(\lambda+\eta_i)\big)$ with $\lambda \to \infty$, and used moderate $\lambda = 1,000$ for approximation. The limit can be simplified as (omitting constant):

$$L(\eta_i; y_i) = \frac{\exp(y_i\eta_i\}}{\exp\{\exp(\eta_i)\}} = \lim_{\lambda\to\infty} \frac{\exp(y_i\eta_i)}{\{1+\exp(\eta_i)/\lambda\}^{\lambda}}. \tag{11}$$

With finite λ approximation, the posterior can be sampled via Polya-Gamma augmented Gibbs sampling:

$$\begin{aligned} z_i \mid \eta_i &\sim \mathrm{PG}(\lambda, \eta_i - \log\lambda) \quad i = 1\ldots n \\ \theta \mid z, y &\sim \mathrm{No}\Bigg(\Big(\tilde{X}'Z\tilde{X} + \begin{bmatrix} 1/\nu^2\cdot I_n & 0 \\ 0 & 1/\sigma_\beta^2 \cdot I_p \end{bmatrix}\Big)^{-1}\{\tilde{X}'\big(y - \lambda/2 + Z\log\lambda\big) + \begin{bmatrix} \tau_0/\nu^2 1_n \\ 0_p \end{bmatrix}\}, \\ &\qquad \Big(\tilde{X}'Z\tilde{X} + \begin{bmatrix} 1/\nu^2\cdot I_n & 0 \\ 0 & 1/\sigma_\beta^2 \cdot I_p \end{bmatrix}\Big)^{-1}\Bigg), \end{aligned}$$

where $Z = \mathrm{diag}\{z_1, \ldots, z_n\}$, $1_n = \{1, \ldots 1\}'$ and $0_p = \{0, \ldots 0\}'$.

However, this approximation-based data augmentation is inherently problematic. For example, setting $\lambda = 1,000$ leads to large approximation error. As in (11), the approximating denominator has $(1+\exp(\eta_i)/\lambda)^{\lambda} = \exp\{\exp(\eta_i) + \mathcal{O}(\exp(2\eta_i)/\lambda)\}$; for moderately large $\eta_i \approx 10$, λ needs to be at least 10^9 to make $\exp(2\eta_i)/\lambda$ close to 0. This large error cannot be corrected with an additional M-H step, since the acceptance rate would be too low. On the other hand, it is not practical to use a large λ in a Gibbs sampler, as it would create extremely large z_i (associated with small conditional covariance for θ), resulting in slow mixing.

We use CDA to solve this dilemma. We first choose a very large λ (10^9) to control the approximation error, then use a small fractional r_i multiplying to λ for calibration. This leads to a proposal likelihood similar to the logistic CDA:

$$L_{r,b}(x_i\theta; y_i) = \frac{\exp(\eta_i - \log\lambda + b_i)^{y_i}}{\{1+\exp(\eta_i - \log\lambda + b_i)\}^{r_i\lambda}},$$

with $r_i \geq (y_i - 1)/\lambda + \epsilon$ for proper likelihood, and proposal update rule:

$$\begin{aligned} z_i &\sim \mathrm{PG}(r_i\lambda, \eta_i - \log\lambda + b_i) \quad i = 1\ldots n \\ \theta^* &\sim \mathrm{No}\Bigg(\Big(\tilde{X}'Z\tilde{X} + \begin{bmatrix} 1/\nu^2\cdot I_n & 0 \\ 0 & 1/\sigma_\beta^2 \cdot I_p \end{bmatrix}\Big)^{-1}\Big\{\tilde{X}'\big(y - r\lambda/2 + Z\log(\lambda - b)\big) + \begin{bmatrix} \tau_0/\nu^2 1_n \\ 0_p \end{bmatrix}\Big\}, \\ &\qquad \Big(\tilde{X}'Z\tilde{X} + \begin{bmatrix} 1/\nu^2\cdot I_n & 0 \\ 0 & 1/\sigma_\beta^2 \cdot I_p \end{bmatrix}\Big)^{-1}\Bigg) \end{aligned}$$

Letting $\eta_i^* = \tilde{X}\theta^*$, the proposal is accepted with probability (based on Poisson density and the approximation $L_{r,b}(x_i\theta; y_i)$):

$$1 \wedge \prod_i \frac{\exp\{\exp(\eta_i)\}}{\exp\{\exp(\eta_i^*)\}} \frac{\{1+\exp(\eta_i^* - \log\lambda + b_i)\}^{r_i\lambda}}{\{1+\exp(\eta_i - \log\lambda + b_i)\}^{r_i\lambda}}.$$

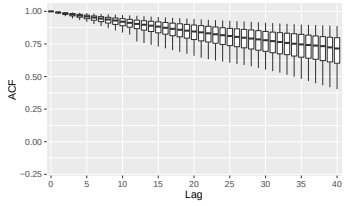
During the tuning, we set

$$r_i = \tau_i \exp(\eta_i) / \left(\frac{\lambda}{2|\eta_i + b_i - \log\lambda|} \tanh \frac{|\eta_i + b_i - \log\lambda|}{2} \right) \vee \big((y_i - 1)/\lambda + \epsilon\big),$$
$$b_i = \log[\exp\{\exp(\eta_i - \log\lambda - \log r_i)\} - 1] - \eta_i + \log\lambda.$$

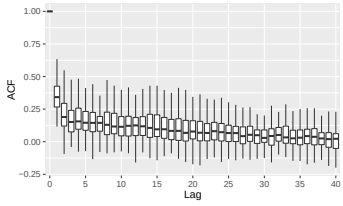
After θ is updated, the other parameters can be sampled via $\tau_0 \sim \text{No}\left((n/\nu^2 + 1/\sigma_\tau^2)^{-1} \sum_i \tau_i/\nu^2, (n/\nu^2 + 1/\sigma_\tau^2)^-\right.$ and $\nu^2 \sim \text{Inverse-Gamma}(n/2 - 1, \sum_i(\tau_i - \tau_0)^2/2)$.

We ran the basic DA with $\lambda = 1,000$ approximation, CDA with $\lambda = 10^9$ and HMC. We ran DA for $200,000$ steps, CDA for $2,000$ steps and HMC for $20,000$ steps so that they have approximately the same effective sample size. For CDA, we used the first $1,000$ steps for adapting r and b. Figure 6 shows the mixing of DA, CDA and HMC. Even with small $\lambda = 1,000$ in DA, all of the parameters mix poorly; HMC seemed to be affected by the presence of random effects, and most of parameters remain highly correlated within 40 lags; CDA substantially improves the mixing. Table 2 compares all three algorithms. CDA has the most efficient computing time per effective sample, and is about $30 - 300$ times more efficient than the other two algorithms.

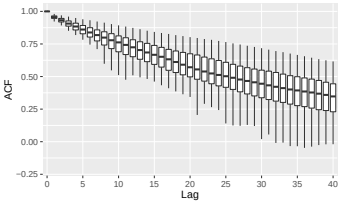
Figure 6: CDA significantly improves the mixing of the parameters in the Poisson log-normal.



(a) Autocorrelation of the parameters from DA.



(b) Autocorrelation of the parameters from CDA.



(c) Autocorrelation of the parameters from HMC.

To evaluate the prediction performance, we use another co-browsing count table for the same high traffic and client sites, collected during a different time period. We use the high traffic co-browsing count $x_{ij}^{\dagger *}$ and their log transform $x_{ij}^{\dagger} = \log(x_{ij}^{\dagger *} + 1)$ for the $j = 1, \ldots, 95$ clients to predict the count for the client of interest $y_i^{\dagger}$, over the high traffic site $i = 1, \ldots, 59792$. We carry out prediction using $\hat{y}_i^{\dagger} = \mathbb{E}_{\beta,\tau|y,x} y_i^{\dagger} = \mathbb{E}_{\beta,\tau|y,x} \exp(x_i^{\dagger}\beta + \tau_i)$ on the client site. The expectation is taken over posterior sample $\beta, \tau \mid y, x$ with training set $\{y, x\}$ discussed above. Cross-validation root-mean-squared error $\left(\sum_i (\hat{y}_i^{\dagger} - y_i^{\dagger})^2/n\right)^{1/2}$ between the prediction and actual count $y_i^{\dagger}$'s is computed. Shown in Table 2, slow mixing

in DA and HMC cause poor estimation of the parameters and high prediction error, while CDA has significantly lower error.

	DA	CDA	HMC
$\sum \beta_j/95$	0.072 (0.071, 0.075)	-0.041 (-0.042, -0.038)	-0.010 (-0.042, -0.037)
$\sum \beta_j^2/95$	0.0034 (0.0033, 0.0035)	0.231 (0.219 0.244)	0.232 (0.216 0.244)
$\sum \tau_i/n$	-0.405 (-0.642, -0.155)	-1.292 (-2.351, -0.446)	-1.297 (-2.354, -0.451)
$\sum \tau_i^2/n$	1.126 (0.968, 1.339)	3.608 (0.696, 7.928)	3.589 (0.678, 8.011)
Prediction RMSE	33.21	8.52	13.18
T_{eff}/T	0.0037 (0.0011 0.0096)	0.3348 (0.0279, 0.699)	0.0173 (0.0065, 0.0655)
Avg Computing Time / T	1.3 sec	1.3 sec	56 sec
Avg Computing Time / T_{eff}	346.4 sec	11.5 sec	3240.6 sec

Table 2: Parameter estimates, prediction error and computing speed of the DA, CDA and HMC in Poisson regression model.

6. Discussion

Data augmentation (DA) is a technique routinely used to enable implementation of simple Gibbs samplers, avoiding the need for expensive and complex tuning of Metropolis-Hastings algorithms. Despite the convenience, DA can slow down mixing when the conditional posterior variance given the augmented data is substantially smaller than the marginal variance. When the data sample size is massive, this problem arises when the rates of convergence of the augmented and marginal posterior differ, leading to critical mixing problems. There is a very rich literature on strategies for improving mixing rates of Gibbs samplers, with centered or non-centered re-parameterizations (Papaspiliopoulos et al., 2007) and parameter-expansion (Liu and Wu, 1999) leading to some improvements. However, existing approaches do not solve large sample mixing problems in not addressing the fundamental rate mismatch issue.

To tackle this problem, we propose to calibrate data augmentation and use a parameter to directly adjust the conditional variance (which is associated with step size). CDA adds a little cost due to the likelihood evaluation, which is often negligible as compared to the random number generation. In this article, we demonstrate that calibration is generally applicable when $\theta \mid z$ belongs to the location-scale family. We expect it to be extensible to any conditional distribution with a variance or scale.

As both CDA and HMC involve M-H steps, we draw some further comparison between the two. Both methods rely on finding a good proposal by searching a region far from the current state. One key difference lies in the computing efficiency. Although HMC is more generally applicable beyond data augmentation, it is computationally intensive since Hamiltonian dynamics often requires multiple numeric steps. CDA only requires one step of calibrated Gibbs sampling, which is often much more efficient leveraging on existing data augmentation algorithms.

Appendix A. Appendix

A.1 Proof of Remark 1

Proof Since $Q_{r,b}(\theta;\theta')$ is the θ marginal of a Gibbs transition kernel, and Gibbs is reversible on its margins, we have

$$Q(\theta;\theta')\Pi_{r,b}(\theta) = Q(\theta';\theta)\Pi_{r,b}(\theta),$$

and so

$$\begin{aligned}\frac{L(\theta';y)\Pi^0(\theta')Q(\theta;\theta')}{L(\theta;y)\Pi^0(\theta)Q(\theta';\theta)} &= \frac{L(\theta';y)\Pi^0(\theta')L_{r,b}(\theta;y)\Pi^0(\theta)}{L(\theta;y)\Pi^0(\theta)L_{r,b}(\theta';y)\Pi^0(\theta')} \\ &= \frac{L(\theta';y)L_{r,b}(\theta;y)}{L(\theta;y)L_{r,b}(\theta';y)}.\end{aligned}$$

■

A.2 Proof of Remark 2

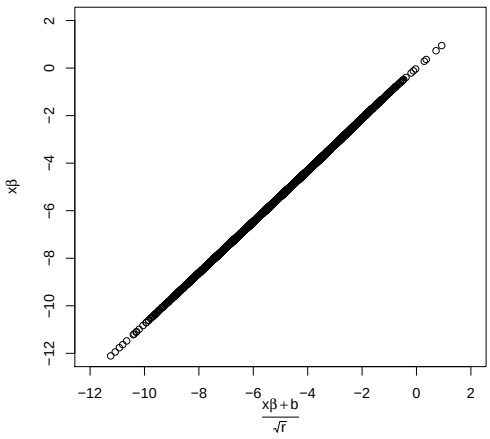
Proof For any r, b, the conditionals $\Pi_{r,b}(z \mid \theta)$ and $\Pi_{r,b}(\theta \mid z)$ are well-defined for all $z \in \mathcal{Z}, \theta \in \Theta$, and therefore the Gibbs transition kernel $K_{r,b}((\theta, z);\cdot)$ and corresponding marginal kernels $Q_{r,b}(\theta;\cdot)$ are well-defined. Moreover, for any $(z,\theta) \in \mathcal{Z} \times \Theta$, we have $\mathbb{P}[(\theta', z') \in A \mid (\theta, z)] > 0$ by assumption. Thus $K_{r,b}$ is aperiodic and $\Pi_{r,b}$-irreducible (see the discussion following Corollary 1 in Roberts and Smith (1994)).

$Q_{r,b}(\theta';\theta)$ is aperiodic and $\Pi_{r,b}(\theta)$-irreducible, since it is the θ marginal transition kernel induced by $K_{r,b}((\theta, z);\cdot)$. Thus, it is also $\Pi(\theta)$-irreducible so long as $\Pi \gg \Pi_{r,b}$, where for two measure μ, ν, $\mu \gg \nu$ indicates absolute continuity. Since $\Pi, \Pi_{r,b}$ have densities with respect to Lebesgue measure, $\Pi_{r,b}$-irreducibility implies Π irreducibility. Moreover, $Q(\theta;\theta') > 0$ for all $\theta \in \Theta$. Thus, by (Roberts and Smith, 1994, Theorem 3), CDA M-H is Π-irreducible and aperiodic. ■

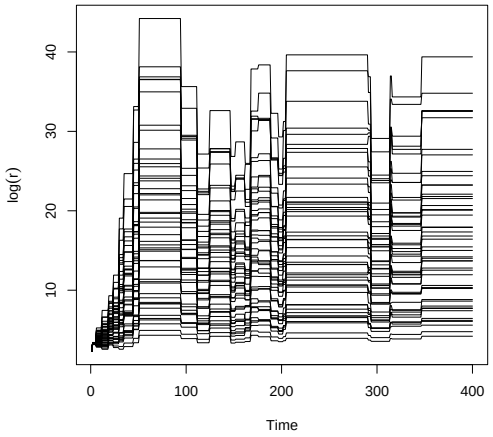
A.3 Diagnostics of Adaptation

We adapt the tuning parameters (r, b) by locally minimizing difference between two Fisher information matrices and optimizing acceptance rate near $\hat{\theta}_{MAP}$ during adaptation. Updating b as a function of θ yields $L_{r,b}(\theta;y) = L(\theta;y)$ exactly; after adaptation, for fixed (r, b), $L_{r,b}(\theta;y)$ is close to $L(\theta;y)$ for θ in the neighborhood around $\hat{\theta}_{MAP}$. To show this empirically, consider the probit Bernoulli regression example. As the $L(\theta;y)$ and $L_{r,b}(\theta;y)$ are parameterized by $\Phi(x_i\beta)$ and $\Phi((x_i\beta + b_i)/\sqrt{r_i})$, Figure 7(a) compares the posterior values of $x_i\beta$ against $(x_i\beta + b_i)/\sqrt{r_i}$. Clearly, the two are very close with a RMSE of 0.23. Figure 7(b) shows the trace of the adaptation of r during the initial 400 iterations; r quickly rises from 1 to the roughly appropriate scale during the initial 50 steps. The values of (r, b) are fixed afterwards to ensure ergodicity.

Figure 7: Diagnostics plot of the adaptation for probit regression.



(a) Posterior sample of $x_i\beta$ and its corresponding transform $(x_i\beta + b_i)/\sqrt{r_i}$ in probit CDA.



(b) Trace of the adaptation of r, shown on log scale.

A.4 Calibrated Polya-Gamma Algorithm with Sub-sampling

Adapting based on Johndrow et al. (2017), we first randomly sample a subset of indices V of size $|V|$. This algorithm generates proposals from

$$
\begin{aligned}
&V = V_1 \cup V_0, \quad V_1 = \{i \in \{1, \ldots, n\} : y_i = 1\}, \quad V_0 \sim \text{Subset}(|V|, \{i \in \{1, \ldots, n\} : y_i = 0\}) \\
&z_i \sim \text{PG}(k_i r_i, |x_i\theta + b_i|) \quad i \in V, \\
&\theta^* \sim \text{No}\left((X_V' Z_V X_V)^{-1} X_V'(y_V - k_V r_V/2 - Z_V b_V), (X_V' Z_V X_V)^{-1}\right),
\end{aligned}
$$

where subscript $._V$ indicates the sub-matrix or sub-vector corresponding to the sub-sample; $k_i = 1$ if $y_i = 1$, and $k_i = (n - |V_1|)/|V_0|$. We accept θ^* in M-H step using calibrated likelihood

$$
L_{r,b}(\theta; y) = \prod_{i \in V_1} \frac{\exp(x_i\theta + b_i)}{\{1 + \exp(x_i\theta + b_i)\}^{r_i}} \Big(\prod_{i \in V_0} \frac{1}{\{1 + \exp(x_i\theta + b_i)\}^{r_i}}\Big)^{\frac{n - |V_1|}{|V_0|}},
$$

with target approximate likelihood $L_{1,0}(\theta; y)$. Using Fisher information, the parameters are adapted initially for 200 steps, via

$$
\begin{aligned}
r_i &= \frac{\exp(x_i\theta)}{\{1 + \exp(x_i\theta)\}^2} \Big/ \left(\frac{1}{2|x_i\theta + b_i|} \tanh \frac{|x_i\theta + b_i|}{2}\right) \vee \big((y_i - 1)/k_i + \epsilon\big) \\
b_i &= \log[\{1 + \exp(x_i\theta + b_i)\}^{1/r_i} - 1] - x_i\theta.
\end{aligned}
$$

References

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

Colin Atkinson and Ann FS Mitchell. Rao's distance measure. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 345–365, 1981.

Bob Carpenter, Andrew Gelman, Matt Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Michael A Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. STAN: a probabilistic programming language. *Journal of Statistical Software*, 2016.

Patrick R Conrad, Youssef M Marzouk, Natesh S Pillai, and Aaron Smith. Accelerating asymptotically exact MCMC for computationally intensive models via local approximations. *Journal of the American Statistical Association*, (1591–1607), 2015.

Bradley Efron and David V Hinkley. Assessing the accuracy of the maximum likelihood estimator: observed versus expected Fisher information. *Biometrika*, 65(3):457–483, 1978.

James E Johndrow, Aaron Smith, Natesh Pillai, and David B Dunson. Inefficiency of data augmentation for large sample imbalanced data. *arXiv preprint arXiv:1605.05798*, 2016.

James E Johndrow, Jonathan C Mattingly, Sayan Mukherjee, and David B Dunson. Optimal approximating Markov chains for Bayesian inference. *arXiv preprint arXiv:1508.03387*, 2017.

Gary King and Langche Zeng. Logistic regression in rare events data. *Political Analysis*, 9 (2):137–163, 2001.

Jun S Liu. The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427): 958–966, 1994a.

Jun S Liu. The fraction of missing information and convergence rate for data augmentation. *Computing Science and Statistics*, pages 490–490, 1994b.

Jun S Liu and Ying Nian Wu. Parameter expansion for data augmentation. *Journal of the American Statistical Association*, 94(448):1264–1274, 1999.

Dougal Maclaurin and Ryan P Adams. Firefly Monte Carlo: exact MCMC with subsets of data. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 543–552, 2015.

Xiao-Li Meng and David A Van Dyk. Seeking efficient data augmentation schemes via conditional and marginal augmentation. *Biometrika*, 86(2):301–320, 1999.

Stanislav Minsker, Sanvesh Srivastava, Lizhen Lin, and David B Dunson. Robust and scalable Bayes via a median of subset posterior measures. *arXiv preprint arXiv:1403.2660*, 2014.

EWT Ngai, Yong Hu, YH Wong, Yijun Chen, and Xin Sun. The application of data mining techniques in financial fraud detection: a classification framework and an academic review of literature. *Decision Support Systems*, 50(3):559–569, 2011.

Omiros Papaspiliopoulos, Gareth O Roberts, and Martin Sköld. A general framework for the parametrization of hierarchical models. *Statistical Science*, pages 59–73, 2007.

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

Matias Quiroz, Mattias Villani, and Robert Kohn. Exact subsampling MCMC. *arXiv preprint arXiv:1603.08232*, 2016.

Gareth O Roberts and Jeffrey S Rosenthal. Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *Journal of Applied Probability*, pages 458–475, 2007.

Gareth O Roberts and Adrian FM Smith. Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. *Stochastic Processes and Their Applications*, 49(2):207–216, 1994.

Donald B Rubin. *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons, 2004.

Sanvesh Srivastava, Volkan Cevher, Quoc Tran-Dinh, and David B Dunson. WASP: scalable Bayes via Barycenters of subset posteriors. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 912–920, 2015.

Martin A Tanner and Wing Hung Wong. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540, 1987.

Minh-Ngoc Tran, Michael K Pitt, and Robert Kohn. Adaptive Metropolis–Hastings sampling using reversible dependent mixture proposals. *Statistics and Computing*, 26(1-2): 361–381, 2016.

Jon Wakefield. Disease mapping and spatial regression with count data. *Biostatistics*, 8(2): 158–183, 2007.

HaiYing Wang, Rong Zhu, and Ping Ma. Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association*, (in press), 2017.

Xuerui Wang, Wei Li, Ying Cui, Ruofei Zhang, and Jianchang Mao. Click-through rate estimation for rare events in online advertising. *Online Multimedia Advertising: Techniques and Technologies*, pages 1–12, 2010.

Mingyuan Zhou, Lingbo Li, David B Dunson, and Lawrence Carin. Lognormal and Gamma mixed negative Binomial regression. In *Proceedings of the International Conference on Machine Learning*, volume 2012, page 1343, 2012.