Bayesian Inference for Logistic Models Using Pólya–Gamma Latent Variables

Nicholas G. Polson, James G. Scott & Jesse Windle

Bayesian Inference for Logistic Models Using Pólya–Gamma Latent Variables

Nicholas G. POLSON, James G. SCOTT, and Jesse WINDLE

We propose a new data-augmentation strategy for fully Bayesian inference in models with binomial likelihoods. The approach appeals to a new class of Pólya–Gamma distributions, which are constructed in detail. A variety of examples are presented to show the versatility of the method, including logistic regression, negative binomial regression, nonlinear mixed-effect models, and spatial models for count data. In each case, our data-augmentation strategy leads to simple, effective methods for posterior inference that (1) circumvent the need for analytic approximations, numerical integration, or Metropolis–Hastings; and (2) outperform other known data-augmentation strategies, both in ease of use and in computational efficiency. All methods, including an efficient sampler for the Pólya–Gamma distribution, are implemented in the R package `BayesLogit`. Supplementary materials for this article are available online.

KEY WORDS: Bayesian methods; Data augmentation; Logistic regression; Negative binomial regression; Pólya–Gamma distribution.

1. INTRODUCTION

Bayesian inference for the logistic regression model has long been recognized as a hard problem, due to the analytically inconvenient form of the model's likelihood function. By comparison, Bayesian inference for the probit model is much easier, owing to the simple latent-variable method of Albert and Chib (1993) for posterior sampling.

In the two decades since the work of Albert and Chib (1993) on the probit model, there have been many attempts to apply the same missing-data strategy to the logit model (e.g., Holmes and Held 2006; Frühwirth-Schnatter and Frühwirth 2010; Gramacy and Polson 2012). The results have been mixed. Certainly, many of these approaches have been used successfully in applied work. Yet they all involve data-augmentation algorithms that are either approximate, or are significantly more complicated than the Albert/Chib method, as they involve multiple layers of latent variables. Perhaps as a result, the Bayesian treatment of the logit model has not seen widespread adoption by nonstatisticians in the way that, for example, the Bayesian probit model is used extensively in both political science and market research (e.g., Rossi, Allenby, and McCulloch 2005; Jackman 2009). The lack of a standard computational approach also makes it more difficult to use the logit link in the kind of complex hierarchical models that have become routine in Bayesian statistics.

In this article, we present a new data-augmentation algorithm for Bayesian logistic regression. Although our method involves a different missing-data mechanism from that of Albert and Chib (1993), it is nonetheless a direct analog of their construction, in that it is both exact and simple. Moreover, because our method works for any binomial likelihood parameterized by log odds, it

leads to an equally painless Bayesian treatment of the negative-binomial model for overdispersed count data.

This approach appeals to a new family of Pólya–Gamma distributions, described briefly here and constructed in detail in Section 2.

Definition 1. A random variable X has a Pólya–Gamma distribution with parameters $b > 0$ and $c \in \mathcal{R}$, denoted as $X \sim \mathrm{PG}(b, c)$, if

$$X \stackrel{D}{=} \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{g_k}{(k - 1/2)^2 + c^2/(4\pi^2)}, \tag{1}$$

where the $g_k \sim \mathrm{Ga}(b, 1)$ are independent gamma random variables, and where $\stackrel{D}{=}$ indicates equality in distribution.

Our main result (shown in Theorem 1) is that binomial likelihoods parameterized by log-odds can be represented as mixtures of Gaussians with respect to a Pólya–Gamma distribution. The fundamental integral identity at the heart of our approach is that, for $b > 0$,

$$\frac{(e^{\psi})^a}{(1 + e^{\psi})^b} = 2^{-b} e^{\kappa \psi} \int_0^{\infty} e^{-\omega \psi^2/2} \, p(\omega) \, d\omega, \tag{2}$$

where $\kappa = a - b/2$ and $\omega \sim \mathrm{PG}(b, 0)$. When $\psi = x^T \boldsymbol{\beta}$ is a linear function of predictors, the integrand is the kernel of a Gaussian likelihood in $\boldsymbol{\beta}$. Moreover, as we will show below, the implied conditional distribution for ω, given ψ, is also a Pólya–Gamma distribution. This suggests a simple strategy for Gibbs sampling across a wide class of binomial models: Gaussian draws for the main parameters, and Pólya–Gamma draws for a single layer of latent variables.

The success of this strategy depends upon the existence of a simple, effective way to simulate Pólya–Gamma random variables. The sum-of-gammas representation in Equation (1) initially seems daunting, and suggests only a naïve finite approximation. But we describe a fast, exact Pólya–Gamma simulation method that avoids the difficulties that can result from truncating an infinite sum. The method, which is implemented in the

Nicholas G. Polson is Professor of Statistics and Econometrics, University of Chicago Booth School of Business, 1100 East 57th Street, Chicago, IL 60637 (E-mail: *ngp@chicagobooth.edu*). James G. Scott is Assistant Professor of Statistics, University of Texas at Austin, 2110 Speedway, Stop B6500, Austin, TX 78712 (E-mail: *james.scott@mccombs.utexas.edu*). Jesse Windle is Postdoctoral Associate, Department of Statistical Science, Duke University, Box 90251, Durham, NC 27708 (E-mail: *jesse.windle@gmail.com*). The authors thank Hee Min Choi and Jim Hobert for sharing an early draft of their article on the uniform ergodicity of the Pólya–Gamma Gibbs sampler. They also thank two anonymous referees, the associate editor, and the editor of the *Journal of the American Statistical Association*, whose many insights and helpful suggestions have improved the article. The second author acknowledges the support of a CAREER grant from the U.S. National Science Foundation (DMS-1255187).

R package `BayesLogit` (Windle, Polson, and Scott 2013a), is an accept/reject sampler based on the alternating-series method of Devroye (1986). For the basic PG(1, c) case, the sampler is very efficient: it requires only exponential and inverse-Gaussian draws, and the probability of accepting a proposed draw is uniformly bounded below at 0.99919. The method is also fully automatic, with no tuning needed to get optimal performance. It is therefore sufficiently fast and reliable to be used as a black-box sampling routine in complex hierarchical models involving the logit link.

Many previous approaches have been proposed for estimating Bayesian logistic regression models. This includes the Metropolis–Hastings (MH) method, along with many other latent-variable schemes that facilitate Gibbs sampling, all described below. Thus, a major aim of our article is to demonstrate the efficiency of the Pólya–Gamma approach versus these alternatives across a wide range of circumstances. We present evidence in support of two claims.

1. In simple logit models with abundant data and no hierarchical structure, the Pólya–Gamma method is a close second to the independence MH sampler, as long as the MH proposal distribution is chosen carefully.
2. In virtually all other cases, the Pólya–Gamma method is most efficient.

The one exception we have encountered to the second claim is the case of a negative-binomial regression model with many counts per observation and with no hierarchical structure in the prior. Here, the effective sample size of the Pólya–Gamma method remains the best, but its effective sampling *rate* suffers. As we describe below, this happens because our present method for sampling PG(n, c) is to sum n independent draws from PG(1, c); with large counts, this becomes a bottleneck. In such cases, the method of Frühwirth-Schnatter et al. (2009) provides a fast approximation, at the cost of introducing a more complex latent-variable structure.

This caveat notwithstanding, the Pólya–Gamma scheme offers real advantages, both in speed and simplicity, across a wide variety of structured Bayesian models for binary and count data. In general, the more complex the model, and the more time that one must spend sampling its main parameters, the larger will be the efficiency advantage of the new method. The difference is especially large for the Gaussian-process spatial models we consider below, which require expensive matrix operations. We have also made progress in improving the speed of the Pólya–Gamma sampler for large shape parameters, beyond the method described in Section 4. These modifications lead to better performance in negative-binomial models with large counts. They are detailed in Windle, Polson, and Scott (2013b) and have been incorporated into the latest version of our R package (Windle, Polson, and Scott 2013a).

Furthermore, in a recent article based on an early technical report of our method, Choi and Hobert (2013) have proven that the Pólya–Gamma Gibbs sampler for Bayesian logistic regression is uniformly ergodic. This result has important practical consequences; most notably, it guarantees the existence of a central limit theorem for Monte Carlo averages of posterior draws. We are aware of no similar result for any other MCMC-based approach to the Bayesian logit model. Together with the numerical

evidence we present here, this provides a strong reason to favor the routine use of the Pólya–Gamma method.

The article proceeds as follows. The Pólya–Gamma distribution is constructed in Section 2, and used to derive a data-augmentation scheme for binomial likelihoods in Section 3. Section 4 describes a method for simulating from the Pólya–Gamma distribution, which we have implemented as a stand-alone sampler in the `BayesLogit` R package. Section 5 presents the results of an extensive benchmarking study comparing the efficiency of our method to other data-augmentation schemes. Section 6 concludes with a discussion of some open issues related to our proposal. Many further details of the sampling algorithm and our empirical study of its efficiency are deferred to a technical supplement.

2. THE PÓLYA–GAMMA DISTRIBUTION

2.1 The Case PG(b, 0)

The key step in our approach is the construction of the Pólya–Gamma distribution. We now describe this new family, deferring our method for simulating PG random variates to Section 4.

The Pólya–Gamma family of distributions, denoted as PG(b, c), is a subset of the class of infinite convolutions of gamma distributions. We first focus on the PG(1, 0) case, which is a carefully chosen element of the class of infinite convolutions of exponentials, also known as Pólya distributions (Barndorff-Nielsen, Kent, and Sorensen 1982). The PG(1, 0) distribution has Laplace transform $\cosh^{-1}(\sqrt{t/2})$. Using this as a starting point, one may define the random variable $\omega \sim$ PG(b, 0), $b > 0$, as the infinite convolution of gamma distributions (hence the name Pólya–Gamma) that has Laplace transform

$$
\mathbb{E}\{\exp(-\omega t)\} = \prod_{k=1}^{\infty} \left(1 + \frac{t}{2\pi^2(k-1/2)^2}\right)^{-b}
$$
$$
= \frac{1}{\cosh^b(\sqrt{t/2})}. \tag{3}
$$

The last equality is a consequence of the Weierstrass factorization theorem. By inverting the Laplace transform, one finds that if $\omega \sim$ PG(b, 0), then it is equal in distribution to an infinite sum of gammas:

$$
\omega \stackrel{D}{=} \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{g_k}{(k-1/2)^2},
$$

where the $g_k \sim$ Ga(b, 1) are mutually independent.

The PG(b, 0) class of distributions is closely related to a subset of distributions that are surveyed by Biane, Pitman, and Yor (2001). This family of distributions, which we denote by $J^*(b)$, $b > 0$, has close connections with the Jacobi Theta and Riemann Zeta functions, and with Brownian excursions. Its Laplace transform is

$$
\mathbb{E}\{e^{-tJ^*(b)}\} = \cosh^{-b}(\sqrt{2t}), \tag{4}
$$

implying that PG(b, 0) $\stackrel{D}{=} J^*(b)/4$.

2.2 The General PG(b, c) Class

The general PG(b, c) class arises through an exponential tilting of the PG(b, 0) density, much in the same way that a Gaussian likelihood combines with a Gamma prior for a precision. Specifically, a PG(b, c) random variable has the probability density function

$$p(x \mid b, c) = \frac{\exp\left(-\frac{c^2}{2} x\right) p(x \mid b, 0)}{\mathbb{E}\left\{\exp x\left(-\frac{c^2}{2}\omega\right)\right\}}, \tag{5}$$

where $p(x \mid b, 0)$ is the density of an $\omega \sim \text{PG}(b, 0)$ random variable. The expectation in the denominator is taken with respect to the PG(b, 0) distribution; it is thus $\cosh^{-b}(c/2)$ by Equation (3), ensuring that $p(x \mid b, c)$ is a valid density.

The Laplace transform of a $\omega \sim \text{PG}(b, c)$ distribution may be calculated by appealing to the Weierstrass factorization theorem again:

$$\mathbb{E}\{\exp(-\omega t)\}$$
$$= \frac{\cosh^b\left(\frac{c}{2}\right)}{\cosh^b\left(\sqrt{\frac{c^2/2+t}{2}}\right)} \tag{6}$$
$$= \prod_{k=1}^{\infty} \left(\frac{1 + \frac{c^2/2}{2(k-1/2)^2 \pi^2}}{1 + \frac{c^2/2+t}{2(k-1/2)^2 \pi^2}}\right)^b$$
$$= \prod_{k=1}^{\infty} \left(1 + d_k^{-1} t\right)^{-b}, \quad \text{where } d_k = 2\left(k - \frac{1}{2}\right)^2 \pi^2 + c^2/2.$$

Each term in the product is recognizable as the Laplace transform of a gamma distribution. We can therefore write a $\omega \sim \text{PG}(b, c)$ as an infinite convolution of gamma distributions,

$$\omega \overset{D}{=} \sum_{k=1}^{\infty} \frac{\text{Ga}(b, 1)}{d_k} = \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{\text{Ga}(b, 1)}{\left(k - \frac{1}{2}\right)^2 + c^2/(4\pi^2)},$$

which is the form given in Definition 1.

2.3 Further Properties

The density of a Pólya–Gamma random variable can be expressed as an alternating-sign sum of inverse-Gaussian densities. This fact plays a crucial role in our method for simulating Pólya–Gamma draws. From the characterization of $J^*(b)$ density given by Biane, Pitman, and Yor (2001), we know that the PG(b, 0) distribution has density

$$f(x \mid b, 0) = \frac{2^{b-1}}{\Gamma(b)}$$
$$\times \sum_{n=0}^{\infty} (-1)^n \frac{\Gamma(n+b)}{\Gamma(n+1)} \frac{(2n+b)}{\sqrt{2\pi x^3}} e^{-\frac{(2n+b)^2}{8x}}.$$

The density of PG(b, z) distribution is then computed by an exponential tilt and a renormalization:

$$f(x \mid b, c) = \{\cosh^b(c/2)\} \frac{2^{b-1}}{\Gamma(b)}$$
$$\times \sum_{n=0}^{\infty} (-1)^n \frac{\Gamma(n+b)}{\Gamma(n+1)} \frac{(2n+b)}{\sqrt{2\pi x^3}} e^{-\frac{(2n+b)^2}{8x} - \frac{c^2}{2} x}.$$

Note that the normalizing constant is known directly from the Laplace transform of a PG(b, 0) random variable.

A further useful fact is that all finite moments of a Pólya–Gamma random variable are available in a closed form. In particular, the expectation may be calculated directly. This allows the Pólya–Gamma scheme to be used in EM algorithms, where the latent ω's will form a set of complete-data sufficient statistics for the main parameter. We arrive at this result by appealing to the Laplace transform of $\omega \sim \text{PG}(b, c)$. Differentiating Equation (6) with respect to t, negating, and evaluating at zero yields

$$\mathbb{E}(\omega) = \frac{b}{2c} \tanh(c/2) = \frac{b}{2c} \left(\frac{e^c - 1}{1 + e^c}\right).$$

Finally, the Pólya–Gamma class is closed under convolution for random variates with the same scale (tilting) parameter. If $\omega_1 \sim \text{PG}(b_1, z)$ and $\omega_2 \sim \text{PG}(b_2, z)$ are independent, then $\omega_1 + \omega_2 \sim \text{PG}(b_1 + b_2, z)$. This follows from the Laplace transform. We will employ this property later when constructing a Pólya–Gamma sampler.

3. THE DATA-AUGMENTATION STRATEGY

3.1 Main Result

The Pólya–Gamma family has been carefully constructed to yield a simple Gibbs sampler for the Bayesian logistic-regression model. The two differences from the Albert and Chib (1993) method for probit regression are that the posterior distribution is a scale mixture, rather than location mixture, of Gaussians; and that Albert and Chib's truncated normals are replaced by Pólya–Gamma latent variables.

To fix notation: let y_i be the number of successes, n_i the number of trials, and $x_i = (x_{i1}, \ldots, x_{ip})$ the vector of regressors for observation $i \in \{1, \ldots, N\}$. Let $y_i \sim \text{Binom}(n_i, 1/\{1 + e^{-\psi_i}\})$, where $\psi_i = x_i^T \boldsymbol{\beta}$ are the log odds of success. Finally, let $\boldsymbol{\beta}$ have a Gaussian prior, $\boldsymbol{\beta} \sim \text{N}(b, B)$. To sample from the posterior distribution using the Pólya–Gamma method, simply iterate two steps:

$$(\omega_i \mid \boldsymbol{\beta}) \sim \text{PG}\left(n_i, x_i^T \boldsymbol{\beta}\right)$$
$$(\boldsymbol{\beta} \mid y, \omega) \sim \text{N}\left(m_\omega, V_\omega\right),$$

where

$$V_\omega = (X^T \Omega X + B^{-1})^{-1}$$
$$m_\omega = V_\omega (X^T \kappa + B^{-1} b),$$

where $\kappa = (y_1 - n_1/2, \ldots, y_N - n_N/2)$, and Ω is the diagonal matrix of ω_i's.

We now derive this sampler, beginning with a careful statement and proof of the integral identity mentioned in Section 1.

Theorem 1. Let $p(\omega)$ denote the density of the random variable $\omega \sim \text{PG}(b, 0)$, $b > 0$. Then the following integral identity holds for all $a \in \mathbb{R}$:

$$\frac{(e^\psi)^a}{(1 + e^\psi)^b} = 2^{-b} e^{\kappa \psi} \int_0^\infty e^{-\omega \psi^2/2} \, p(\omega) \, d\omega, \tag{7}$$

where $\kappa = a - b/2$.

Moreover, the conditional distribution

$$p(\omega \mid \psi) = \frac{e^{-\omega\psi^2/2} \, p(\omega)}{\int_0^\infty e^{-\omega\psi^2/2} \, p(\omega) \, d\omega},$$

which arises in treating the integrand in Equation (7) as an unnormalized joint density in (ψ, ω), is also in the Pólya–Gamma class: $(\omega \mid \psi) \sim \text{PG}(b, \psi)$.

Proof. Appealing to Equation (3), we may write the left-hand side of Equation (7) as

$$\frac{(e^\psi)^a}{(1 + e^\psi)^b} = \frac{2^{-b} \exp\{\kappa\psi\}}{\cosh^b(\psi/2)}$$

$$= 2^{-b} e^{\kappa\psi} \, \mathbb{E}\{\exp(-\omega\psi^2/2)\},$$

where the expectation is taken with respect to $\omega \sim \text{PG}(b, 0)$ and where $\kappa = a - b/2$.

Turn now to the conditional distribution

$$p(\omega \mid \psi) = \frac{e^{-\omega\psi^2/2} \, p(\omega)}{\int_0^\infty e^{-\omega\psi^2/2} \, p(\omega) \, d\omega},$$

where $p(\omega)$ is the density of the prior, $\text{PG}(b, 0)$. This is of the same form as Equation (5), with $\psi = c$. Therefore $(\omega \mid \psi) \sim \text{PG}(b, \psi)$. $\square$

To derive our Gibbs sampler, we appeal to Theorem 1 and write the likelihood contribution of observation i as

$$L_i(\boldsymbol{\beta}) = \frac{\left\{\exp\left(x_i^T \boldsymbol{\beta}\right)\right\}^{y_i}}{1 + \exp\left(x_i^T \boldsymbol{\beta}\right)}$$

$$\propto \exp\left(\kappa_i x_i^T \boldsymbol{\beta}\right) \int_0^\infty \exp\left\{-\omega_i \left(x_i^T \boldsymbol{\beta}\right)^2 / 2\right\} p(\omega_i \mid n_i, 0),$$

where $\kappa_i = y_i - n_i/2$, and where $p(\omega_i \mid n_i, 0)$ is the density of a Pólya–Gamma random variable with parameters $(n_i, 0)$.

Combining the terms from all n data points gives the following expression for the conditional posterior of $\boldsymbol{\beta}$, given $\omega = (\omega_1, \ldots, \omega_N)$:

$$p(\boldsymbol{\beta} \mid \omega, y) \propto p(\boldsymbol{\beta}) \prod_{i=1}^N L_i(\boldsymbol{\beta} \mid \omega_i)$$

$$= p(\boldsymbol{\beta}) \prod_{i=1}^N \exp\left\{\kappa_i x_i^T \boldsymbol{\beta} - \omega_i \left(x_i^T \boldsymbol{\beta}\right)^2 / 2\right\}$$

$$\propto p(\boldsymbol{\beta}) \prod_{i=1}^N \exp\left\{\frac{\omega_i}{2} \left(x_i^T \boldsymbol{\beta} - \kappa_i/\omega_i\right)^2\right\}$$

$$\propto p(\boldsymbol{\beta}) \exp\left\{-\frac{1}{2}(z - X\boldsymbol{\beta})^T \Omega(z - X\boldsymbol{\beta})\right\},$$

where $z = (\kappa_1/\omega_1, \ldots, \kappa_n/\omega_N)$, and where $\Omega = \text{diag}(\omega_1, \ldots, \omega_N)$. This is a conditionally Gaussian likelihood in $\boldsymbol{\beta}$, with working responses z, design matrix X, and diagonal covariance matrix Ω^{-1}. Since the prior $p(\boldsymbol{\beta})$ is Gaussian, a simple linear-model calculation leads to the Gibbs sampler defined above.

3.2 Existing Data-Augmentation Schemes

A comparison with the methods of Holmes and Held (2006) and Frühwirth-Schnatter and Frühwirth (2010) clarifies how the Pólya–Gamma method differs from previous attempts at data augmentation. Both of these methods attempt to replicate

the missing-data mechanism of Albert and Chib (1993), where the outcomes y_i are assumed to be thresholded versions of an underlying continuous quantity z_i. For simplicity, we assume that $n_i = 1$ for all observations, and that y_i is either 0 or 1. Let

$$y_i = \begin{cases} 1, & z_i \geq 0 \\ 0, & z_i < 0 \end{cases}$$

$$z_i = x_i^T \boldsymbol{\beta} + \epsilon_i, \quad \epsilon_i \sim \text{Lo}(1), \tag{8}$$

where $\epsilon_i \sim \text{Lo}(1)$ has a standard logistic distribution. Upon marginalizing over the z_i, often called the latent utilities, the original binomial likelihood is recovered.

Although Equation (8) would initially seem to be a direct parallel with Albert and Chib (1993), it does not lead to an easy method for sampling from the posterior distribution of $\boldsymbol{\beta}$. This creates additional complications compared to the probit case. The standard approach has been to add another layer of auxiliary variables to handle the logistic error model on the latent-utility scale. One strategy is to represent the logistic distribution as a normal-scale mixture (Holmes and Held 2006):

$$(\epsilon_i \mid \phi_i) \sim \text{N}(0, \phi_i)$$

$$\phi_i = (2\lambda_i)^2, \quad \lambda_i \sim \text{KS}(1),$$

where λ_i has a Kolmogorov–Smirnov distribution (Andrews and Mallows 1974). Alternatively, one may approximate the logistic error term as a discrete mixture of normals (Frühwirth-Schnatter and Frühwirth 2010):

$$(\epsilon_i \mid \phi_i) \sim \text{N}(0, \phi_i)$$

$$\phi_i \sim \sum_{k=1}^K w_k \delta_{\phi^{(k)}},$$

where δ_ϕ indicates a Dirac measure at ϕ. The weights w_k and the points $\phi^{(k)}$ in the discrete mixture are fixed for a given choice of K so that the Kullback–Leibler divergence from the true distribution of the random utilities is minimized. Frühwirth-Schnatter and Frühwirth (2010) found that the choice of $K = 10$ leads to a good approximation, and list the optimal weights and variances for this choice.

In both cases, posterior sampling can be done in two blocks, sampling the complete conditional of $\boldsymbol{\beta}$ in one block and sampling the joint complete conditional of both layers of auxiliary variables in the second block. The discrete mixture of normals is an approximation, but it outperforms the scale mixture of normals in terms of effective sampling rate, as it is much faster.

One may also arrive at the hierarchy above by manipulating the random utility derivation of McFadden (1974); this involves the difference of random utilities, or "dRUM," using the term of Frühwirth-Schnatter and Frühwirth (2010). The dRUM representation is superior to the random utility approach explored in Frühwirth-Schnatter and Frühwirth (2007). Further work by Fussl, Frühwirth-Schnatter, and Frühwirth (2013) improves the approach for binomial logistic models. In this extension, one must use a table of different weights and variances representing different normal mixtures, to approximate a finite collection of type-III logistic distributions, and interpolate within this table to approximate the entire family.

Both Albert and Chib (1993) and O'Brien and Dunson (2004) suggested another approximation: namely, the use of a Student-t link function as a close substitute for the logistic link. But this
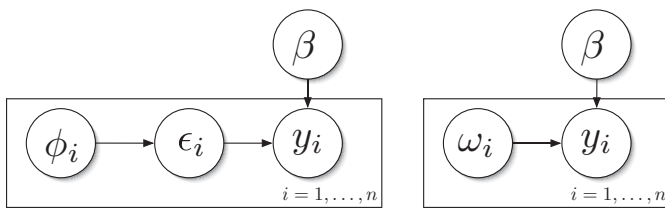
Figure 1. Directed acyclic graphs depicting two latent-variable constructions for the logistic-regression model: the difference of random-utility model of Holmes and Held (2006) and Frühwirth-Schnatter and Frühwirth (2010), on the left; versus our direct data-augmentation scheme, on the right.

also introduces a second layer of latent variables, in that the Student-t error model for z_i is represented as a scale mixture of normals.

Our data-augmentation scheme differs from each of these approaches in several ways. First, it does not appeal directly to the random-utility interpretation of the logit model. Instead, it represents the logistic CDF as a mixture with respect to an infinite convolution of gammas. Second, the method is exact, in the sense of making draws from the correct joint posterior distribution, rather than an approximation to the posterior that arises out of an approximation to the link function. Third, like the Albert and Chib (1993) method, it requires only a single layer of latent variables (see Figure 1).

A similar approach to ours is that of Gramacy and Polson (2012), who proposed a latent-variable representation of a powered-up version of the logit likelihood (see Polson and Scott 2013). This representation is useful for obtaining classical penalized-likelihood estimates via simulation, but for the ordinary logit model it leads to an improper mixing distribution for the latent variable. This requires modifications of the basic approach that make simulation difficult in the general logit case. As our experiments show, the method does not seem to be competitive on speed grounds with the Pólya–Gamma representation, which results in a proper mixing distribution for all common choices of a_i, b_i in Equation (2).

For negative-binomial regression, Frühwirth-Schnatter et al. (2009) employed the discrete-mixture/table-interpolation approach, like that used by Fussl, Frühwirth-Schnatter, and Frühwirth (2013), to produce a tractable data augmentation scheme. In some instances, the Pólya–Gamma approach outperforms this method; in others, it does not. The reasons for this

discrepancy can be explained by examining the inner workings of our Pólya–Gamma sampler, discussed in Section 4.

3.3 Mixed Model Example

We have introduced the Pólya–Gamma method in the context of a binary logit model. We do this with the understanding that, when data are abundant, the MH algorithm with independent proposals will be efficient, as asymptotic theory suggests that a normal approximation to the posterior distribution will become very accurate as data accumulate. This is well understood among Bayesian practitioners (e.g., Carlin 1992; Gelman et al. 2004).

But the real advantage of data augmentation, and the Pólya–Gamma technique in particular, is that it becomes easy to construct and fit more complicated models. For instance, the Pólya–Gamma method trivially accommodates mixed models, factor models, and models with a spatial or dynamic structure. For most problems in this class, good MH samplers are difficult to design, and at the very least will require ad hoc tuning to yield good performance.

Several relevant examples are considered in Section 5. But as an initial illustration of the point, we fit a binomial logistic mixed model using the data on contraceptive use among Bangladeshi women provided by the R package `mlmRev` (Bates, Maechler, and Bolker 2011). The data come from a Bangladeshi survey whose predictors include a woman's age, the number of children at the time of the survey, whether the woman lives in an urban or rural area, and a more specific geographic identifier based upon the district in which the woman resides. Some districts have few observations and district 54 has no observations; thus, a mixed model is necessary if one wants to include this effect. The response identifies contraceptive use. We fit the mixed model

$$y_{ij} \sim \text{Binom}(1, p_{ij}), \quad p_{ij} = \frac{e^{\psi_{ij}}}{1 + e^{\psi_{ij}}},$$
$$\psi_{ij} = m + \delta_j + x'_{ij}\beta,$$
$$\delta_j \sim N(0, 1/\phi),$$
$$m \sim N(0, \kappa^2/\phi),$$

where i and j correspond to the ith observation from the jth district. The fixed effect β is given an $N(0, 100I)$ prior, while the precision parameter ϕ is given $\text{Ga}(1, 1)$ prior. We take $\kappa \to \infty$ to recover an improper prior for the global intercept m. Figure 2 shows the box plots of the posterior draws of the
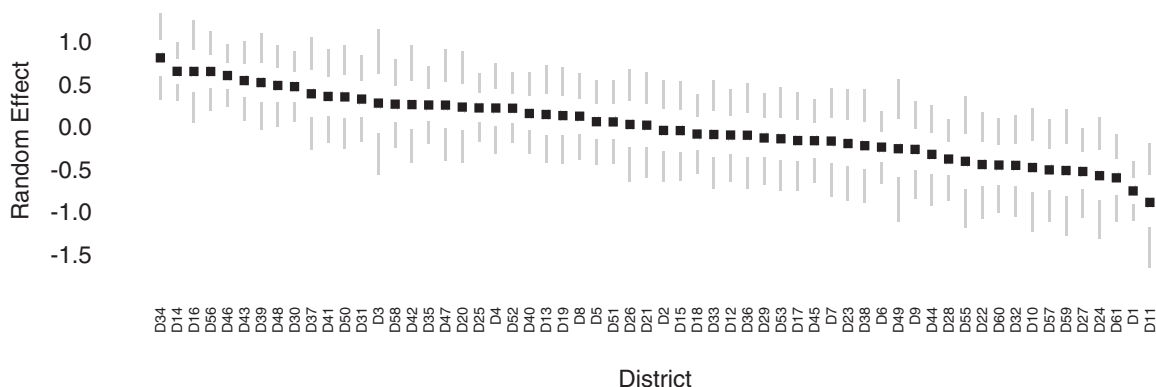

Figure 2. Marginal posterior distribution of random intercepts for each district found in a Bangladeshi contraception survey. For 10,000 samples after 2000 burn-in, median ESS = 8168 and median ESR = 59.88 for the PG method. Gray/white bars: 90%/50% posterior credible intervals. Black dots: posterior means.

random intercepts $m + \delta_j$. If one does not shrink these random intercepts to a global mean using a mixed model, then several take on unrealistic values due to the unbalanced design.

We emphasize that there are many ways to model these data, and that we do not intend our analysis to be taken as definitive. It is merely a proof of concept, showing how various aspects of Bayesian hierarchical modeling—in this case, models with both fixed and random effects—can be combined routinely with binomial likelihoods using the Pólya–Gamma scheme. Together these changes require just a few lines of code and a few extra seconds of runtime compared to the nonhierarchical logit model. A posterior draw of 2000 samples for this dataset takes 26.1 seconds for a binomial logistic regression, versus 27.3 seconds for a binomial logistic mixed model. As seen in the negative binomial examples below, one may also painlessly incorporate a more complex prior structure using the Pólya–Gamma technique. For instance, if given information about the geographic location of each district, one could place a spatial process prior upon the random offsets $\{\delta_j\}$.

4. SIMULATING PÓLYA–GAMMA RANDOM VARIABLES

4.1 The PG(1,z) Sampler

All our developments thus far require an efficient method for sampling Pólya–Gamma random variates. In this section, we derive such a method, which is implemented in the R package `BayesLogit`. We focus chiefly on simulating PG(1,z) efficiently, as this is most relevant to the binary logit model.

First, observe that one may sample Pólya–Gamma random variables naïvely (and approximately) using the sum-of-gammas representation in Equation (1). But this is slow and involves the potentially dangerous step of truncating an infinite sum.

We therefore construct an alternate, exact method by extending the approach of Devroye (2009) for simulating $J^*(1)$ from Equation (4). The distribution $J^*(1)$ is related to the Jacobi theta function, so we call $J^*(1)$ the Jacobi distribution. One may define an exponentially tilted Jacobi distribution $J^*(1, z)$ via the density

$$f(x \mid z) = \cosh(z) \, e^{-xz^2/2} \, f(x), \qquad (9)$$

where $f(x)$ is the density of $J^*(1)$. The PG(1, z) distribution is related to $J^*(1, z)$ through the rescaling

$$\mathrm{PG}(1, z) = \frac{1}{4} J^*(1, z/2). \qquad (10)$$

Devroye (2009) developed an efficient $J^*(1, 0)$ sampler. Following this work, we develop an efficient sampler for an exponentially tilted J^* random variate. In both cases, the density of interest can be written as an infinite, alternating sum that is amenable to the series method described in chap. IV.5 of Devroye (1986). Recall that a random variable with density f may be sampled using the accept/reject algorithm by (1) proposing X from a density g; (2) drawing $U \sim \mathcal{U}(0, cg(X))$ where $\|f/g\|_\infty \leq c$; and (3) accepting X if $U \leq f(X)$ and rejecting X otherwise. When $f(x) = \sum_{n=0}^{\infty}(-1)^n a_n(x)$ and the coefficients $a_n(x)$ are decreasing for all $n \in \mathbb{N}_0$, for fixed x in the support of f, then the partial sums, $S_n(x) = \sum_{i=0}^{n}(-1)^i a_i(x)$,

satisfy

$$S_0(x) > S_2(x) > \cdots > f(x) > \cdots > S_3(x) > S_1(x). \qquad (11)$$

In that case, step (3) above is equivalent to accepting X if $U \leq S_i(X)$ for some odd i, and rejecting X if $U > S_i(X)$ for some even i. Moreover, the partial sums $S_i(X)$ can be calculated iteratively. Below we show that for the $J^*(1, z)$ distribution the algorithm will accept with high probability upon checking $U \leq S_1(X)$.

The Jacobi density has two alternating-sum representations, $\sum_{n=0}^{\infty}(-1)^n a_n^L(x)$ and $\sum_{n=0}^{\infty}(-1)^n a_i^R(x)$, neither of which satisfy Equation (11) for all x in the support of f. However, each satisfies Equation (11) on an interval. These two intervals, respectively, denoted as I_L and I_R, satisfy $I_L \cup I_R = (0, \infty)$ and $I_L \cap I_R \neq \oslash$. Thus, one may pick $t \in I_L \cap I_R$ and define the piecewise coefficients

$$a_n(x) = \begin{cases} \pi(n + 1/2) \left(\dfrac{2}{\pi x}\right)^{3/2} \exp\left\{-\dfrac{2(n + 1/2)^2}{x}\right\}, \\ \qquad\qquad\qquad\qquad 0 < x \leq t, \quad (12) \\[2mm] \pi(n + 1/2) \exp\left\{-\dfrac{(n + 1/2)^2 \pi^2}{2} x\right\}, \\ \qquad\qquad\qquad\qquad x > t, \quad (13) \end{cases}$$

so that $f(x) = \sum_{n=0}^{\infty}(-1)^n a_n(x)$ satisfies the partial sum criterion (11) for $x > 0$. Devroye shows that the best choice of t is near 0.64.

Employing Equation (9), we now see that the $J^*(1, z)$ density can be written as an infinite, alternating sum $f(x|z) = \sum_{n=0}^{\infty}(-1)^n a_n(x|z)$, where

$$a_n(x|z) = \cosh(z) \exp\left\{-\frac{z^2 x}{2}\right\} a_n(x).$$

This satisfies Equation (11), as $a_{n+1}(x|z)/a_n(x|z) = a_{n+1}(x)/a_n(x)$. Since $a_0(x|z) \geq f(x|z)$, the first term of the series provides a natural proposal:

$$c(z) g(x|z)$$

$$= \frac{\pi}{2} \cosh(z) \begin{cases} \left(\dfrac{2}{\pi x}\right)^{3/2} \exp\left\{-\dfrac{z^2 x}{2} - \dfrac{1}{2x}\right\}, & 0 < x \leq t, \\[2mm] \exp\left\{-\left(\dfrac{z^2}{2} + \dfrac{\pi^2}{8}\right) x\right\}, & x > t. \end{cases}$$
$$\qquad (14)$$

Examining these two kernels, one finds that $X \sim g(x|z)$ may be sampled from a mixture of an inverse-Gaussian and an exponential:

$$X \sim \begin{cases} \mathrm{IG}(|z|^{-1}, 1)\mathbb{I}_{(0,t]} & \text{with prob. } p/(p + q) \\ \mathrm{Ex}(-z^2/2 + \pi^2/8)\mathbb{I}_{(t,\infty)} & \text{with prob. } q/(p + q), \end{cases}$$

where $p(z) = \int_0^t c(z) g(x|z)dx$ and $q(z) = \int_t^{\infty} c(z) g(x|z)dx$. Note that we are implicitly suppressing the dependence of p, q, c, and g upon t.

With this proposal in hand, sampling $J^*(1, z)$ proceeds as follows:

1. Generate a proposal $X \sim g(x|z)$.
2. Generate $U \sim \mathcal{U}(0, c(z)g(X|z))$.
3. Iteratively calculate $S_n(X|z)$, starting at $S_1(X|z)$, until $U \leq S_n(X|z)$ for an odd n or until $U > S_n(X|z)$ for an even n.
4. Accept X if n is odd; return to step 1 if n is even.

To sample $Y \sim \text{PG}(1, z)$, draw $X \sim J^*(1, z/2)$ and then let $Y = X/4$. The details of the implementation, along with pseudocode, can be found in the technical supplement.

4.2 Analysis of Acceptance Rate

This $J^*(1, z)$ sampler is very efficient. The parameter $c = c(z, t)$ found in Equation (13) characterizes the average number of proposals we expect to make before accepting. Devroye shows that in the case of $z = 0$, one can pick t so that $c(0, t)$ is near unity. We extend this result to nonzero tilting parameters and calculate that, on average, the $J^*(1, z)$ sampler rejects no more than 9 out of every 10,000 draws, regardless of z.

Proposition 1. Define

$$p(z, t) = \int_0^t \frac{\pi}{2} \cosh(z) \exp\left\{-\frac{z^2 x}{2}\right\} a_0^L(x) dx,$$

$$q(z, t) = \int_t^\infty \frac{\pi}{2} \cosh(z) \exp\left\{-\frac{z^2 x}{2}\right\} a_0^R(x) dx.$$

The following facts about the Pólya–Gamma rejection sampler hold.

1. The best truncation point t^* is independent of $z \geq 0$.
2. For a fixed truncation point t, $p(z, t)$ and $q(z, t)$ are continuous, $p(z, t)$ decreases to zero as z diverges, and $q(z, t)$ converges to 1 as z diverges. Thus, $c(z, t) = p(z, t) + q(z, t)$ is continuous and converges to 1 as z diverges.
3. For fixed t, the average probability of accepting a draw, $1/c(z, t)$, is bounded below for all z. For t^*, this bound to five digits is 0.99919, which is attained at $z \simeq 1.378$.

Proof. We consider each point in turn. Throughout, t is assumed to be in the interval of valid truncation points, $I_L \cap I_R$.

1. We need to show that for fixed z, $c(z, t) = p(z, t) + q(z, t)$ has a maximum in t that is independent of z. For fixed $z \geq 0$, $p(z, t)$ and $q(z, t)$ are both differentiable in t. Thus, any extrema of c will occur on the boundary of the interval $I_L \cap I_R$, or at the critical points for which $\frac{\partial c}{\partial t} = 0$; that is, $t \in I_L \cap I_R$, for which

$$\cosh(z) \exp\left\{-\frac{z^2}{2} t\right\} \left[a_0^L(t) - a_0^R(t)\right] = 0.$$

The exponential term is never zero, so an interior critical point must satisfy $a_0^L(t) - a_0^R(t) = 0$, which is independent of z. Devroye shows there is one such critical point, $t^* \simeq 0.64$, and that it corresponds to a maximum.
2. Both p and q are integrals of recognizable kernels. Rewriting the expressions in terms of the corresponding densities and integrating yields

$$p(z, t) = \cosh(z) \frac{\pi}{2} \frac{1}{y(z)} \exp\{-y(z)t\},$$

$$y(z) = \frac{z^2}{2} + \frac{\pi^2}{8},$$

and

$$q(z, t) = (1 + e^{-2z})\Phi_{\text{IG}}(t|1/z, 1),$$

where Φ_{IG} is the cumulative distribution function of an IG$(1/z, 1)$ distribution.

One can see that $p(z, t)$ is eventually decreasing in z for fixed t by noting that the sign of $\frac{\partial p}{\partial z}$ is determined by

$$\tanh(z) - \frac{z}{\frac{z^2}{2} + \frac{\pi^2}{8}} - zt,$$

which is eventually negative. (In fact, for the t^* calculated above it appears to be negative for all $z \geq 0$, which we do not prove here.) Further, $p(z, t)$ is continuous in z and converges to 0 as z diverges.

To see that $q(z, t)$ converges to 1, consider a Brownian motion (W_s) defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and the subsequent Brownian motion with drift $X_s^z = zs + W_s$. The stopping time $T^z = \inf\{s > 0 | X_s^z \geq 1\}$ is distributed as IG$(1/z, 1)$ and $\mathbb{P}(T^z < t) = \mathbb{P}(\max_{s \in [0, t]} X_s^z \geq 1)$.

Hence $\mathbb{P}(T^z < t)$ is increasing and $\lim_{z \to \infty} \mathbb{P}(T^z < t) = 1$, ensuring that $q(z, t) \propto (1 + e^{-2z})\mathbb{P}(T^z < t)$ converges to 1 as $z \to \infty$ as well. Continuity follows by considering the cumulative distribution $\mathbb{P}(T^z < t) = \Phi\{(zt - 1)/\sqrt{t}\} - \exp(2zt)\Phi\{(-1 - zt)/\sqrt{t}\}$, which is a composition of continuous functions in z.

By the continuity and tail behavior of p and q, it follows that $c(z, t) = p(z, t) + q(z, t)$, for fixed t, is continuous for all z and converges to 1 as z diverges. Further $c(z, t) \geq 1$ since the target density and proposal density satisfy $f(x|z) \leq c(z, t)g(x|z)$ for all $x \geq 0$. Thus, c takes on its maximum over z.
3. Since, for each t, $c(z, t)$ is bounded above in z, we know that $1/c(z, t)$ is bounded below above zero. For t^*, we numerically calculate that $1/c(z, t^*)$ attains its minimum 0.9991977 at $z \simeq 1.378$; thus, $1/c(z, t^*) > 0.99919$ suggesting that no more than 9 of every 10,000 draws are rejected on average.

$\square$

Since t^* is the best truncation point regardless of z, we will assume that the truncation point has been fixed at t^* and suppress it from the notation.

4.3 Analysis of Tail Probabilities

Proposition 1 tells us that the sampler rarely rejects a proposal. One possible worry, however, is that the algorithm might calculate many terms in the sum before deciding to accept or reject, and that the sampler would be slow despite rarely rejecting.

Happily, this is not the case, as we now prove. Suppose one samples $X \sim J^*(1, z)$. Let N denote the total number of proposals made before accepting, and let L_n be the number of partial sums S_i ($i = 1, \ldots, L_n$) that are calculated before deciding to accept or reject proposal $n \leq N$. A variant of Theorem 5.1 from Devroye (1986) employs Wald's equation to show that that $\mathbb{E}[\sum_{n=1}^N L_n] = \sum_{i=0}^\infty \int_0^\infty a_i(x|z) dx$. For the worst enclosing envelope, $z \simeq 1.378$, $\mathbb{E}[N] = 1.0016$; that is, on average, one rarely calculates anything beyond S_1 of the first proposal. A slight alteration of this theorem gives a more precise sense of how many terms in the partial sum must be calculated.

Proposition 2. When sampling $X \sim J^*(1, z)$, the probability of deciding to accept or reject upon checking the nth partial sum

$S_n, n \geq 1$, is

$$\frac{1}{c(z)} \int_0^\infty \{a_{n-1}(x|z) - a_n(x|z)\} \, dx.$$

Proof. Let L denote the number of partial sums that are calculated before accepting or rejecting the proposal. That is, a proposal X is generated; U is drawn from $\mathcal{U}(0, a_0(X|z))$; and L is the smallest natural number $n \in \mathbb{N}$ for which $U \leq S_n$ if n is odd or $U > S_n$ if n is even, where S_n denotes $S_n(X|z)$. But since L is the smallest n for which this holds, $S_{L-2} < U \leq S_L$ when L is odd and $S_L < U \leq S_{L-2}$ when L is even. Thus, the algorithm accepts or rejects if and only if $U \in K_L(X|z)$, where

$$K_n(x|z) = \begin{cases} (S_{n-2}(x|z), S_n(x|z)], & \text{odd } n \\ (S_n(x|z), S_{n-2}(x|z)], & \text{even } n. \end{cases}$$

In either case, $|K_n(x|z)| = a_{n-1}(x|z) - a_n(x|z)$. Thus,

$$\mathbb{P}(L = n | X = x) = \frac{a_{n-1}(x|z) - a_n(x|z)}{a_0(x|z)}.$$

Marginalizing over x yields

$$\mathbb{P}(L = n) = \frac{1}{c(z)} \int_0^\infty \{a_{n-1}(x|z) - a_n(x|z)\} dx. \qquad \square$$

Since each coefficient a_n is the piecewise composition of an inverse Gaussian kernel and an exponential kernel, these integrals may be evaluated. In particular,

$$a_n(x|z) = \cosh(z) \begin{cases} 2e^{-(2n+1)z} p_{\text{IG}}(x|\mu_n(z), \lambda_n), & x < t \\ \pi \left(n + \frac{1}{2}\right) \frac{1}{y_n(z)} p_{\mathcal{E}}(x|y_n(z)), & x \geq t, \end{cases}$$

where $\mu_n(z) = \frac{2n+1}{z}$, $\lambda_n = (2n+1)^2$, $y_n(z) = 0.5(z^2 + (n + 1/2)^2 \pi^2)$, and p_{IG} and $p_{\mathcal{E}}$ are the corresponding densities. The table below shows the first several probabilities for the worst-case envelope, $z \simeq 1.378$. Clearly $\mathbb{P}(L > n)$ decays rapidly with n.

n	1	2	3	4
$\mathbb{P}(L > n)$	8.023×10^{-4}	1.728×10^{-9}	8.213×10^{-18}	8.066×10^{-29}

Together with Proposition 2, this provides a strong guarantee of the efficiency of the PG(1,z) sampler.

4.4 The General PG(b, z) Case

To sample from the entire family of PG(b, z) distributions, we exploit the additivity of the Pólya–Gamma class. In particular, when $b \in \mathbb{N}$, one may sample PG(b, z) by taking b iid draws from PG($1, z$) and summing them. In binomial logistic regression, one will always sample PG(b, z) using integral b. This will also be the case in negative-binomial regression if one chooses an integer overdispersion parameter. In the technical supplement, we discuss the case of nonintegral b.

The runtime of the latent-variable sampling step is therefore roughly linear in the number of total counts in the dataset. For example, to sample 1 million Pólya–Gamma(1,1) random variables took 0.70 seconds on a dual-core Apple laptop, versus 0.17 seconds for the same number of Gamma random variables.

By contrast, to sample 1 million PG(10,1) random variables required 6.43 seconds, and to sample 1 million PG(100,1) random variables required 60.0 seconds.

We have had some initial success in developing a faster method to simulate from the PG(n, z) distribution that does not require summing together n PG(1,z) draws, and that works for noninteger values of n. This is an active subject of research, though somewhat beyond the scope of the present article, where we use the sum-of-PG(1,z)'s method on all our benchmark examples. A full report on the alternative simulation method for PG(n, z) may be found in Windle, Polson, and Scott (2013b).

5. EXPERIMENTS

We benchmarked the Pólya–Gamma method against several alternatives for logit and negative-binomial models. Our purpose is to summarize the results presented in detail in our online technical supplement, to which we refer to the interested reader.

Our primary metrics of comparison are the effective sample size and the effective sampling rate, defined as the effective sample size per second of runtime. The effective sampling rate quantifies how rapidly a Markov-chain sampler can produce independent draws from the posterior distribution. Following Holmes and Held (2006), the effective sample size (ESS) for the ith parameter in the model is

$$\text{ESS}_i = M \Bigg/ \left\{ 1 + 2 \sum_{j=1}^{k} \rho_i(j) \right\},$$

where M is the number of post-burn-in samples, and $\rho_i(j)$ is the jth autocorrelation of the chain corresponding to β_i. We use the `coda` package (Plummer et al. 2006), which fits an AR model to approximate the spectral density at zero, to estimate each ESS_i. All of the benchmarks are generated using R so that timings are comparable. Some R code makes external calls to C. In particular, the Pólya–Gamma method calls a C routine to sample the Pólya–Gamma random variates, just as R routines for sampling common distributions use externally compiled code. Here, we report the median effective sample size across all parameters in the model. Minimum and maximum effective sample sizes are reported in the technical supplement.

Our numerical experiments support several conclusions.

In binary logit models. First, the Pólya–Gamma is more efficient than all previously proposed data-augmentation schemes. This is true both in terms of effective sample size and effective sampling rate. Table 1 summarizes the evidence: across 6 real and 2 simulated datasets, the Pólya–Gamma method was always more efficient than the next-best data-augmentation scheme (typically by a factor of 200%–500%). This includes the approximate random-utility methods of O'Brien and Dunson (2004) and Frühwirth-Schnatter and Frühwirth (2010), and the exact method of Gramacy and Polson (2012). Frühwirth-Schnatter and Frühwirth (2010) find that their own method beats several other competitors, including the method of Holmes and Held (2006). We find this as well and omit these timings from our comparison. Further details can be found in Section 3 of the technical supplement.

Second, the Pólya–Gamma method always had a higher effective sample size than the two default Metropolis samplers

Table 1. Summary of experiments on real and simulated data for binary logistic regression

	Dataset							
	Nodal	Diab.	Heart	AC	GC1	GC2	Sim1	Sim2
ESS								
Pólya–Gamma	4860	5445	3527	3840	5893	5748	7692	2612
Best RU-DA	1645	2071	621	1044	2227	2153	3031	574
Best Metropolis	3609	5245	1076	415	3340	1050	4115	1388
ESR								
Pólya–Gamma	1632	964	634	300	383	258	2010	300
Best RU-DA	887	382	187	69	129	85	1042	59
Best Metropolis	2795	2524	544	122	933	223	2862	537

NOTE: ESS, the median effective sample size for an MCMC run of 10,000 samples; ESR, the median effective sample rate, or median ESS divided by the runtime of the sampler in seconds; AC, Australian credit dataset; GC1 and GC2, partial and full versions of the German credit dataset; Sim1 and Sim2, simulated data with orthogonal and correlated predictors, respectively; Best RU-DA, the result of the best random-utility data-augmentation algorithm for that dataset. Best Metropolis: the result of the Metropolis algorithm with the most efficient proposal distribution among those tested. See the technical supplement for full details.

we tried. The first was a Gaussian proposal using Laplace's approximation. The second was a multivariate t_6 proposal using Laplace's approximation to provide the centering and scale-matrix parameters, recommended by Rossi, Allenby, and McCulloch (2005) and implemented in the R package bayesm (Rossi 2012).

On five of the eight datasets, the best Metropolis algorithm did have a higher effective sampling rate than the Pólya–Gamma method, due to the difference in run times. But this advantage depends crucially on the proposal distribution, where even small perturbations can lead to surprisingly large declines in performance. For example, on the Australian credit dataset (labeled AC in the table), the Gaussian proposal led to a median effective sampling rate of 122 samples per second. The very similar multivariate t_6 proposal led to far more rejected proposals and gave an effective sampling rate of only 2.6 samples per second. Diagnosing such differences for a specific problem may cost the user more time than is saved by a slightly faster sampler.

Finally, the Pólya–Gamma method truly shines when the model has a complex prior structure. In general, it is difficult to design good Metropolis samplers for these problems. For example, consider a binary logit mixed model with grouped data and a random-effect structure, where the log-odds of success for observation j in group i are $\psi_{ij} = \alpha_i + x_{ij}\beta_i$, and where either the α_i, the β_i, or both receive further hyperpriors. It is not clear that a good default Metropolis sampler is easily constructed unless there are a large number of observations per group. Table 2 shows the results of naïvely using an independence Metropolis sampler based on the Laplace approximation to the full joint posterior. For a synthetic dataset with a balanced design of 100 observations per group, the Pólya–Gamma method is slightly better. For the two real datasets with highly unbalanced designs, it is much better.

Of course, it is certainly possible to design and tune better MH samplers for mixed models; see, for example, Gamerman (1997). We simply point out that what works well in the simplest case need not work well in a slightly more complicated case. The advantages of the Pólya–Gamma method are that it requires no tuning, is simple to implement, is uniformly ergodic (Choi and Hobert 2013), and gives optimal or near-optimal performance across a range of cases.

In negative-binomial models. The Pólya–Gamma method consistently yields the best effective sample sizes in negative-binomial regression. However, its effective sampling rate suffers when working with a large count or a nonintegral overdispersion parameter. Currently, our Pólya–Gamma sampler can draw from $PG(b, \psi)$ quickly when $b = 1$, but not for general, integral b: to sample from $PG(b, \psi)$ when $b \in \mathbb{N}$, we take b independent samples of $PG(1, \psi)$ and sum them. Thus in negative-binomial models, one must sample at least $\sum_{i=1}^{N} y_i$ Pólya–Gamma random variates, where y_i is the ith response at every MCMC iteration. When the number of counts is relatively high, this becomes a burden. (The sampling method described in Windle, Polson, and Scott (2013b) leads to better performance, but describing the alternative method is beyond the subject of this article.)

The columns labeled Sim1 and Sim2 of Table 3 show results for data simulated from a negative-binomial model with 400 observations and 3 regressors. (See the technical supplement for details.) In the first case (Sim1), the intercept is chosen so that the average outcome is a count of 8 (3244 total counts). Given the small average count size, the Pólya–Gamma method has a superior effective sampling rate compared to the approximate method of Frühwirth-Schnatter et al. (2009), the next-best choice. In the second case (Sim2), the average outcome is a count of 24 (9593 total counts). Here, the Frühwirth-Schnatter

Table 2. Summary of experiments on real and simulated data for binary logistic mixed models

	Dataset		
	Synthetic	Polls	Xerop
ESS			
Pólya–Gamma	6976	9194	3039
Metropolis	3675	53	3
ESR			
Pólya–Gamma	957	288	311
Metropolis	929	0.36	0.01

NOTE: Metropolis: the result of an independence Metropolis sampler based on the Laplace approximation. Using a t_6 proposal yielded equally poor results. See the technical supplement for full details.

Table 3. Summary of experiments on simulated data for
negative-binomial models

	Dataset			
	Sim1	Sim2	GP1	GP2
Total Counts	3244	9593	9137	22732
ESS				
Pólya–Gamma	7646	3590	6309	6386
FS09	719	915	1296	1157
Metropolis	749	764	—	—
ESR				
Pólya–Gamma	285	52	62	3.16
FS09	86	110	24	0.62
Metropolis	73	87	—	—

NOTE: Metropolis: the result of an independence Metropolis sampler based on a t_6 proposal. FS09: the algorithm of Frühwirth-Schnatter et al. (2009). Sim1 and Sim2: simulated negative-binomial regression problems. GP1 and GP2: simulated Gaussian-process spatial models. The independence Metropolis algorithm is not applicable in the spatial models, where there as many parameters as observations.

et al. algorithm finishes more quickly and therefore has a better effective sampling rate. In both cases, we restrict the sampler to integer overdispersion parameters.

As before, the Pólya–Gamma method starts to shine when working with more complicated hierarchical models that devote proportionally less time to sampling the auxiliary variables. For instance, consider a spatial model where we observe counts $y_1, \ldots, y_n$ at locations $x_1, \ldots, x_n$, respectively. It is natural to model the log rate parameter as a Gaussian process:

$$y_i \sim \text{NB}(n, 1/\{1 + e^{-\psi_i}\}), \quad \psi \sim \text{GP}(0, K),$$

where $\psi = (\psi_1, \ldots, \psi_n)^T$ and K is constructed by evaluating a covariance kernel at the locations x_i. For example, under the squared-exponential kernel, we have

$$K_{ij} = \kappa + \exp\left\{\frac{d(x_i, x_j)^2}{2\ell^2}\right\},$$

with characteristic length scale ℓ, nugget κ, and distance function d (in our examples, Euclidean distance).

Using either the Pólya–Gamma or the Frühwirth-Schnatter et al. (2009) techniques, one arrives at a multivariate Gaussian conditional for ψ whose covariance matrix involves latent variables. Producing a random variate from this distribution is expensive, as one must calculate the Cholesky decomposition of a relatively large matrix at each iteration. Therefore, the overall sampler spends relatively less time drawing auxiliary variables. Since the Pólya–Gamma method leads to a higher effective sample size, it wastes fewer of the expensive draws for the main parameter.

The columns labeled GP1 and GP2 of Table 3 show two such examples. In the first synthetic dataset, 256 equally spaced x points were used to generate a draw for ψ from a Gaussian process with length scale $\ell = 0.1$ and nugget $\kappa = 0.0$. The average count was $\bar{y} = 35.7$ or 9137 total counts (roughly the same as in the second regression example, Sim2). In the second synthetic dataset, we simulated ψ from a Gaussian process over 1000 x points, with length scale $\ell = 0.1$ and a nugget $= 0.0001$. This yielded 22,720 total counts. In both cases, the Pólya–Gamma

method led to a more efficient sampler—by a factor of three for the smaller problem, and five for the larger.

6. DISCUSSION

We have shown that Bayesian inference for logistic models can be implemented using a data augmentation scheme based on the novel class of Pólya–Gamma distributions. This leads to simple Gibbs-sampling algorithms for posterior computation that exploit standard normal linear-model theory and that are notably simpler than previous schemes. We have also constructed an accept/reject sampler for the new family, with strong guarantees of efficiency (Propositions 1 and 2).

The evidence suggests that our data-augmentation scheme is the best current method for fitting complex Bayesian hierarchical models with binomial likelihoods. It also opens the door for exact Bayesian treatments of many modern-day machine-learning classification methods based on mixtures of logits (e.g., Salakhutdinov, Mnih, and Hinton 2007; Blei and Lafferty 2007). Applying the Pólya–Gamma mixture framework to such problems is currently an active area of research.

Moreover, posterior updating via exponential tilting is a quite general situation that arises in Bayesian inference incorporating latent variables. In our case, the posterior distribution of ω that arises under normal pseudo-data with precision ω and a PG$(b, 0)$ prior is precisely an exponentially titled PG$(b, 0)$ random variable. This led to our characterization of the general PG(b, c) class. An interesting fact is that we were able to identify the conditional posterior for the latent variable strictly using its moment-generating function, without ever appealing to the Bayes' rule for density functions. This follows the Lévy-penalty framework of Polson and Scott (2012) and relates to work by Ciesielski and Taylor (1962) on the sojourn times of Brownian motion. There may be many other situations where the same idea is applicable.

Our benchmarks have relied upon serial computation. However, one may trivially parallelize a vectorized Pólya–Gamma draw on a multicore CPU. Devising such a sampler for a graphical-processing unit (GPU) is less straightforward, but potentially more fruitful. The massively parallel nature of GPUs offer a solution to the sluggishness found when sampling PG(n, z) variables for large, integral n, which was the largest source of inefficiency with the negative-binomial results presented earlier.

SUPPLEMENTARY MATERIALS

Technical Supplement: Additional details on the sampling algorithm and our empirical study of its efficiency.

REFERENCES

Albert, J. H., and Chib, S. (1993), "Bayesian Analysis of Binary and Polychotomous Response Data," *Journal of the American Statistical Association*, 88, 669–679. [1339,1341,1342,1343]

Andrews, D., and Mallows, C. (1974), "Scale Mixtures of Normal Distributions," *Journal of the Royal Statistical Society,* Series B, 36, 99–102. [1342]

Barndorff-Nielsen, O. E., Kent, J., and Sorensen, M. (1982), "Normal Variance-Mean Mixtures and z Distributions," *International Statistical Review*, 50, 145–159. [1340]

Bates, D., Maechler, M., and Bolker, B. (2011), *mlmRev: Examples from Multilevel Modelling Software Review*. Available at *http://CRAN.R-project.org/package=mlmRev*. R Package Version 1.0-1. [1343]

Biane, P., Pitman, J., and Yor, M. (2001), "Probability Laws Related to the Jacobi Theta and Riemann Zeta Functions, and Brownian Excursions," *Bulletin of the American Mathematical Society*, 38, 435–465. [1340,1341]

Blei, D. M., and Lafferty, J. (2007), "A Correlated Topic Model of *Science*," *The Annals of Applied Statistics*, 1, 17–35. [1348]

Carlin, J. (1992), "Meta-Analysis for 2×2 Tables: A Bayesian Approach," *Statistics in Medicine*, 11, 141–158. [1343]

Choi, H. M., and Hobert, J. P. (2013), The Polya-Gamma Gibbs Sampler for Bayesian Logistic Regression is Uniformly Ergodic," Technical Report, University of Florida. [1340,1347]

Ciesielski, Z., and Taylor, S. J. (1962), "First Passage Times and Sojourn Times for Brownian Motion in Space and the Exact Hausdorff Measure of the Sample Path," *Transactions of the American Mathematical Society*, 103, 434–450. [1348]

Devroye, L. (1986), *Non-Uniform Random Variate Generation*, Berlin: Springer. Available for download at *http://luc.devroye.org/rnbookindex.html*. [1340,1344,1345]

——— (2009), "On Exact Simulation Algorithms for Some Distributions Related to Jacobi Theta Functions," *Statistics & Probability Letters*, 79, 2251–2259. [1344]

Frühwirth-Schnatter, S., and Frühwirth, R. (2007), "Auxiliary Mixture Sampling With Applications to Logistic Models," *Computational Statistics and Data Analysis*, 51, 3509–3528. [1342]

——— (2010), "Data Augmentation and MCMC for Binary and Multinomial Logit Models," in *Statistical Modelling and Regression Structures*, Berlin: Springer-Verlag, pp. 111–132. Available from UT library online. [1339,1342,1346]

Frühwirth-Schnatter, S., Frühwirth, R., Held, L., and Rue, H. (2009), "Improved Auxiliary Mixture Sampling for Hierarchical Models of Non-Gaussian Data," *Statistics and Computing*, 19, 479–492. [1340,1343,1347,1348]

Fussl, A., Frühwirth-Schnatter, S., and Frühwirth, R. (2013), "Efficient MCMC for Binomial Logit Models," *ACM Transactions on Modeling and Computer Simulation*, 22, 1–21. [1342,1343]

Gamerman, D. (1997), "Sampling From the Posterior Distribution in Generalized Linear Mixed Models," *Statistics and Computing*, 7, 57–68. [1347]

Gelman, A., Carlin, J., Stern, H., and Rubin, D. (2004), *Bayesian Data Analysis* (2nd ed.), New York: Chapman and Hall/CRC. [1343]

Gramacy, R. B., and Polson, N. G. (2012), "Simulation-Based Regularized Logistic Regression," *Bayesian Analysis*, 7, 567–590. [1339,1343,1346]

Holmes, C., and Held, L. (2006), "Bayesian Auxiliary Variable Models for Binary and Multinomial Regression," *Bayesian Analysis*, 1, 145–168. [1339,1342,1346]

Jackman, S. (2009), *Bayesian Analysis for the Social Sciences*, New York: Wiley. [1339]

McFadden, P. (1974), "Conditional Logit Analysis of Qualitative Choice Behavior," in *Frontiers of Econometrics*, ed. P. Zarembka, New York: Academic, pp. 105–142. [1342]

O'Brien, S. M., and Dunson, D. B. (2004), "Bayesian Multivariate Logistic Regression," *Biometrics*, 60, 739–746. [1342,1346]

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006), "CODA: Convergence Diagnosis and Output Analysis for MCMC," *R News*, 6, 7–11. Available at *http://CRAN.R-project.org/doc/Rnews/*. [1346]

Polson, N. G., and Scott, J. G. (2012), "Local Shrinkage Rules, Lévy Processes, and Regularized Regression," *Journal of the Royal Statistical Society,* Series B, 74, 287–311. [1348]

——— (2013), "Data Augmentation for Non-Gaussian Regression Models Using Variance-Mean Mixtures," *Biometrika*, 100, 459–471. [1343]

Rossi, P. E. (2012), *Bayesm: Bayesian Inference for Marketing/Micro-Econometrics*, *http://CRAN.R-project.org/package=bayesm*. R Package Version 2.2-5. [1347]

Rossi, P. E., Allenby, G. M., and McCulloch, R. E. (2005), *Bayesian Statistics and Marketing*, New York: Wiley. [1339,1347]

Salakhutdinov, R., Mnih, A., and Hinton, G. (2007), "Restricted Boltzmann Machines for Collaborative Filtering," in *Proceedings of the 24th Annual International Conference on Machine Learning*, pp. 791–798. [1348]

Windle, J., Polson, N. G., and Scott, J. G. (2013a), *BayesLogit: Bayesian Logistic Regression*, *http://cran.r-project.org/web/packages/BayesLogit/index.html*. R Package Version 0.2-4. [1340]

——— (2013b), "Improved Pólya-Gamma Sampling," Technical Report, University of Texas at Austin. [1340,1346,1347]