The Dissertation Committee for Jesse Bennett Windle
certifies that this is the approved version of the following dissertation:

Forecasting High-Dimensional, Time-Varying Variance-Covariance Matrices with High-Frequency Data and Sampling Pólya-Gamma Random Variates for Posterior Distributions Derived from Logistic Likelihoods

Committee:

Carlos M. Carvalho, Supervisor

Peter Müller, Supervisor

James G. Scott

Jonathan Pillow

William Beckner

William H. Press

Forecasting High-Dimensional, Time-Varying Variance-Covariance Matrices with High-Frequency Data and Sampling Pólya-Gamma Random Variates for Posterior Distributions Derived from Logistic Likelihoods

by

Jesse Bennett Windle, B.S.; M.S.C.A.M.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2013

Preface

This thesis is the culmination of work with Profs. Carlos Carvalho, James Scott, and Nicholas Polson over the course of three years. Much of what is presented here has been reported in two papers: Windle and Carvalho [2012] and Polson et al. [2013b]. I refer to these works often, sometimes quoting passages, propositions, and so forth so long as I was the one that penned the words originally. The narrative of the thesis focuses on my contributions to these projects, though I am indebted to my collaborators for their many helpful insights and suggestions without which this dissertation would not have been possible. I must recognize that I had nothing to do with one particularly brilliant idea: the recognition that $1/\cosh\sqrt{t/2}$ is the Laplace transform of a tractable distribution and, further, that this integral identity can be used for data augmentation in binary logistic regression and related models is solely attributable to Profs. Scott and Polson.

Forecasting High-Dimensional, Time-Varying Variance-Covariance Matrices with High-Frequency Data and Sampling Pólya-Gamma Random Variates for Posterior Distributions Derived from Logistic Likelihoods

Publication No. _____

Jesse Bennett Windle, Ph.D.
The University of Texas at Austin, 2013

Supervisors: Carlos M. Carvalho
Peter Müller

The first portion of this thesis develops efficient samplers for the Pólya-Gamma distribution, an essential component of the eponymous data augmentation technique that can be used to simulate posterior distributions derived from logistic likelihoods. Building fast computational schemes for such models is important due to their broad use across a range of disciplines, including economics, political science, epidemiology, ecology, psychology, and neuroscience. The second portion of this thesis explores models of time-varying covariance matrices for financial time series. Covariance matrices describe the dynamics of risk and the ability to forecast future variance and covariance has a direct impact on the investment decisions made by individuals, banks, funds, and governments. Two options are pursued. The first incorporates information from high-frequency statistics into factor stochastic volatility models while the second models high-frequency statistics directly. The performance of each is assessed based upon its ability to hedge risk within a class of similarly risky assets.

Table of Contents

List of Tables

List of Figures

List of Symbols and Abbreviations

β_m	multivariate beta distribution
χ^2	χ^2 distribution
$\mathcal{E}$	exponential distribution
W_m	Wishart distribution
ESDMVP	empirical standard deviation of the minimum variance portfolios
Ga	gamma distribution
GP	Gaussian process distribution
IG	inverse Gaussian distribution
IGa	inverse gamma distribution
iid	independent and identically distributed
KS	Kolmogorov-Smirnov distribution
Lo	logistic distribution
MN	multinomial distribution
N	normal distribution
PG	Pólya-Gamma distribution

Chapter 1

Introduction

This thesis follows two paths of research. First, we discuss posterior simulation for logistic likelihoods. Such likelihoods are encountered in a number of statistical models, such as binary logistic regression, contingency tables, and negative binomial regression for count data. We focus on a data augmentation strategy called the Pólya-Gamma (PG) technique [Polson et al., 2013b], which, as its name suggests, employs Pólya-Gamma random variables to facilitate posterior simulation. A key factor in the efficiency of the PG technique is the rate at which one can generate such variates. Thus, developing good sampling schemes is important for the success of the PG method. Herein we develop three approaches, both exact and approximate, that taken together efficiently produce random variates from the Pólya-Gamma family across its entire parametric space. Second, we discuss forecasts of high-dimensional, time-varying covariance matrices that use high-frequency data. Over the last few decades, advances in storing and processing vast quantities of data have enabled statisticians and econometricians to measure variations and covariations in daily asset returns using data collected *within* a day, as opposed to data collected over several days. These statistics appear to offer better methods for estimating and predicting daily covariance matrices. We find that this is indeed the case compared to factor stochastic volatility, a well-worn Bayesian approach. To improve predictions, we

explore extensions to factor stochastic volatility that incorporate information from high-frequency statistics as well as matrix-variate methods that treat high-frequency statistics as data to be modeled.

Chapter 2

Sampling Pólya-Gamma Random Variates for Posterior Distributions Derived from Logistic Likelihoods

Efficient sampling from the Pólya-Gamma family of distributions is a key ingredient in the approach advocated by Polson et al. [2013b] for posterior inference when modeling count data, categorical data, or binary data. Their technique, called the Pólya-Gamma method, is useful when working with logistic likelihoods, which can be written as a product of terms of the form

$$\frac{(e^{\psi_i})^{a_i}}{(1+e^{\psi_i})^{b_i}}$$

where $\psi_i = x_i\beta$ is a linear combination of predictors and a_i and b_i are quantities determined by the number of trials, the response, the hyperparameters, or some combination thereof. Such models arise when one can describe the outcome variable in terms a proportion, such as in binary logistic regression, multinomial logistic regression, contingency tables, and negative binomial regression for count data, and this proportion is modeled on the log-odds scale. The aforementioned models are employed within a variety of quantitative research areas, such as economics, epidemiology, neuroscience, psychology, ecology, and political science. Thus it is of great interest to develop efficient, exact, and user friendly inference techniques for these models. The

PG method satisfies all of these criteria, though only when one may efficiently sample from the Pólya-Gamma family of distributions. This chapter develops samplers for that purpose.

Competing methods of posterior inference for these models include Metropolis-Hastings-based methods as well as Gibbs sampling based methods. Metropolis-Hastings methods require specifying a proposal distribution that is often accompanied by a tuning parameter, which is an impediment to out-of-the box usability as well as aesthetically unappealing. Gibbs sampling based methods, on the other hand, are fully automatic in the sense that they require no such tuning parameter. Instead, these methods introduce auxiliary variables that yield convenient complete conditional distributions. The Pólya-Gamma method follows this approach.

The structure of the chapter is as follows. First, we review binary logistic regression, which motivates the data augmentation trick and sampler to follow, and is one of the the two main benchmarks we will use to test the Pólya-Gamma method. The preliminary discussion (§2.1) summarizes alternate data augmentation schemes and shows how to employ the Pólya-Gamma data augmentation technique. Second, we define the Pólya-Gamma distribution, enumerate its properties, and present a general approach for Pólya-Gamma data augmentation (§2.2). Third, we devise a Pólya-Gamma sampler and analyze its efficiency both mathematically and empirically (§2.3, §2.4, and §2.5). The empirical portion includes an extensive suite of binary logistic regression and negative binomial regression benchmarks to highlight the performance of the Pólya-Gamma sampler and Pólya-Gamma technique against other methods. A discussion of dynamic binomial logistic regression (§2.6) follows

to exemplify the flexibility of the Pólya-Gamma approach. Lastly, we present alternate (§2.7) and approximate (§2.8) Pólya-Gamma samplers to address deficiencies of the original Pólya-Gamma generator and then compare all three samplers to identify where each sampler excels within the Pólya-Gamma parametric space (§2.9).

2.1 Background

2.1.1 Binary Logistic Regression

The canonical example that motivates the discussion to come is binary logistic regression, in which the response variable y_i may only take on two values, $\{0, 1\}$, with the probability that

$$P(y_i = 1|\beta) = p_i.$$

We write this conditionally since p_i depends upon β. In particular, logistic regression stipulates a linear relationship on the log-odds scale,

$$\psi_i = \log\Big(\frac{p_i}{1-p_i}\Big) = x_i\beta.$$

Here x_i is the d-dimensional row-vector of explanatory variables and may include a pseudo-variable to account for an intercept while β is a d-dimensional column vector. Given n observations and $\vec{p} = \{p_i\}$, the conditional density is

$$p(y|\vec{p}) \propto \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}.$$

Inverting the relationship between ψ_i and p_i yields

$$p_i = \frac{e^{\psi_i}}{1+e^{\psi_i}}$$

and hence the conditional density in $\psi = \{\psi_i\}$ is

$$p(y|\psi) \propto \prod_{i=1}^{n} \Big(\frac{e^{\psi_i}}{1+e^{\psi_i}}\Big)^{y_i} \Big(\frac{1}{1+e^{\psi_i}}\Big)^{1-y_i} \propto \prod_{i=1}^{n} \frac{e^{\psi_i y_i}}{1+e^{\psi_i}} \tag{2.1}$$

$$\propto \prod_{i=1}^{n} \delta_1(y_i)\Big(\frac{e^{\psi_i}}{1+e^{\psi_i}}\Big) + \delta_0(y_i)\Big(\frac{1}{1+e^{\psi_i}}\Big). \tag{2.2}$$

One may construct Gibbs samplers in at least two different ways. The best alternative Gibbs-sampling based methods introduce auxiliary random variables to augment the conditional density $p(y|\beta)$. In contrast, the PG method introduces auxiliary random variables to augment the posterior density $p(\beta|y)$.

2.1.2 Alternate Data Augmentation Approaches

Holmes and Held [2006], Frühwirth-Schnatter and Frühwirth [2007, 2010], and Fussl et al. [2013] introduce auxiliary random variables that yield convenient complete conditionals for binary or binomial logistic regression. A simple, approximate model is given in Albert and Chib [1993] using the t-link function. Holmes and Held [2006] and Frühwirth-Schnatter and Frühwirth [2010] take inspiration from the data augmentation approach of Albert and Chib [1993] for the probit model while Frühwirth-Schnatter and Frühwirth [2007] and Fussl et al. [2013] follow the random utility approach of McFadden [1974]. We summarize the various approaches below.

Proceeding as in Albert and Chib [1993], one may introduce a single auxiliary random variable z_i so that the data generating process for the conditional distribution $p(y|\psi)$ is expressed as

$$\begin{cases} y_i = \mathbf{1}\{z_i > 0\} \\ z_i = \psi_i + \nu_i, \qquad \nu_i \sim \text{Lo}(0,1) \end{cases} \tag{2.3}$$

where Lo is the logistic distribution. One may verify this by writing

$$p(y_i|\psi_i) = p(y_i|z_i)p(z_i|\psi_i)$$

where

$$p(y_i|z_i) = \delta_1(y_i)\mathbf{1}\{z_i > 0\} + \delta_0(y_i)\mathbf{1}\{z_i \leq 0\}$$

and use the cumulative distribution function of the logistic distribution to show that $P(z_i > 0) = e^{\psi_i}/(1 + e^{\psi_i})$ and hence

$$p(y_i|\psi_i) = \int_{-\infty}^{\infty} p(y_i|z_i)p(z_i|\psi_i)dz_i.$$

This representation does not produce a convenient form for Gibbs sampling and hence another layer of auxiliary random variables must be introduced.

Holmes and Held use the scale mixture of normals representation found in Andrews and Mallows [1974] to write the logistic error term as

$$\begin{cases} \text{Lo}(0,1) = N(0,\lambda) \\ \lambda \sim 4\text{KS}^2 \end{cases}$$

where KS is a Kolmogorov-Smirnov distribution [Devroye, 1986]. Thus, their augmented representation of y_i is

$$\begin{cases} y_i = \mathbf{1}\{z_i > 0\} \\ z_i = \psi_i + \varepsilon_i, \qquad \varepsilon_i \sim N(0,\lambda_i) \\ \lambda_i \sim 4\text{KS}^2. \end{cases}$$

This is precisely the form one seeks because $p(z|\beta)$ is now normal (recall that $\psi_i = x_i\beta$) and hence the conditional posterior, $p(\beta|z,\lambda)$, will also be normal given a normal prior for β. Further, this augmentation yields a manageable, though complex, posterior distribution $p(\lambda_i|\psi_i, y_i)$ that engenders a Gibbs-sampler.

Frühwirth-Schnatter and Frühwirth [2010] approximate $\mathrm{Lo}(0,1)$ with a discrete mixture of normals, that is

$$\begin{cases} \mathrm{Lo}(0,1) \approx N(0, v_r) \\ r \sim \mathrm{MN}(1, w). \end{cases}$$

where v is a known vector and $\mathrm{MN}(1,w)$ is a multinomial draw with a single trial and known weights w. Though this is not an exact representation of the logistic density, Frühwirth-Schnatter and Frühwirth show that the size of the discrete mixture need not be large to arrive at a very good approximation. The approximate, augmented representation is then

$$\begin{cases} y_i = \mathbf{1}\{z_i > 0\} \\ z_i = \psi_i + \varepsilon_i, \qquad \varepsilon_i \sim N(0, v_{r_i}) \\ r_i \sim \mathrm{MN}(1, w). \end{cases}$$

Again, this makes $p(z|\beta)$ normal and hence $p(\beta|z,r)$ is normal given a normal prior for β. The posterior distribution of r is multinomial.

In yet another data augmentation scheme, Frühwirth-Schnatter and Frühwirth [2007] make use the fact that the difference of two type-I extreme value distributions is logistically distributed. In particular, if ν_i^u and ν_i^0 are distributed as $-\log \mathcal{E}(1)$, then $\nu_i^u - \nu_i^0 \sim \mathrm{Lo}(0,1)$; thus, one may substitute

$$z_i = \psi_i + \nu_i^u - \nu_i^0$$

in (2.3). Letting

$$z_i^u = \psi_i + \nu_i^u$$

we see that $z_i > 0 \iff z_i^u > z_i^0$ where $z_i^0 = \nu_i^0$ so that (2.3) becomes

$$\begin{cases} y_i = \mathbf{1}\{z_i^u > z_i^0\} \\ z_i^u = \psi_i + \nu_i^u, \ \nu_i^u \sim -\log \mathcal{E}(1). \end{cases}$$

This is the random utility approach of McFadden [1974]. Approximating ν_i^u as a discrete normal of mixture $N(m_r, v_r)$, where m and v are vectors and $r \sim \text{MN}(1, w)$, though v and w are different than those from the proceeding paragraph, yields the augmented representation

$$\begin{cases} y_i = \mathbf{1}\{z_i^u > z_i^0\} \\ z_i^u = \psi_i + \varepsilon_i^u, \qquad \varepsilon_i^u \sim N(m_{r_i}, v_{r_i}) \\ r_i \sim \text{MN}(1, w). \end{cases}$$

This has tractable complete conditional or marginal posteriors.

Fussl et al. [2013] follow this approach for binomial models, though one must tabulate not a single discrete mixture, but an entire family. In particular, suppose that there there are k trials at each observation so that y_i is the number of successes accumulated from $y_{ij}, j = 1, \ldots, k$ binary responses. The McFadden [1974] model becomes

$$\begin{cases} y_i = \sum_{j=1}^n y_{ij} \\ y_{ij} = \mathbf{1}\{z_{ij}^u > z_{ij}^0\} \\ z_{ij}^u = \psi_i + \nu_{ij}^u \\ z_{ij}^0 = \nu_{ij}^0 \\ \nu_{ij}^u, \nu_{ij}^0 \sim -\log \mathcal{E}(1), \quad j = 1, \ldots, k. \end{cases}$$

Let $f : \mathbb{R}^n \to \mathbb{R}$ map $v \in \mathbb{R}^n$ to $-\log \sum_{j=1}^k \exp(-v_j)$. Then, letting

$$\begin{cases} z_i^* = f(\{y_{ij}^u\}_j) - f(\{y_{ij}^0\}_j) \\ \nu_i^* = f(\nu_{ij}^u) - f(\nu_{ij}^0), \end{cases}$$

we can collapse $z_{ij}^u = \psi_i + \nu_{ij}^u$ into

$$z_i^* = \psi_i + \nu_i^*, \;\; \nu_i^* \sim \text{Lo}_k,$$

where Lo_k is a type III logistic distribution, which in turn can be approximated by a discrete mixture of normals so that

$$\begin{cases} z_i^* = \psi_i + \varepsilon_i^*, \;\; \varepsilon_i^* \sim N(0, v_{k,r_i}) \\ r_i \sim \text{MN}(1, w_k). \end{cases}$$

Note the dependence of w and v on k: one must tabulate the weights and variances for the entire family Lo_k, $k \in \mathbb{N}$ for this to be useful. In practice, Fussl et al. [2013] tabulate weights and variances for some k and then interpolate within those weights and variances for arbitrary $k \in \mathbb{N}$.

Albert and Chib [1993] provide an approximate approach using a t(d)-link, which is a good choice since the logistic quantiles are almost a linear function of t(8) quantiles. The subsequent model is

$$\begin{cases} y_i = \mathbf{1}\{z_i > 0\} \\ z_i = \psi_i + \varepsilon_i, \qquad \varepsilon_i \sim N(0, 1) \end{cases}$$

to t(d)-link regression by introducing another random variable ϕ_i so that the latent structure is

$$\begin{cases} z_i = \psi_i + \varepsilon_i \sim N(0, 1/\phi_i) \\ \phi_i \sim \text{Ga}(d/2, d/2). \end{cases}$$

This requires only adding an additional step to sample $(\phi | z, \beta)$ to the Albert and Chib probit sampler.

All of the proceeding methods employ an augmented representation that requires at least *two* auxiliary variables. A heuristic rule is that the more variables one introduces, the greater the autocorrelation between samples. One prefers less autocorrelation because fewer samples are required to get a decent representation of the population. Thus, one would ideally employ as few auxiliary random variables as

possible when Gibbs sampling. The PG method uses only a single layer of auxiliary variables and hence has an intrinsic advantage compared to these other techniques.

2.1.3 The Pólya-Gamma Method

The alternate methods we discussed above can all be phrased in terms of data augmentation for the conditional distribution $p(y|\beta)$. In contrast, the PG method can be seen as data augmentation for the posterior distribution $p(\beta|y)$. Consider the posterior distribution generated from (2.1),

$$p(\beta|y) = c(y)\Big[\prod_{i=1}^{n} \frac{e^{y_i x_i \beta}}{1+e^{x_i\beta}}\Big]p(\beta).$$

If one can write a single term from the product as (recall that $\psi_i = x_i\beta$)

$$\frac{e^{\psi_i y_i}}{1+e^{\psi_i}} \propto e^{\psi_i \kappa_i} \int_0^\infty \exp\Big(-\frac{\psi_i^2}{2}\omega_i\Big)p(\omega_i|y)d\omega_i, \tag{2.4}$$

then one can produce a joint distribution

$$p(\beta,\omega|y) = C(y)\exp\Big(\kappa' X\beta - \frac{1}{2}\beta X'\Omega X\beta\Big)p(\beta)\prod_{i=1}^{n} p(\omega_i|y),$$

where Ω is a diagonal matrix with entries $\Omega_{ii} = \omega_i$, that marginalizes to the proper distribution for $(\beta|y)$. Note that the integral in (2.4) is like a Laplace transform. A consequence of this integral representation is that the conditional distribution $p(\beta|y,\omega)$ will be normal given a normal prior $p(\beta)$. If furthermore, one can sample from

$$p(\omega_i|y,\beta) \propto \exp\Big(-\frac{\psi_i^2}{2}\omega_i\Big)p(\omega_i|y), \tag{2.5}$$

then one can do Gibbs sampling as the conditional density for ω is

$$p(\omega|y,\beta) = \prod_{i=1}^{n} p(\omega_i|y,\beta).$$

The challenge is to find the distribution for ω_i for which (2.4) holds.

Of course, the Pólya-Gamma family of distributions is the class that satisfies (2.4) and (2.5). When $\omega_i \sim \text{PG}(1)$

$$1/\cosh(\psi_i/2) = \int_0^\infty \exp\Big(-\frac{\psi_i^2}{2}\omega_i\Big)p(\omega_i)d\omega_i \tag{2.6}$$

so that we have

$$\frac{e^{\psi_i y_i}}{1+e^{\psi_i}} = 2^{-1}e^{\psi_i(y_i-1/2)}\cosh^{-1}(\psi_i/2),$$

satisfying equation (2.4). Furthermore, we define the Pólya-Gamma family by exponentially tilting a PG(1) random variate so that $x \sim PG(1,z)$ has a density

$$p(x|z) \propto \exp(-\frac{z^2}{2}x)p(x),\ p(x) \sim PG(1). \tag{2.7}$$

Hence, by definition, the conditional distribution of ω_i given in (2.5) is $\text{PG}(1,\psi_i)$, which we will show later can be sampled efficiently.

Posterior Calculation 2.1 (Gibbs sampling for binary logistic regression)**.** *Given a normal prior $\beta \sim N(\beta; m_0, V_0)$, we may Gibbs sample $(\beta, \omega|y)$ by*

1. *$(\beta|y,\omega) \sim N(m_1, V_1)$ where*

$$\begin{cases} V_1^{-1} = V_0^{-1} + X'\Omega X \\ V_1^{-1}m_1 = X'\kappa + V_0^{-1}m_0 \end{cases}$$

   and $\kappa = y - 1/2$;

2. *$(\omega|y,\beta) \sim \prod_{i=1}^n p(\omega_i|y,\beta)$ where $\omega_i \sim PG(1,\psi_i)$ and $\psi = X\beta$.*

As we mentioned at the outset, the key to this being an efficient posterior simulation technique is the ability to efficiently sample from the $\text{PG}(b, \psi)$ distribution. Before tackling that problem, we define the Pólya-Gamma distribution more rigorously and present some of its properties.

2.2 The Pólya-Gamma Distribution

The Pólya-Gamma distribution is closely related to one of the probability laws described by Biane et al. [2001]. Thus this section is influenced by their work. We define the Pólya-Gamma distribution to satisfy equations similar to (2.6) and (2.7) and then show that such a distribution exists using Biane et al. [2001].

Definition 2.2 (The Pólya-Gamma Distribution)**.** *Suppose $b > 0$ and $z \geq 0$. A density $p(x|b)$ on $\mathbb{R}^+$ is $PG(b)$ if its Laplace transform is*

$$\cosh^{-b}(\sqrt{t/2}) = \int_0^\infty \exp(-tx)p(x|b)dx.$$

A random variable $X \sim PG(b, z)$ for $z > 0$ is defined by the density

$$p(x|b, z) = \cosh^b(z/2)\exp(-xz^2/2)p(x|b).$$

This characterizes the distribution since the Laplace transform uniquely determines a probability distribution [Billingsley, 1986]. We will see that such a Laplace transform exists below. Substituting $t = \psi^2/2$ into the Laplace transform shows that

$$\cosh^{-b}(z/2) = \int_0^\infty \exp(-xz^2/2)p(x|b)dx$$

and hence the expression for $p(x|b, z)$ is indeed a density.

Recall that we are interested in doing posterior inference for posteriors for the form

$$p(\beta|y) \propto p(\beta) \prod_{i=1}^{n} \frac{(e^{\psi_i})^{a_i}}{(1+e^{\psi_i})^{b_i}} \tag{2.8}$$

where a_i and b_i are some functions of the data y and other parameters and $\psi_i = x_i\beta$. The Pólya-Gamma distribution is the exact distribution needed to augment this posterior for simulation via Gibbs sampling. In fact, we may state a more general result by letting $\psi = \beta(\{x_i\}_i)$ where β is now a Gaussian process not a vector in $\mathbb{R}^p$. This framework subsumes many models, including regression and dynamic regression. See Rasmussen and Williams [2006] (an excellent book that is freely available online) for an introduction to Gaussian processes.

Posterior Calculation 2.3 (Pólya-Gamma Data Augmentation)**.** *Suppose the posterior is (2.8) where $\psi = \beta(X)$, X is the collection of covariates, and $p(\beta)$ is a Gaussian process prior for β. Then the density $p(\beta|y)$ may be augmented with random variables $(\omega_i|\psi_i, y) \sim PG(b_i, \psi_i), i = 1, \ldots, n$ so that the complete conditional $p(\beta|y, \omega)$ is equivalent to that derived from the non-parametric regression*

$$\begin{cases} z_i = \psi_i + \varepsilon_i, & \varepsilon_i \sim N(0, 1/\omega_i) \\ \psi = \beta(X) \\ \beta \sim GP(m, K). \end{cases}$$

where m and K are a mean and a covariance function,

$$z_i = \kappa_i/\omega_i,$$

is pseudo-data, and $\kappa_i = a_i - b_i/2$. Further, the complete conditional $p(\omega|y, \beta)$ is

$$\prod_{i=1}^{n} p(\omega_i|b_i, \psi_i).$$

We are implicitly conditioning on the covariates X throughout. Note that the Pólya-Gamma data augmentation turns simulation of the regression coefficient into weighted least squares.

Proof. Multiply (2.8) by $p(\omega|\beta, y) = \prod_{i=1}^n p(\omega_i|b_i, \psi_i)$ to get

$$\begin{aligned} p(\beta, \omega|y) &= p(\beta|y)p(\omega|\beta, y) \\ &\propto p(\beta) \prod_{i=1}^n \frac{(e^{\psi_i})^{a_i}}{(1+e^{\psi_i})^{b_i}} p(\omega_i|b_i, \psi_i) \\ &\propto p(\beta) \prod_{i=1}^n e^{\kappa_i \psi_i - \frac{1}{2}\psi_i^2 \omega_i} p(\omega_i|b_i). \end{aligned}$$

First, note that integrating in ω recovers the marginal posterior $p(\beta|y)$, so this is a valid data augmentation. Second, by construction, the complete conditional $p(\omega|\beta, y) = \prod_{i=1}^n p(\omega_i|b_i, \psi_i)$ as desired. Lastly, completing the square in the last proportional relationship shows that

$$p(\beta|\omega, y) \propto p(\beta) \prod_{i=1}^n \exp\Big(-\frac{1}{2}\omega_i (z_i - \psi_i)^2 \Big)$$

where $z_i = \kappa_i/\omega_i$. Since this product is identical to the likelihood in ψ of a collection of independent normal observations $z_i \sim N(\psi_i, 1/\omega_i)$, the posterior for $(\beta|\omega, y)$ is identical to the one generated by

$$\begin{cases} z_i = \psi_i + \varepsilon_i, & \varepsilon_i \sim N(0, 1/\omega_i), \\ \psi = \beta(X) \\ \beta \sim \mathrm{GP}(m, K). \end{cases}$$

□

Example 2.4 (Regression)**.** *Consider the case of regression. Suppose the posterior is (2.8) and that*

$$\begin{cases} \psi = X\beta, \\ \beta \sim N(m_0, V_0). \end{cases}$$

Then, following standard posterior calculations, $p(\beta|\omega, y) \propto N(\beta; m_1, V_1)$ *where*

$$\begin{cases} V_1^{-1} = V_0^{-1} + X'\Omega X, \;\; \Omega = diag\{\omega_i\}, \\ m_1 = V_1\Big[V_0^{-1}m_0 + X'\Omega z\Big] = V_1\Big[V_0^{-1}m_0 + X'\kappa\Big], \end{cases}$$

and $\kappa_i = a_i - b_i/2, \; i = 1, \ldots, n.$

We have yet to establish that the Pólya-Gamma distribution exists and that it has a density. However, Biane et al. [2001] essentially show this and many subsequently properties. They survey laws that connect analytic number theory and Brownian excursions. One such distribution, which we denote by $J^*(b)$, has Laplace transform given by

$$\mathbb{E}[e^{-tJ^*(b)}] = \cosh^{-b}(\sqrt{2t}).$$

Biane et al. [2001] show that this distribution has a density and derive *one* of its representations. Thus, the existence of $\text{PG}(b) = J^*(b)/4$ is verified and our definition of $\text{PG}(b, z)$ assured. When devising samplers, we find it convenient to work with the $J^*(b)$ distribution, since there is then a trove of prior work to reference directly, instead of obliquely by a re-scaling. Above, we extended the definition of the Pólya-Gamma family by exponential tilting so that

$$p_{PG}(x|z, b) = \cosh^b(z/2)e^{-xz^2/2}p_{PG}(x|b).$$

Similarly, we define $J^*(b, z)$ so that a $J^*(b, z)$ random variable has density

$$p_{J^*}(x|z, b) = \cosh^b(z)e^{-xz^2/2}p_{J^*}(x|b),$$

in which case we have the following definition.

Definition 2.5. *$J^*(b, z)$ is the distribution with Laplace transform*

$$\cosh^b(z)\cosh^{-b}(\sqrt{2t + z^2}).$$

As noted above, Biane et al. [2001] show that there is a distribution with Laplace transform $\cosh^{-b}(\sqrt{2t})$ and that it has a density. A single distribution may be turned into an entire family by exponential tilting (see p. 6 of Jensen [1995]). In particular, if a density $p(x)$ has Laplace transform $\varphi(t)$, then

$$e^{-\lambda x}p(x)/\varphi(\lambda)$$

engenders a family of densities, indexed by λ, that have Laplace transform

$$\varphi(t + \lambda)/\varphi(\lambda).$$

This is precisely the path we have taken to construct $J^*(b, z)$ and $\mathrm{PG}(b, z)$ using $\lambda = z^2/2$.

Fact 2.6. *The following aspects of the $J^*(b, z)$ distribution are useful.*

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

2. *$J^*(b)$ has a density and it may be written as*

$$p_{J^*}(x|b) = \frac{2^b}{\Gamma(b)}\sum_{n=0}^{\infty}(-1)^n\frac{\Gamma(n + b)}{\Gamma(n + 1)}\frac{(2n + b)}{\sqrt{2\pi x^3}}\exp\Big(-\frac{(2n + b)^2}{2x}\Big).$$

   Thus, the density of $J^*(b, z)$ is

$$p_{J^*}(x|b, z) = \cosh^b(z)e^{-xz^2/2}p_{J^*}(x|b).$$

3. *The $J^*(b,z)$ distribution is infinitely divisible. Thus, if $X \sim J^*(nb,z)$ where $b>0$ and $n \in \mathbb{N}$, and $X_i \overset{iid}{\sim} J^*(b,z)$ for $i=1,\ldots,n$, then*

$$X \overset{\mathcal{D}}{=} \sum_{i=1}^{n} X_i.$$

4. *The moment generating function of $J^*(b,z)$ is*

$$M(t;b,z) = \cosh^b(z)\cos^b(\sqrt{2t-z^2})$$

and may be written as an infinite product

$$\prod_{n=0}^{\infty}\Big(1-\frac{t}{d_n}\Big)^{-b},\ d_n = \frac{\pi^2}{2}\Big(n+\frac{1}{2}\Big)^2+\frac{z^2}{2}.$$

5. *Hence, $J^*(b,z)$ is an infinite convolution of gammas and can be represented as*

$$J^*(b,z) \sim \sum_{n=0}^{\infty}\frac{g_n}{d_n},\ g_n \overset{iid}{\sim} Ga(b,1).$$

Proof. Biane et al. [2001] provide justification for items (2), (3), and essentially (5). Justification for items (1) and (4) are in Polson et al. [2013b], though we present the arguments here. For item (1), let $X = J^*(b,z/2)$ and $Y = X/4$ transform

$$p_{J^*}(x|b,z/2)dx = \cosh^b(z/2)\exp\Big(-\frac{x}{4}\frac{z^2}{2}\Big)p_{J^*}(x|b)dx$$

to

$$\cosh^b(z/2)\exp\Big(-y\frac{z^2}{2}\Big)p_{J^*}(4y|b)d(4y) = \cosh^b(z/2)\exp(-yz^2/2)p_{PG}(y|b)dy.$$

The last expression is by definition $Y \sim \text{PG}(b,z)$. Regarding (4), recall the Laplace transform of $J^*(b,z)$ (Definition 2.5) is

$$\varphi(t|b,z) = \cosh^b(z)\cosh^{-b}(\sqrt{2t+z^2}).$$

By the Weierstrass factorization theorem [Pennisi, 1976], $\cosh(\sqrt{2t})$ can be written as

$$\cosh(\sqrt{2t}) = \prod_{n=0}^{\infty} \left(1 + \frac{t}{c_n}\right), \; c_n = \frac{\pi^2}{2}(n + 1/2)^2.$$

Taking the reciprocal of $\varphi(t|1, z)$ yields

$$\frac{\cosh(\sqrt{2t + z^2})}{\cosh(z)} = \frac{\prod_{n=0}^{\infty}\left(1 + \frac{t+z^2/2}{c_n}\right)}{\prod_{n=0}^{\infty}\left(1 + \frac{z^2/2}{c_n}\right)} = \prod_{i=0}^{\infty}\left(1 + \frac{t}{c_n + z^2/2}\right);$$

thus,

$$\varphi(t|b, z) = \prod_{n=0}^{\infty}\left(1 + \frac{t}{d_n}\right)^{-b}, \; d_n = \frac{\pi^2}{2}(n + 1/2)^2 + \frac{z^2}{2}.$$

Since $\varphi(-t; b, z) = M(t; b, z)$ we have

$$M(t; b, z) = \prod_{n=0}^{\infty}\left(1 - \frac{t}{d_n}\right)^{-b}$$

and

$$\frac{M(t; b, z)}{\cosh^b(z)} = \cosh^{-b}(\sqrt{-2t + z^2}) = \cos^{-b}(\sqrt{2t - z^2}).$$

Regarding item (5), one may invert the infinite product representation of Laplace transform to show that

$$J^*(b, z) \sim \sum_{n=0}^{\infty} \frac{g_n}{d_n}, g_n \overset{iid}{\sim} \text{Ga}(b, 1).$$

□

2.2.1 An Alternate Density

Below we devise a $J^*(1, z)$, which is motivated by Devroye [2009], and relies on a reciprocal relationship noticed by Ciesielski and Taylor [1962], who show that in

addition to Fact (2.6.2) one may represent the density of a $J^*(1)$ random variable as

$$\sum_{n=0}^{\infty}(-1)^n\pi\Big(n+\frac{1}{2}\Big)e^{(n+1/2)^2\pi^2x/2}. \tag{2.9}$$

By pasting these two densities together, one can construct an extremely efficient sampler. Unfortunately, there is no known general reciprocal relationship that would extend this approach to $J^*(n)$ for general n; however, Biane et al. [2001] provide an alternate density for the $J^*(2)$ distribution based upon a reciprocal relationship with another random variable.

While there may not be an obvious reciprocal relationship to use, one may find other alternate representations for the density of $J^*(h)$ random variables when h is a positive integer. Exploiting an idea from Kent [1980] for infinite convolutions of exponential random variables, one may invert the moment generating function using partial fractions. Consider the moment generating function of $J^*(h)$:

$$M(t)=\prod_{n=0}^{\infty}\Big(1-\frac{t}{c_n}\Big)^{-h},\ c_n=\frac{\pi^2}{2}(n+1/2)^2 \tag{2.10}$$

This can be expanded by partial fractions so that

$$M(t)=\sum_{n=0}^{\infty}\sum_{m=1}^{h}\frac{A_{nm}}{(t-c_n)^m}. \tag{2.11}$$

Inverting this sum term by term we find that one can represent the density as

$$f(x|h)=\sum_{n=0}^{\infty}\sum_{m=1}^{h}A_{nm}\frac{x^{m-1}e^{-c_ix}}{(m-1)!},$$

and infinite sum of gamma kernels.

To find formulas for the $\{A_{nm}\}_{nm}$ coefficients, consider the Laurent series expansion of $M(t)$ about c_i.

$$M(t) = \sum_{n=0}^{\infty} a_n^{(i)}(t - c_i)^n + \sum_{m=1}^{h} \frac{b_m^{(i)}}{(t - c_i)^m}. \tag{2.12}$$

Such an expansions is valid since c_n is an isolated singular point. Since the coefficients at the pole are unique, comparing coefficients in (2.11) and (2.12) shows that $A_{im} = b_m^{(i)}$. Further, one may calculate $b_m^{(i)}$ by considering the function

$$\nu_h(t) = (t - c_i)^h M(t)$$

and the computing

$$b_m^{(i)} = \frac{\nu_h^{(h-m)}(c_i)}{(h-m)!}.$$

(See Churchill and Brown [1984].) Writing the MGF in product form, as in (2.10), we see that

$$\nu_h(t) = (-c_i)^h \prod_{n \neq i} \Big(1 - \frac{t}{c_n}\Big)^h.$$

Define

$$\psi_h(t) = h \log(-c_i) - h \sum_{n \neq i} \Big(1 - \frac{t}{c_n}\Big).$$

Then $\log \nu_h(t) = \exp \psi_h(t)$ and the derivatives of ν can then be expressed as

$$\begin{aligned}
\nu_h' &= e^{\psi_h}\psi_h'; \\
\nu_h'' &= e^{\psi_h}(\psi_h')^2 + e^{\psi_h}\psi_h''; \\
\nu_h''' &= e^{\psi_h}(\psi_h')^3 + 3e^{\psi_h}\psi_h'\psi_h'' + e^{\psi_h}\psi_h''' \\
\ldots &= \ldots
\end{aligned}$$

where

$$\psi_1^{(k)}(t) = (k-1)! \sum_{n \neq i} (c_n - t)^{-k}.$$

Thus, one may calculate $b_m^{(i)}$ numerically using $\psi_h^{(k)}$, though the convergence may be slow.

However, the most important coefficient, $b_h^{(i)}$, is already known. Make the dependence of $b_m^{(i)}$ on h explicit by writing $b_m^{(i)}(h)$. From the formulas above we know that $b_h^{(i)}(h) = \nu_h(c_i)$ and that $\nu_h(c_i) = \exp(\psi_1(c_i))^h$. But $\exp(\psi_1(c_i)) = \nu_1(c_i) = b_1^{(i)}(1)$. From the reciprocal relationship provided at the start of the section, we know that $b_1^{(i)}(1) = (-1)^i \sqrt{2c_i}$. Thus,

$$A_{ih} = b_h^{(i)}(h) = (-1)^{ih} (2c_i)^{h/2}.$$

For $h \in \mathbb{N}$, the density for $J^*(h)$ takes the form

$$f(x|h) = \sum_{n=0}^{\infty} \Big[\sum_{m=1}^{h} \frac{A_{nm}(h-1)!}{A_{nh}(m-1)!} \frac{1}{x^{h-m}} \Big] \frac{A_{nh} x^{h-1} e^{-c_i x}}{(h-1)!}$$

so the A_{nh} terms dominate for large x. Further, among those terms, the first,

$$\frac{A_{0h} x^{h-1} e^{-c_0 x}}{(h-1)!} = \frac{(\pi/2)^h x^{h-1} e^{-c_0 x}}{(h-1)!},$$

should dominate as $x \to \infty$.

Remark 2.7. *This provides insight into the tail behavior of the $J^*(h)$ distribution. For the right tail, we expect the density to decay as a $Ga(h, c_0)$ distribution. Examining the representation (2.6.2), we expect the left tail to decay like $IGa(1/2, h^2/2)$. These two observations will prove useful when finding an approximation of the $J^*(h)$ density.*

We may multiply each of these densities by $e^{-xz^2/2}$ to determine the tail behavior of $J^*(h,z)$: the right tail should look like $Ga(h, c_0 + z^2/2)$ while the left tail should look like $IG(\mu = h/z, h^2)$.

2.3 A $J^*(n,z)$ sampler for $n \in \mathbb{N}$

Efficient Pólya-Gamma sampling, or equivalently $J^*(n,z)$ sampling, is the focus of this chapter and essential for the success of the PG method. One could truncate the sum-of-gammas representation (Fact 2.6.5) to generate an approximate random variate, but this is inexact, potentially leading to errors when simulating posterior distributions. It also requires generating many gamma random variates for each J^* random variate, which is a rather large computational cost to bear. However, in the case of $J^*(1,z)$ one may avoid this problem. Simulating from the $J^*(1,z)$ distribution is (1) pertinent for doing Pólya-Gamma data augmentation for binary logistic regression and (2) useful for generating $J^*(n,z)$ random variates, $n \in \mathbb{N}$, as $J^*(n,z)$ is equivalent in distribution to the sum of n independent $J^*(1,z)$ random variates. Devroye [2009] develops an efficient, exact sampler for $J^*(1)$ and thus his work is an important foundation upon which the $J^*(1,z)$ sampler is built.

The $J^*(1,z)$ sampler employs von Neumann's alternating sum method [Devroye, 1986], which is an accept/reject algorithm for densities that may be represented as infinite, alternating sums. To remind the reader about accept/reject samplers, one generates a random variable Y with density f by repeatedly generating a proposal X from density g and U from $\mathcal{U}(0, c\,g(X))$ where $c \geq \|f/g\|_\infty$ until

$$U \leq f(X); \text{ then set } Y \leftarrow X.$$

(See Robert and Casella [2005] for more details.) The von Neumann alternating sum method requires that the density be expressed as an infinite, alternating sum

$$f(x) = \lim_{n\to\infty} S_n(x),\ S_n(x) = \sum_{i=0}^{n} (-1)^i a_i(x)$$

for which the partial sums S_i satisfy the *partial sum criterion*

$$\forall x,\ S_0(x) > S_2(x) > \ldots > f(x) > \ldots > S_3(x) > S_1(x), \tag{2.13}$$

which is equivalent to the sequence $\{a_i(x)\}_{i=1}^{\infty}$ decreasing in i for all x. In that case, we have that $u < f(x)$ if and only if there is some odd i such that $u \leq S_i(x)$ and $u > f(x)$ if and only if there is some even i such that $u \geq S_i(x)$. Thus one need not calculate the infinite sum to see if $u < f(x)$, one only needs to calculate as many terms as necessary to find that $u \leq S_i(x)$ for odd i or $u \geq S_i(x)$ for even i. (We must be careful when $x = 0$.) Below we will see that for the $J^*(1, z)$ sampler, one rarely needs to calculate a partial sum past $S_1(x)$ before deciding to accept or reject.

2.3.1 Sampling from $J^*(1, z)$

The $J^*(1)$ density may be represented in two different ways

$$f(x) = \sum_{i=0}^{n} (-1)^n a_n^L(x) = \sum_{i=0}^{n} (-1)^n a_n^R(x),$$

corresponding to Fact (2.6.2) and (2.9), where

$$a_n^L(x) = \pi(n + \frac{1}{2})\Big(\frac{2}{\pi x}\Big)^{3/2} \exp\Big(-\frac{2(n + 1/2)^2}{x}\Big) \tag{2.14}$$

and

$$a_n^R(x) = \pi\big(n + \frac{1}{2}\big) \exp\Big(-\frac{(n + 1/2)^2\pi^2 x}{2}\Big). \tag{2.15}$$

Neither $\{a_n^L(x)\}_{n=0}^{\infty}$ or $\{a_n^R(x)\}_{n=0}^{\infty}$ are decreasing for all x, thus neither satisfy the partial sum criterion. However, Devroye shows that $a_n^R(x)$ is decreasing on $I_R = [(\log 3)/\pi^2, \infty)$ and that $a_n^L(x)$ is decreasing for $I_L = [0, 4/\log 3]$. These intervals overlap and hence one may pick t in the intersection of these two intervals to define the piecewise coefficient

$$a_n(x) = \begin{cases} a_n^L(x), & x \leq t \\ a_n^R(x), & x > t \end{cases}$$

so that $a_n(x) \geq a_{n+1}(x)$ for all n and all $x \geq 0$. Devroye finds that $t = 2/\pi$ is the best choice of t for his $J^*(1,0)$ sampler, which is where $a_0^L(x) = a_0^R(x)$ Below we show that this still holds for $J^*(1,z)$. Thus the density f may be written as

$$f(x) = \sum_{i=0}^{\infty} (-1)^n a_n(x)$$

and this representation does satisfy the partial sum criterion (2.13). The density of $J^*(1,z)$ is then

$$f(x|z) = \cosh(z) \exp(-xz^2/2) f(x)$$

according to our construction of $J^*(1,z)$, in which case it also has an infinite sum representation

$$f(x|z) = \sum_{i=0}^{\infty} (-1) a_n(x|z), \; a_n(x|z) = \cosh(z) \exp(-xz^2/2) a_n(x)$$

that satisfies (2.13) for the partial sums $S_n(x|z) = \sum_{i=0}^{n} (-1)^n a_i(x|z)$, as

$$a_n(x) \geq a_{n+1}(x) \implies a_n(x|z) \geq a_{n+1}(x|z).$$

Following our initial discussion of the von Neumann alternating sum method, all that remains is to find a suitable proposal distribution g. One would like to find

a distribution g for which $\|f/g\|_\infty$ is small, since this controls the rejection rate. A natural candidate for g is the density defined by the kernel $S_0(x|z) = a_0(x|z)$ as $S_0(x|z) \geq f(x|z)$ for all x. In that case, we sample $X \sim g$ until $U \sim \mathcal{U}(0, a_0(x|z))$ has $U \leq f(X)$.

The proposal g is thus defined from (2.14) and (2.15) by

$$g(x|z) \propto a_0(x|z) = \cosh(z) \begin{cases} \Big(\frac{2}{\pi x^3}\Big)^{1/2} \exp\Big(\frac{-1}{2x} - \frac{z^2}{2}x\Big) & x < t \\ \frac{\pi}{2} \exp\Big(-\Big[\frac{\pi^2}{8} + \frac{z^2}{2}\Big]x\Big) & x \geq t. \end{cases}$$

Let $a_0^L(x|z) = a_0(x|z)\mathbf{1}\{x < t\}$ be the left-hand kernel and define the right-hand kernel $a_0^R(x|z)$ similarly. Rewriting the exponent in the left-hand kernel yields

$$\begin{aligned} \frac{-1}{2x} - \frac{z^2}{2}x &= \frac{-z^2}{2x}(x^2 - 2x|z|^{-1} + 2x|z|^{-1} + z^{-2}) \\ &= \frac{-z^2}{2x}(x - |z|^{-1})^2 - |z|; \end{aligned}$$

hence

$$a_0^L(x|z) = (1 + e^{-2|z|})\ IG(x|\mu = |z|^{-1}, \lambda = 1)$$

where $IG(x|\mu, \lambda)$ is the density of the inverse Gaussian distribution,

$$IG(x|\mu, \lambda) = \Big(\frac{\lambda}{2\pi x^3}\Big)^{1/2} \exp\Big(\frac{-\lambda(x-\mu)^2}{2\mu^2 x}\Big).$$

The normalizing constants

$$p = \int_0^t a_0^L(x|z)dx \text{ and } q = \int_t^\infty a_0^R(x|z)dx \tag{2.16}$$

let us express g as the mixture

$$g(x|z) = \frac{q}{p+q}\frac{a_0^L(x|z)}{q} + \frac{p}{p+q}\frac{a_0^R(x|z)}{p}.$$

and shows that (suppressing the dependence on t)

$$c(z)g(x|z) = a_0(x|z) \text{ where } c(z) = p(z) + q(z).$$

Thus, one may draw $X \sim g(x|z)$ as

$$X \sim \begin{cases} IG(\mu = |z|^{-1}, \lambda = 1)\mathbf{1}\{x < t\}, & \text{with prob. } p/(p+q) \\ \mathcal{E}\Big(\text{rate} = \dfrac{\pi^2}{8} + \dfrac{z^2}{2}\Big)\mathbf{1}\{x \geq t\}, & \text{with prob. } q/(p+q). \end{cases}$$

One may sample from the truncated exponential by taking $X \sim Ex\Big(\text{rate} = \frac{\pi^2}{8} + \frac{z^2}{2}\Big)$ and returning $X + t$. Sampling from the truncated inverse Gaussian requires a bit more work, which we describe in Appendix 1. To recapitulate, to draw $J^*(1, z)$:

1. Sample $X \sim g(x|z)$.
2. Generate $U \sim \mathcal{U}(0, a_0(X|z))$.
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 $U > S_n(X|z)$ for an even n.
4. Accept if n is odd; return to step 1 if n is even.

2.3.2 Sampling $J^*(n, z)$

One can use the $J^*(1, z)$ sampler to generate draws from the $J^*(n, z)$ distribution when n is a positive integer. As shown by Fact (2.6.3), sample $X_i \sim J^*(1, z)$ for $i = 1, \ldots, n$ and then return $Y = \sum_{i=1}^{n} X_i$.

2.4 Analysis of the $J^*(1, z)$ Sampler

We are interested in quantifying the efficiency of the $J^*(1, z)$ sampler. There are several metrics of interest, including (1) the rejection rate of the sampler, which describes the number of proposals one must make before accepting; (2) the number of partial sums one must calculate before deciding to accept or reject, which controls the average time to sample a random variate; and (3) the total number of partial sums one must calculate over all of the proposals made.

2.4.1 Acceptance Rate

The probability of accepting a proposal $X \sim g(x|z)$, given the value of that proposal is

$$\mathbb{P}(U \leq f(X)|X = x) = \frac{f(x|z)}{c(z,t)g(x|z)}.$$

Integrating over the proposal density we find that the marginal probability of accepting a proposal is

$$\mathbb{P}(U \leq f(X)) = c(z,t)^{-1}.$$

Thus for each z we want to find $t = t(z)$ that minimizes $c(z,t)$; maximizing $c(z,t(z))$ over z then yields the worst possible acceptance rate. The following proposition and proof, which was taken from Polson et al. [2013b], shows the best value of t is independent of z and that even in the worst case scenario, the acceptance rate $c(z,t)^{-1}$ is close to unity.

Proposition 2.8. *Writing out the expressions for p and q from (2.16) yields*

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

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

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

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

   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, $t^* \simeq 2/\pi$, 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

$$p(z,t) = \cosh(z)\frac{\pi}{2}\frac{1}{y(z)}\exp\Big\{-y(z)t\Big\}, \quad y(z) = \frac{z^2}{2} + \frac{\pi^2}{8},$$

and

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

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 that 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 $q(z,t) = (1+e^{-2z})\mathbb{P}(T^z < t)$ 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.

□

Remark 2.9. *Having found that the best truncation point t^* is independent of z, we henceforth assume that value is fixed and drop it from the notation.*

2.4.2 The Distribution of Partial Sums Calculated

In the alternating sum algorithm, one proposes $X \sim g(x|z)$, generates $U \sim \mathcal{U}(0, S_0(X|z))$, and then checks $U \leq f(X|z)$ by iteratively computing partial sums. As seen in Figure 2.1, X will be accepted or rejected after calculating the nth partial sum if $U \in (S_{n-2}, S_n]$ for odd n (with the convention that $S_{-1} = 0$) or $U \in (S_n, S_{n-2}]$ for even n, for $n \in \mathbb{N}$. This observation motivates the following proposition.

Proposition 2.10. *When sampling $X \sim J^*(1,z)$, the probability of deciding to accept or reject a proposal after calculating the nth partial sum S_n, $n \in \mathbb{N}$, is*

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

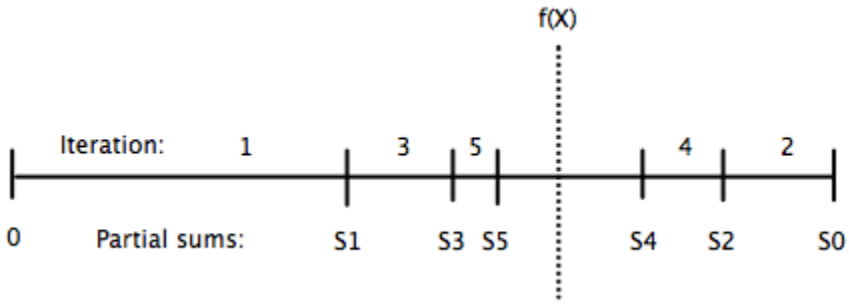


Figure 2.1: The probability of accepting or rejecting after calculating the jth partial sum. When deciding to accept or reject using von Neumann's alternating sum method one iteratively checks $U \leq S_j(X)$ if j is odd and $U > S_j(X)$ is j is even for $j = 1, 2, \ldots$. Thus the decision to stop is made on the jth iteration if U is in $(S_{j-2}, S_j]$ if j is odd and $(S_j, S_{j-2}]$ if j is even (with the convention that $S_{-1} = 0$). Since U is uniformly distributed given X the probability that U is in the jth interval is the ratio of the interval's length to S_0.

Proof. Let $S_n(x|z)$ denote the partial sums $\sum_{i=0}^{n}(-1)^i a_i(x|z)$ and let $S_{-1}(x|z) = 0$. Let L denote the number of partial sums calculated before deciding to accept or reject a proposal. That is X is drawn from $g(x|z)$, U is drawn from $\mathcal{U}(0, a_0(X|z))$, and L is the smallest natural number for which $U \in K_n(X|z)$ where

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

The probability

$$\mathbb{P}(L = n|X = x) = \mathbb{P}(U \in K_n|X = x) = \frac{|K(x|z)|}{a_0(x|z)}$$

since U is uniformly distributed. The length of the nth interval is

$$|K_n(x|z)| = a_{n-1}(x|z) - a_n(x|z);$$

so marginalizing over x we find that

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

□

We may calculate each integral analytically as the piecewise definition of a_i is composed of an exponential kernel and an inverse Gaussian kernel. For the exponential kernel:

$$a_n^R(x|z) = \cosh(z)\pi\Big(n+\frac{1}{2}\Big)\exp\Big\{-\frac{x}{2}\Big(z^2+(n+1/2)^2\pi^2\Big)\Big\}.$$

For the inverse Gaussian kernel,

$$2\cosh(z)\Big(2n+1\Big)\Big(\frac{1}{2\pi x^3}\Big)^{1/2}\exp\Big\{-\Big(\frac{z^2}{2}x+\frac{(2n+1)^2}{2x}\Big)\Big\},$$

we may rearrange the term in the exponent,

$$\begin{aligned}
-\frac{z^2}{2}x - \frac{(2n+1)^2}{2x} &= -\frac{z^2}{2x}\Big(x^2+\mu_n^2\Big),\ \mu_n^2 = (2n+1)^2/z^2 \\
&= \frac{-z^2}{2x}\Big(x^2 - 2\mu_n x + \mu_n^2 + 2\mu_n x\Big) \\
&= -z^2\mu_n - \frac{\lambda_n}{2x\mu_n^2}(x-\mu_n)^2,\ \lambda_n = (2n+1)^2,
\end{aligned}$$

to get

$$a_n^L(x|z) = 2\cosh(z)\exp(-|z|(2n+1))\Big(\frac{\lambda_n}{2\pi x^3}\Big)^{1/2}\exp\Big\{-\frac{\lambda_n}{2x\mu_n^2}(x-\mu_n)^2\Big\}.$$

The corresponding integrals of interest are

$$\int_t^\infty a_n^R(x|z) = \cosh(z)\frac{y_n}{\frac{1}{2}\Big(z^2+y_n^2\Big)}\exp\Big\{-\frac{t}{2}\Big(z^2+y_n^2\Big)\Big\}$$

where $y_n = (n + 1/2)\pi$; and

$$\int_0^t a_n^L(x|z) = (1 + e^{-2|z|})e^{-2n|z|}\Phi_{IG}(t|\mu = \mu_n, \lambda = \lambda_n).$$

The first four probabilities for the worst case z, $z = 1.378$, are

n	1	2	3	4
$\mathbb{P}(L = n)$	9.991977e-01	8.023005e-04	1.727943e-09	8.213354e-18

.

2.4.3 Average Number of Partial Sums Calculated

Given this distribution of inner loop iterations, one may calculate a variety of summary statistics, such as the average number of inner loop iterations. Namely,

$$\mathbb{E}[L] = \sum_{n=1}^{\infty} n\, \mathbb{P}(L = n).$$

Writing down this sum explicitly, one finds that

$$\sum_{n=1}^{\infty} n\, \mathbb{P}(L = n) = \frac{1}{c(z)} \sum_{n=0}^{\infty} \int_0^{\infty} a(x|z)dx,$$

which simplifies the calculation.

2.4.4 Wald's Theorem

The calculations above are done marginally, per proposal. For instance, we calculated the average number of partial sums calculated for each proposal, that is the average number of partial sums we expect to calculate before making a single decision to accept or reject. However, it is possible to go one step further and calculate these averages not just on a per proposal basis, but summed over all of the proposals made before accepting. This is Wald's Theorem [Devroye, 1986].

Theorem 2.11 (Wald's Theorem)**.** *Assume that $W_1, W_2, \ldots,$ are i.i.d. $\mathbb{R}^d$-valued random variables, and that Ψ is an arbitrary non-negative Borel measurable function of $\mathbb{R}^d$. Then, for all stopping rules N,*

$$\mathbb{E}\Big[\sum_{i=1}^{N} \Psi(W_i)\Big] = \mathbb{E}[N]\ \mathbb{E}[\Psi(W_1)].$$

To see how this theorem might be useful, imagine an accept/reject type algorithm. Each time one makes a proposal, one generates a vector of random variables $W_i = (X_i, U_i)$ that are i.i.d. It may be the case that W_i has more than two random variables. Further, there is some stopping rule N that records the total number of proposals and a function Ψ that maps W_i to some quantity of interest such as the number of floating point operations, the number of if statements evaluated, or the number of partial sums evaluated. We know that N is a stopping time because, as pointed out by Devroye, $\{N = n\}$ is determined by $W_1, \ldots, W_n$ and thus $N = n$ is $\mathcal{F}_n$-measurable. Wald's theorem tells us how to calculate the expected *total* number of whatever it is we are interested in, i.e. the expected value of

$$\sum_{i=1}^{N} \Psi(W_i).$$

Example 2.12 (Theorem 5.1, Devroye [1986])**.** *Let L_i denote the total number of partial sums calculated until deciding to accept or reject the ith proposal. Let N be the total number of proposals made before accepting. Then the average total number of partial sums calculated, over all proposals, is*

$$\sum_{n=0}^{\infty} \int_0^{\infty} a_n(x|z)dx.$$

Proof. By Wald's theorem

$$\mathbb{E}[\sum_{i=1}^{N} L_i] = \mathbb{E}[N]\mathbb{E}[L_i].$$

The probability of accepting a proposal is

$$\rho = c(z)^{-1}$$

and the number of proposals made before accepting, M, is $\text{NB}(1, 1-\rho)$. Thus, the expected value of N is $\mathbb{E}[N] = \mathbb{E}[M+1] = c(z)$. Hence,

$$\mathbb{E}[\sum_{i=1}^{N} L_i] = c(z)\frac{1}{c(z)}\sum_{n=0}^{\infty}\int_0^{\infty} a_n(x|z)dx.$$

□

Doing the calculation, $\mathbb{E}\Big[\sum_{i=1}^{N} L_i\Big]$ is 1.0016 when $z = 1.378$, the worst case scenario for z. In other words, on average, one only needs to execute 1.0016 inner-loop iterations in total when $z = 1.378$.

2.5 The $J^*(1, z)$ sampler in practice

We have shown that the $J^*(1, z)$ sampler has excellent theoretical performance, very often accepting the first proposal after calculating the first partial sum. However, the most important measure of the $J^*(1, z)$ sampler is not its theoretical properties, but rather the performance of the Pólya-Gamma data augmentation technique when using the $J^*(1, z)$ sampler. Thus, to gauge the strengths and weaknesses of the $J^*(1, z)$ sampler and the corresponding $J^*(n, z)$ sampler we will conduct an extensive suite of empirical tests comparing the Pólya-Gamma technique using this sampler to competing methods.

Two models are considered, binary logistic regression and negative binomial regression for count data. Binary logistic requires generating a single $J^*(1, z)$ random variate for each observation while negative binomial regression requires generating roughly y_i $J^*(1, z)$ random variates where y_i is the ith response. When the number of counts at each observation is somewhat large this may become a problem. Thus, the $J^*(1, z)$ sampler should perform well in the first case, but poorly in the second.

For binary logistic regression, the Pólya-Gamma technique is compared against 8 other techniques, 4 of which are data augmentation techniques and 4 of which are Metropolis-Hastings based techniques. (For more details on Metropolis-Hastings see Robert and Casella [2005].) For negative binomial regression, the Pólya-Gamma technique is compared against 2 other techniques, one of which is a data augmentation approach, the other of which is a random walk Metropolis sampler.

Following Frühwirth-Schnatter and Frühwirth [2010], the primary metric of comparison will be the effective sampling rate (ESR), which is the effective sample size per second (ESS). The ESR quantifies how quickly a sampler can produce independent draws from the posterior distribution. However, the ESR is sensitive to numerous idiosyncrasies relating to the implementation of the routines, the language in which they are written, and the hardware on which they are run. We generate these benchmarks using R, an interpretive language that is not nearly as fast as a compiled language, though some of the R routines make calls to external C code. Details on the implementations of the routines can be found in Appendix 3.

We draw several conclusions from these benchmarks. In terms of effective sample size, the Pólya-Gamma technique performs well. In terms of effective sam-

pling rate, the Pólya-Gamma method using the $J^*(1, z)$ sampler out-performs other data augmentation techniques for binary logistic regression. While the Pólya-Gamma method does not out-perform the independence Metropolis samplers for binary logistic regression, it is easier to code and more amenable to complex models. For instance, the independence Metropolis samplers that work well in the simplest case are not easily transferred to mixed models or factor models. For negative binomial regression the Pólya-Gamma method only performs well when the count sizes are relatively small. One can mitigate this deficiency by considering more complicated models in which a greater proportion of time will be spent simulating from other parameters or states present in the model. The poor performance in the negative binomial case is a consequence of the fact that the $J^*(1, z)$ sampler generates $J^*(n, z)$ by summing $n \in \mathbb{N}$ $J^*(1, z)$ random variates, an issue we will address later.

2.5.1 Benchmarking Procedure

For each data set, we run 10 MCMC batches, each batch generating 12,000 samples, the first 2,000 of which are discarded as burn in to leave a total of 10 batches of 10,000 samples. Each simulation produces a Markov chain whose invariant distribution is the posterior $p(\beta, \ldots | y)$ where $\ldots$ denote the auxiliary variables. For each component of the regression coefficient $\beta \in \mathbb{R}^p$ the effective sample size is calculated using the `coda` package [Plummer et al., 2006] and averaged across the 10 batches. The effective sample size estimates the number of (effectively) independent samples that have been produced by the Markov chain. Following Holmes and Held [2006],

the ESS of β_i from the kth batch is

$$\text{ESS}_{i,k} = M/\{1 + 2\sum_{j=1}^{\infty} \rho_{i,k}(j)\}$$

where M is the number of post-burn-in samples and $\rho_{i,k}(j)$ is the jth autocorrelation of the chain $C_{i,k} = \{\beta_i^{(t,k)}\}_{t=1}^{M}$. The `coda` package estimates the autocorrelation by fitting $C_{i,k}$ to an autoregressive process [Hamilton, 1994]. The component-wise effective sample sizes are then averaged across batches to produce

$$\text{ESS}_i = \frac{1}{10}\sum_{i=1}^{10} \text{ESS}_{i,k},$$

the point estimate of the effective sample size the the ith component of β.

The effective sample size is an important theoretical quantity, but from a practical perspective the rate at which one produces effectively independent samples is more meaningful. Hence, we normalize the effective sample sizes using the time it takes to generate the sample, yielding the component-wise effective sampling rates:

$$\text{ESR}_{i,k} = \frac{\text{ESS}_{i,k}}{\text{time to produce } M \text{ samples}}.$$

Note that the effective sampling rate does not include the time spent preprocessing or burning into the Markov chain. As with the ESS, we average the effective sampling rates over batches to produce component-wise ESR_i, $i = 1, \ldots, p$. We summarize the component-wise ESS and ESR via their minimum, median, and maximum. The numerical experiments are conducted using R 2.15.1 on an Ubuntu machine with an Intel Core i5 quad core processor and 8GB of RAM.

2.5.2 Binary Logistic Regression

Data Sets and Alternate Methods

The data sets used are featured in either Holmes and Held [2006] or Frühwirth-Schnatter and Frühwirth [2010]. We also construct two synthetic data sets to assess the effect of correlation in the predictors. The competing methods include data augmentation approaches, such as O'Brien and Dunson [2004], Frühwirth-Schnatter and Frühwirth [2010], Gramacy and Polson [2012], and Fussl et al. [2013], as well as Metropolis-based approaches such as Rossi et al. [2005]. We omit a comparison with Holmes and Held [2006] and Frühwirth-Schnatter and Frühwirth [2007] since a few initial tests showed these methods to be inferior Frühwirth-Schnatter and Frühwirth [2010], verifying the work therein. A complete description of the data sets and methods can be found in Appendix 3.

Results

As seen in Table 2.1, the Pólya-Gamma method beats all other data augmentation techniques for binary logistic regression in terms of both median effective sample size and median effective sampling rate. (Henceforth, we will always be referring to median values.) The Pólya-Gamma method always beats the independence Metropolis samplers in terms of effective sample size; however, it does not beat the independence Metropolis samplers in terms of effective sampling rate. Independence Metropolis samplers perform well since (1) proposals are cheap, a consequence of only performing large computations in the preprocessing stage, and (2) a Gaussian or t distribution that matches the mode and Hessian at the mode of the posterior will

		Nodal	Diab.	Heart	AC	GC1	GC2	Sim1	Sim2
ESS	Pólya-Gamma	4860	5445	3527	3840	5893	5748	7692	2612
	PG:Best Aug.	2.95	2.63	5.68	3.68	2.65	2.67	2.54	4.55
	PG:Best Met.	1.35	1.04	3.28	9.25	1.76	5.47	1.87	1.88
ESR	Pólya-Gamma	1632	964	634	300	383	258	2010	300
	PG:Best Aug.	1.84	2.52	3.39	4.35	2.97	3.04	1.93	5.08
	PG:Best Met.	0.58	0.38	1.17	2.46	0.41	1.16	0.70	0.56

Table 2.1: A summary of the binary logistic regression benchmarks. For each data set, an MCMC simulation generates 12,000 samples, the first 2,000 are discarded, leaving a total of 10,000 samples. The effective sample size (ESS) and effective sampling rates (ESR) are calculated for each component of the regression coefficient, β, individually. The rows labeled Pólya-Gamma report the median effective sample size of $\{\beta_i\}_{i=1}^{p}$. The rows labeled PG:Best Aug. and PG: Best Met. report the ratio of Pólya-Gamma median ESS or ESR to the ESS or ESR of the best alternative data augmentation scheme or Metropolis-Hastings scheme. A more detailed table may be found in Appendix 3.3.

make a good proposal given sufficient data. (Such an approach a proposal is called a Laplace approximation.)

However, a major advantage of data augmentation, and hence the Pólya-Gamma technique, is that it is easily adapted to more complicated models. Consider, for instance, a binary logistic mixed model whose intercepts are random effects, in which case the log odds for observation j from group i, ψ_{ij}, is modeled by:

$$\begin{cases} \psi_{ij} = \alpha_i + x_{ij}\beta \\ \alpha_i \sim N(m, 1/\phi) \\ m \sim N(0, \kappa^2/\phi) \\ \phi \sim Ga(1,1) \\ \beta \sim N(0, 100I). \end{cases} \tag{2.17}$$

An extra step is easily added to the Pólya-Gamma Gibbs sampler to estimate α, β, m and ϕ. However, if one follows the path taken for binary logistic regression, and

chooses a proposal based upon the Laplace approximation for all of the parameters together, then the independence Metropolis approach will perform poorly, as seen in Table 2.2. The results in Table 2.2 make use of a Gaussian proposal but a student-t proposal performs no better. Altering the independence Metropolis sampler to draw in blocks would require recalculating the posterior mode and variance for each block, a time consuming process that would negate the advantages of the independence Metropolis approach. It may be possible to find better Metropolis-based methods, such as Gamerman [1997]. We leave the task of trying all such possibilities to another researcher and simply observe that the Metropolis-based approaches that work well in the simple case do not transfer over to slightly more complicated cases; further that the very openness of the aforementioned proposition is a point in favor of the Pólya-Gamma technique. The Pólya-Gamma method combines speed, ease of use, and flexibility, requiring no input from the user in terms of selecting proposals or tuning parameters and accommodating more complicated models without hassle. Thus, when using the Pólya-Gamma approach, any sacrifice one might make in terms of effective sampling rate is made up for in simplicity.

2.5.3 Negative Binomial Regression

Data Sets and Alternate Methods

For negative binomial regression, we conduct benchmarks using two synthetic data sets consisting of $N = 400$ predictors and responses. The main difference between data sets is the average count size: one has fewer counts on average, while the

Synthetic: $N = 500$, $P_a = 5$, $P_b = 1$, samp=10,000, burn=2,000, thin=1								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	7.29	1.00	4289.29	6975.73	9651.69	588.55	957.18	1324.31
Ind-Met.	3.96	0.70	1904.71	3675.02	4043.42	482.54	928.65	1022.38
Polls: $N = 2015$, $P_a = 49$, $P_b = 1$, samp=100,000, burn=20,000, thin=10								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	31.94	1.00	5948.62	9194.42	9925.73	186.25	287.86	310.75
Ind-Met.	146.76	0.006	31.36	52.81	86.54	0.21	0.36	0.59
Xerop: $N = 1200$, $P_a = 275$, $P_b = 8$, samp=100,000, burn=20,000, thin=10								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	174.38	1.00	850.34	3038.76	4438.99	4.88	17.43	25.46
Ind-Met.	457.86	$\simeq 0$	1.85	3.21	12.32	0.00	0.01	0.03

Table 2.2: A set of three benchmarks for binary logistic mixed models. N denotes the number of samples, P_a denotes the number of groups, and P_b denotes the dimension of the fixed effects coefficient. The random effects are limited to group dependent intercepts. Notice that the second and third benchmarks are thinned every 10 samples to produce a total of 10,000 posterior draws. Even after thinning, the effective sample size for each is low compared to the PG method. The effective samples sizes are taken for the collection (α, β, m) and do not include ϕ. Taken from Polson et al. [2013b].

other has more counts on average. Synthetic responses are generated using the model

$$\begin{cases} y_i \sim NB(\text{mean} = \mu_i, d) \\ \log \mu_i = \iota + x_i \beta \end{cases}$$

where $\beta \in \mathbb{R}^3$. The model with fewer counts corresponds to $\iota = 2$ while the model with more counts corresponds to $\iota = 3$, producing a sample mean of roughly 8 in the former and 24 in the latter. In both cases, $d = 4$.

We compare the Pólya-Gamma method to the random walk Metropolis sampler of Rossi et al. [2005] and the discrete mixture of normals approach of Frühwirth-Schnatter et al. [2009] who exploit the Poisson-Gamma mixture representation of the negative binomial distribution. In particular, $y_i \sim \text{NB}(d, p_i)$ can be simulated via

$$\begin{cases} y_i \sim \text{Pois}(\lambda_i) \\ \lambda_i \sim \alpha_i \text{Ga}(d, 1) \end{cases}$$

where $\log(\alpha_i) = \psi_i = \log \frac{p_i}{1-p_i}$. Letting $z_i = \log \lambda_i$ we see that

$$z_i = \psi_i + \nu_i \,,\ \nu_i \sim \log \mathrm{Ga}(d, 1).$$

Approximating ν_i by a discrete normal of mixtures yields

$$\begin{cases} z_i = \psi_i + \varepsilon_i, & \varepsilon_i \sim N(m_{r,d}, v_{r,d}) \\ r_i \sim \mathrm{MN}(1, w_d), \end{cases}$$

where m_d, v_d, and w_d are the means, variances, and weights of the normal mixture that approximates a $\log \mathrm{Ga}(d, 1)$ distribution. As is the case for Fussl et al. [2013], this requires tabulating a large, finite number of discrete mixtures and then interpolating for those values of d not directly calculated. One can easily move between the log-odds and the log mean of the negative binomial distribution by $\log(\mu_i) = \psi_i + \log(d)$.

Results

The Pólya-Gamma approach out-performs the other methods in terms of effective sample size; however, its effective sampling rates fare less well when working with anything but small counts. Recall that, currently, a $\mathrm{PG}(n, z)$ random variate is generated by summing n $J^*(1, z)$ random variates and that the likelihood for negative binomial regression is

$$\prod_{i=1}^{n} \frac{(e^{\psi_i})^{y_i}}{(1 + e^{\psi_i})^{d+y_i}}.$$

Following Posterior Calculation 2.3, the auxiliary variables will be sampled as $\mathrm{PG}(b_i, \psi_i)$ where $b_i = d + y_i$, $i = 1, \ldots, N$. Thus, one must sample $Nd + \sum_{i=1}^{N} y_i$ $J^*(1, z)$ random variates, where y_i is the response, at every MCMC iteration. When the number of counts is relatively high this becomes a burden and the $J^*(n, z)$ sampler performs

poorly. We see this in Table (2.3), where the Pólya-Gamma sampler does well when working with relatively small count sizes, but poorly when that is not the case. (For all models we consider, the parameter d is estimated using a random-walk Metropolis-Hastings step over the integers.)

The Pólya-Gamma method does better when working with models that devote proportionally less time to sampling the auxiliary variables. For instance, consider the model

$$\begin{cases} y_i \sim NB(\text{mean} = \mu_i, d), \ i = 1, \ldots, N \\ \log \mu_i = \upsilon(x_i), \\ \upsilon \sim GP(0, K) \end{cases}$$

where K is the square exponential covariance kernel,

$$K(x_1, x_2) = \kappa + \exp\Big(\frac{\|x_1 - x_2\|^2}{2\ell^2}\Big),$$

with characteristic length scale ℓ and nugget κ. Again, one must sample $Nd + \sum_{i=1}^{N} y_i$ $J^*(1, z)$ random variates to generate the PG auxiliary variables. However, to sample $\log \mu$, one must calculate the Cholesky decomposition of a precision matrix of order N, the number of responses. One can preprocess this decomposition if the other parameters are known; however, if one wants to estimate d or one of the hyperparameters in K, then this decomposition must be repeated at every step in the Gibbs sampler. In that case, the time taken to sample the PG random variates is the same as in the parametric regression setting above, but the proportion of time spent sampling them is reduced. The same holds for the auxiliary variables in Frühwirth-Schnatter et al. [2009].

Table (2.4) shows that the time spent calculating a large Cholesky decomposition at each step alters the ESR calculation to make the Pólya-Gamma approach

Less Counts: $\alpha = 2$, $\bar{y} = 8.11$, $\sum y_i = 3244$, $N = 400$								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	26.84	1.00	7269.13	7646.16	8533.51	270.81	284.85	317.91
FS	8.10	1.00	697.38	719.36	759.13	86.10	88.80	93.70
RAM	10.17	30.08	737.95	748.51	758.57	72.59	73.62	74.61
More Counts: $\alpha = 3$, $\bar{y} = 23.98$, $\sum y_i = 9593$, $N = 400$								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	58.99	1.00	3088.04	3589.67	4377.21	52.35	60.85	74.20
FS	8.21	1.00	901.50	915.39	935.06	109.73	111.45	113.84
RAM	8.69	30.33	757.91	763.81	771.73	87.25	87.93	88.84

Table 2.3: Negative binomial regression benchmarks. PG is the Pólya-Gamma Gibbs sampler. FS follows Frühwirth-Schnatter et al. [2009]. RAM is the random walk Metropolis-Hastings sampler from the `bayesm` package. α is the true intercept and y_i is the ith response. Each model has three continuous predictors. Taken from Polson et al. [2013b].

competitive. In the first synthetic data set in Table (2.4), 256 equally spaced points in $\mathbb{R}^2$ are used to generate a draw $\upsilon(x_i)$ and y_i for $i = 1, \ldots, 256$ where $\upsilon \sim GP(0, K)$ and K has length scale $\ell = 0.1$ and a nugget $= 0.0$. The average count value of the synthetic data set is $\bar{y} = 35.7$, yielding 9137 total counts, which is roughly the same amount as in the larger negative binomial example discussed earlier. Whereas before the Pólya-Gamma method lost when working with this number of total counts, it now wins. In the second synthetic data set, 1000 randomly selected points were chosen to generate a draw from $\upsilon(x_i)$ where $\upsilon \sim GP(0, K)$ and K has length scale $\ell = 0.1$ and a nugget $= 0.0001$. The average count value is $\bar{y} = 22.72$, yielding $22,720$ total counts. The larger problem shows an even greater improvement in performance over the method of Frühwirth-Schnatter et al. Hence, despite the $J^*(1, z)$ sampler's poor performance for negative binomial regression, there are situations where it is still useful.

Gaussian Process: $\bar{y} = 35.7$, $\sum y_i = 9137$, $N = 256$, $\ell = 0.1$, nugget=0.0								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	101.89	1.00	790.55	6308.65	9798.04	7.76	61.92	96.19
FS	53.17	1.00	481.36	1296.27	2257.27	9.05	24.38	42.45
Gaussian Process: $\bar{y} = 22.7$, $\sum y_i = 22732$, $N = 1000$, $\ell = 0.1$, nugget=0.0001								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	2021.78	1.00	1966.77	6386.43	9862.54	0.97	3.16	4.88
FS	1867.05	1.00	270.13	1156.52	1761.70	0.14	0.62	0.94

Table 2.4: Non-parametric negative binomial regression benchmarks. PG is the Pólya-Gamma method. FS follows Frühwirth-Schnatter et al. [2009]. There are roughly as many total counts as in the first table as their are in the larger example in Table 2.3; however, the cost of drawing the posterior mean at the observed data points is much greater in this case, which reduces the penalty associated with sampling many Pólya-Gamma random variables. The second table shows that the cost drawing the posterior mean is even more pronounced for larger problems. N is the total number of observations and y_i denotes the ith observation. Taken from Polson et al. [2013b].

2.5.4 Recapitulation

The $J^*(1, z)$ sampler works well for binary logistic regression and binary logistic mixed models, but poorly for negative binomial regression. The poor performance is due to the way in which one samples $J^*(n, z)$ for $n \in \mathbb{N}$. In particular, such random variates heretofore have been generated by summing n independent draws from $J^*(1, z)$, a time consuming process when n is large. Thus, a major goal in the sequel is develop new $J^*(n, z)$ samplers that do better when drawing $J^*(n, z)$ for large n and, consequently, that improve the performance of the Pólya-Gamma technique for negative binomial regression. First, though, we will take a closer look the the inherent advantages of the Pólya-Gamma technique over Metropolis-based approaches when working with slightly more complicated models.

The Pólya-Gamma approach is superior in general because it is fast, easy to

use, and flexible. Above, we justify this claim by running benchmarks for a set of binary logistic mixed models. Those benchmarks did not look closely at the collection of Metropolis-Hastings methods available and hence our conclusion is not definitive. Mixed models are but one example we could have considered: factor models and dynamic models are two more cases where the independence Metropolis techniques used in regression do not easily transfer. Thus, to bolster our claim further, we take a more detailed look at the dynamic binary logistic models.

2.6 Dynamic Models : A Case Study

In §2.5.2, we found that independence Metropolis often works well for binary logistic regression. Simply extending what works well in that simple case does not apply to more complicated models, such as mixed models, though we did not examine all of the possible variations in Metropolis-Hastings like algorithms one might pursue. Here, we examine another class of more complicated models, dynamic generalized linear models with logistic likelihoods, and argue that there is no Metropolis-based analog that performs better in this setting. Thus, there are settings in which the Pólya-Gamma approach decisively out-performs all other options.

Recall that, in the most general setting we have considered, posterior distributions amenable to the Pólya-Gamma technique take the form

$$p(\beta|y) = p(\beta) \prod_{i=1}^{n} \frac{(e^{\psi_i})^{a_i}}{(1+e^{\psi_i})^{b_i}}$$

where $\psi_i = \beta(x_i)$ and $\beta \sim \mathrm{GP}(m, K)$. In the dynamic case, the prior for β becomes a Gaussian time series, for instance, insisting that β is an autoregressive process of

order 1 [Hamilton, 1994], that is $\beta \sim \text{AR}(1)$, imposes the prior dynamics

$$\beta_i = \mu + \phi(\beta_{i-1} - \mu) + \eta_i, \eta_i \sim N(0, W).$$

Applying Posterior Calculation 2.3, the complete conditional for β under the augmented model is identical to the posterior for β under

$$\begin{cases} z_i = \psi_i + \varepsilon_i, & \varepsilon_i \sim N(0, 1/\omega_i), \\ \psi_i = x_i \beta_i, & \\ \beta_i = \mu + \phi(\beta_{i-1} - \mu) + \eta_i, & \eta_i \sim N(0, W). \end{cases}$$

where $z_i = (a_i - b_i/2)/\omega_i$. This is a dynamic linear model and hence one may forward filter and backwards sample to draw $p(\beta|W)$ (see Frühwirth-Schnatter [1994], Carter and Kohn [1994]), a relatively fast procedure that takes $\mathcal{O}(n)$ operations, unlike the general Gaussian process case, which requires $\mathcal{O}(n^3)$ operations per sample.

We now turn to previous efforts for simulating dynamic models like the one above. Much of this work focuses on the larger class of dynamic linear models

$$\begin{cases} y_i \sim \mathbb{P}(\psi_i) \\ \psi_i = x_i \beta_i \\ \beta_i \sim \text{AR}(1), \end{cases}$$

where the response $(y_i|\psi_i)$ is drawn from an exponential family [Casella and Berger, 2002]. Such models are called dynamic generalized linear models (DGLM).

2.6.1 Previous Efforts

Bayesian inference for dynamic generalized linear models dates back to at least West et al. [1985] who used conjugate updating with backwards sampling by linear Bayes (CUBS) to sample the dynamic regression coefficients of DGLMs when

the observation $(y_i|\psi_i)$ comes from an exponential family; but their method is only approximate. Much effort has been devoted to developing exact posterior samplers, though none has proved to be completely satisfactory. A primary goal of any such sampler is to sample states jointly, like the forward filter backwards sampler (FFBS) of Frühwirth-Schnatter [1994] and Carter and Kohn [1994], since jointly sampling states tends to result in less autocorrelation than sampling the states component-wise, an approach suggested by Carlin et al. [1992] prior to the advent of the advent of the FFBS. However, the FFBS procedure requires Gaussian, linear state-space evolution equations and observation equations. Without these assumptions, as is the case with exponential families in general, the machinery of the FFBS breaks down. To resurrect the FFBS, one may approximate the posterior with some convenient proposal density and then accept or reject using Metropolis-Hastings, or one may use data augmentation so that, conditionally, the observations and states are generated by a DLM. Neither method is guaranteed to work well.

Gamerman [1998] discusses various Metropolis-Hastings based approaches, all of which rely on some Laplace-type approximation (and hence all of which can be phrased as iteratively reweighed least squares [Wedderburn, 1974]) for generating proposals. None of the various approaches is completely satisfactory: component-wise proposals have decent acceptance rates, though high autocorrelation between consecutive samples, while joint proposals suffer from unacceptably small acceptance rates. Gamerman's solution is to transform the problem so that one samples the innovation variances component-wise using a Laplace approximation with Metropolis-Hastings update, arguing that the new coordinate system possesses less intrinsic correlation.

But this approach is more computationally intensive since one must transform the proposals back to the original coordinate system at each iteration to evaluate Metropolis-Hastings acceptance probability.

Shephard and Pitt [1997] attempt to strike a balance between the autocorrelation of consecutive samples and the acceptance probabilities of proposed samples by drawing blocks of states. Sampling in blocks reduces autocorrelation while restraining the size of the blocks ensures a reasonable Metropolis-Hastings acceptance probability. However, their method still suffers from autocorrelation between consecutive draws for the hidden states.

More recently, techniques have emerged that do generate joint draws of the states. Ravines et al. [2006] built upon West et al. [1985] by adding a Metropolis-Hastings step to sample the states exactly. Though this at first would seem like a poor choice due to the high dimensionality often encountered in time series, they find that, in fact, the technique results in reasonable acceptance rates unlike a global Laplace approximation. That conjugate updating improves the Metropolis-Hastings proposal enough to allow for "efficient" joint draws is somewhat surprising.

All of the data augmentation techniques described in Section 2.1.2 for binary logistic regression, that is Holmes and Held [2006], Frühwirth-Schnatter and Frühwirth [2007], Frühwirth-Schnatter and Frühwirth [2010], and Fussl et al. [2013], can be immediately extended to dynamic binary logistic regression. The comparisons below make use of Fussl et al. [2013], since that is the most recent method from the Frühwirth-Schnatter school. The data augmentation approach of Frühwirth-Schnatter et al. [2009] can be extended to dynamic negative binomial regression for

count data.

DGLMs can be cast within the more general framework of non-linear non-Gaussian state-space models. Geweke and Tanizaki [2001] highlight the various works of Kitagawa, Tanizaki, and Mariano, among others, to filter, smooth, or simulate states within this context using numerical integration, re-sampling, or rejection sampling. However, the more general setting does not provide more insight into how one may jointly samples states in DGLMs. Each of the approaches they review is flawed: numerical integration does not work well outside of the simplest settings, sampling marginally smoothed states using sequential methods is time consuming, and rejection sampling may have poor acceptance probabilities. None of the methods cited by Geweke and Tanizaki are useful for generating posterior samples of the states jointly, an extremely desirable property. Their solution is to sample the states componentwise using a Laplace approximation and a Metropolis-Hastings step, which in the case of exponential families returns us to the methods discussed by [Gamerman, 1998]. Godsill et al. [2004] show how one may jointly sample states using particle filters; however, that approach is relatively slow, on the order of $\mathcal{O}(Mn)$ for each draw from $p(\beta|y)$ where M is the number of particles and n is the number of observations.

2.6.2 Benchmarks

We compare the Pólya-Gamma data augmentation technique using the $J^*(1, z)$ sampler for dynamic binomial logistic regression against the data augmentation strategy of Fussl et al. [2013] and the Metropolis-Hastings strategy of Ravines et al. [2006], who report a better effective sample size than Gamerman [1998]. As in the static case,

the primary metric of comparison is the median effective sampling rate. (See §2.5 for the definition of effective sample size and effective sampling rate.) However, in this case, the quantity of interest is not a p-dimensional regression coefficient, but rather a $p \times n$-dimensional dynamic regression coefficient $\{\beta_{it} : t = 1, \ldots, T, i = 1, \ldots, p\}$. Thus, the median effective sample size and median effective sampling rate is calculated over both i and t.

Three data sets comprise the suite of benchmarks. The first data set is the Tokyo rainfall data set found in Kitagawa [1987], which has 366 binomial observations, 365 of which have 2 trials, and one of which has a single trial. (There is one observation on February 29.) The dynamic regression coefficient β is given a local level prior,

$$\beta_t = \beta_{t-1} + \eta_t \,,\ \eta_t \sim N(0, W).$$

The second and third data sets consist of synthetic binary responses, periodic covariates, and prior dynamics $\beta \sim \text{AR}(1)$,

$$\beta_t = \Phi\beta_{t-1} + \eta_t \,,\ \eta_t \sim N(0, W),$$

where $\beta_t \in \mathbb{R}^2$ for $T = 500$ observations, $\Phi = 0.95I_2$, and the innovation variance is $W = 0.172I_2$.

As seen in Table 2.5, the Pólya-Gamma approach has superior effective sample size and superior effective sampling rate compared to the other methods. Unlike the static case, these models possess hyperparameters, which can dilute the differences in effective sample size. A more detailed table can be found in Appendix 4, where one finds that the CUBS method has a poor effective sampling rate due to its long run

		Tokyo Rain	Synth 1	Synth 2
ESS	Pólya-Gamma	7735	7063	4802
	PG:Fussl	2.12	1.25	1.27
	PG:CUBS	10.15	12.24	8.88
ESR	Pólya-Gamma	356	228	155
	PG:Fussl	2.01	1.29	1.30
	PG:CUBS	109.20	247.83	178.16

Table 2.5: Dynamic binary logistic regression benchmarks. As in Section , the median effective sample size and median effective sampling of $\{\beta_i\}_{i=1}^n$ has been calculated for each method. Here those quantities are reported for the Pólya-Gamma technique as well as the data augmentation scheme of Fussl et al. [2013] and the Metropolis-Hastings based approach of Ravines et al. [2006]. PG:Fussl and PG:CUBS report the ratio of the Pólya-Gamma median ESS or ESR to the ESS and ESR of each competing method.

time. This is a consequence of the way in which one forward filters under CUBS. At each time step, one must use a root finding algorithm to transform the first and second moments of $p(\beta_t|D_{t-1})$ to a new coordinate system. The run time may be improved by picking better initial conditions or loosening the conditions for convergence; options that we do not explore here. Whatever improvements one might make, the CUBS approach still has vastly inferior effective sample size compared to the Pólya-Gamma approach and so it is extremely unlikely that tweaking the root finding algorithm could result in the huge improvements necessary to even make it competitive with either data augmentation approach. Thus, in the case of dynamic binomial regression, the Pólya-Gamma method is definitively the most efficient technique.

2.7 An Alternate $J^*(h, z)$ Sampler

In Section 2.5, the $J^*(1, z)$ sampler performed well in binary logistic regression, but performed poorly for negative binomial regression, a consequence of the fact that one must sample n $J^*(1, z)$ random variates to produce a single $J^*(n, z)$ random variate for $n \in \mathbb{N}$. Here we devise an alternate algorithm that will directly draw from $J^*(n, z)$ for $n \geq [1, 4]$.

2.7.1 An Alternate $J^*(h)$ sampler

The basic strategy will be the same as in §2.3: find two functions ℓ and r such that the density f is dominated by ℓ on $(0, t]$ and r on (t, ∞). Truncated versions of ℓ and r can then be used to generate a proposal. Previously, these proposals came from the density of f, which when $h = 1$, has two infinite, alternating sum representations. Pasting together these two representations together one may immediately appeal to the von Neumann alternating sum technique to accept or reject a proposal; but this only works when $h = 1$. For $h \neq 1$, the density in Fact 2.6.2 is still valid:

$$f(x|h) = \frac{2^h}{\Gamma(h)} \sum_{n=0}^{\infty} (-1)^n \frac{\Gamma(n+h)}{\Gamma(n+1)} \frac{(2n+h)}{\sqrt{2\pi x^3}} \exp\Big(- \frac{(2n+h)^2}{2x} \Big). \tag{2.18}$$

We know that the coefficients of this alternating sum, which we call a_n^L, are not decreasing in $n \in \mathbb{N}_0$ for all $x > 0$; they are only decreasing in $n \in \mathbb{N}_0$ for x in some interval I_L. However, it is the case that $a_n^L(x|h)$ is decreasing for sufficiently large n for all $x > 0$. Thus, we may still appeal to a von Neumann-like procedure, but only once we know that we have reached an $n^*(x)$ so that $a_n^L(x|h)$ is decreasing for $n \geq n^*$. The following proposition shows that we can identify when this is the case.

Proposition 2.13. *Fix $h \geq 1$ and $x > 0$. The coefficients $\{a_n^L(x)\}_{n=0}^{\infty}$ in (2.18) are decreasing, or they are increasing and then decreasing. Further, if $a_n^L(x^*)$ is decreasing for $n \geq n^*$, then $a_n^L(x)$ is decreasing for $n \geq n^*$ for $x \leq x^*$.*

Proof. Fix $h \geq 1$ and $x > 0$; calculate $a_{n+1}^L(x|h)/a_n^L(x|h)$. It is

$$\begin{aligned}
&\frac{\Gamma(n+1)}{\Gamma(n+2)}\frac{\Gamma(n+1+h)}{\Gamma(n+h)}\frac{2n+2+h}{2n+h}\exp\Big\{-\frac{1}{2x}\Big[(2n+2+h)^2-(2n+h)^2\Big]\Big\} \\
&\qquad = \frac{n+h}{n+1}\frac{2n+h+2}{2n+h}\exp\Big\{-\frac{1}{2x}\Big[4(2n+h)+4\Big]\Big\} \\
&\qquad = \Big(1+\frac{h-1}{n+1}\Big)\Big(1+\frac{2}{2n+h}\Big)\exp\Big\{-\frac{2}{x}\Big[(2n+h)+1\Big]\Big\}.
\end{aligned}$$

Since $x > 0$, the exponential term decays to zero as n diverges and there is smallest $n^* \in \mathbb{N}_0$ for which this quantity is less than unity. Further, it is less than unity for all such $n \geq n^*$ as all three terms in the product are decreasing in n. The ratio also decreases as x decreases, thus $a_n^L(y)$ is decreasing for $n \geq n^*$ when $y \leq x$. □

Corollary 2.14. *Suppose $h \geq 1$ and $x > 0$ and let $S_n^L(x|h) = \sum_{i=0}^{n}(-1)^i a_i^L(x|h)$. There is an $n^* \in \mathbb{N}_0$ for which $f(y|h) < S_n^L(y|h)$ for all even $n \geq n^*$ and $f(y|h) > S_n^L(y|h)$ for all odd $n \geq n^*$ for $y \leq x$.*

Corollary 2.15. *There is an $x^*(h)$,*

$$x^*(h) = \sup\Big\{x : \{a_n^L(x|h)\}_{n=0}^{\infty} \text{ is decreasing } \Big\},$$

so that $\{a_n^L(x|h)\}_{n=0}^{\infty}$ is decreasing for all $x < x^*$. Thus $\ell(x|h) = a_0^L(x|h)$ satisfies

$$S_n(x|h) \leq \ell(x|h),\ \forall n \in \mathbb{N}_0, \forall x < x^*(h).$$

When $h = 1$, we have another representation of $f(x|h)$ as an infinite alternating sum. This is not the case when $h \neq 1$; however, revisiting §2.2.1, when $h \in \mathbb{N}$, we may also write $f(x|h)$ as

$$f(x|h) = \sum_{n=0}^{\infty} \Big[\sum_{m=1}^{h} \frac{A_{nm}(h-1)!}{A_{nh}(m-1)!} \frac{1}{x^{h-m}} \Big] \frac{A_{nh} x^{h-1} e^{-c_n x}}{(h-1)!}, \quad c_n = \frac{\pi^2}{2}(n+1/2)^2.$$

When x is large, the term with $m = h$ will dominate, leaving

$$\sum_{n=0}^{\infty} \frac{A_{nh} x^{h-1} e^{-c_n x}}{(h-1)!}, \; A_{nh} = (-1)^{nh}(2c_n)^{h/2}.$$

Again, since $e^{-c_n x}$ decays rapidly in n the first term of this sum should be the most important. Hence, for sufficiently large x, $f(x|h)$ should look like

$$r(x|h) = \frac{A_{0h} x^{h-1} e^{-c_0 x}}{(h-1)!} = \frac{(\pi/2)^{h/2} x^{h-1} e^{-c_0 x}}{(h-1)!}.$$

This will be the right hand side proposal.

Conjecture 2.16. *The functions $\ell(x|h)$ and $r(x|h)$ dominate $f(x|h)$ on overlapping intervals that contain a point $t(h)$.*

For $h \geq 1$, we know that $\ell(x|h)$ will dominate $f(x|h)$ on some interval $[0, x^*(h))$ from Corollary 2.15. We have not proved that $r(x|h)$ dominates $f(x|h)$ on an overlapping interval; however, we do have numerical evidence that this is the case. Let $\rho^L(x|h) = f(x|h)/\ell(x|h)$ and $\rho^R(x|h) = f(x|h)/r(x|h)$. If both $\rho^L(x|h)$ and $\rho^R(x|h)$ are less than unity on overlapping intervals, then ℓ and r dominate f on overlapping intervals. As seen in Figure 2.2, this appears to be the case for both ρ^L and ρ^R on the entire real line. In that case, ℓ and r are both valid bounding kernels and the

proposal density

$$g(x|h) \propto k(x|h) = \begin{cases} \ell(x|h), & x < t \\ r(x|h), & x \geq t. \end{cases}$$

has

$$f(x|h) \leq k(x|h) \text{ for all } x > 0;$$

further, $g(x|h)$ is a mixture

$$g(x|h) = \frac{p}{p+q}\frac{\ell(x|h)}{p} + \frac{q}{p+q}\frac{r(x|h)}{q}$$

where

$$p(t|h) = \int_0^t \ell(x|h)dx \text{ and } q(t|h) = \int_t^\infty r(x|h)dx$$

and the normalizing constant of $k(x|h)$ is $c(t|h)^{-1}$ where

$$c(t|h) = p(t|h) + q(t|h).$$

Thus, Corollary 2.14 and Conjecture 2.16 lead to the following sampler:

1. Sample $X \sim g(x|h)$
2. Sample $U \sim \mathcal{U}(0, k(X|h))$.
3. Iteratively calculate the partial sums $S_n^L(x|h)$ until
   - $S_n^L(X|h)$ has decreased from $n-1$ to n, *and*
   - $U < S_n^L(X|h)$ for odd n or $S_n^L(X|h) < U$ for even n.

Both $\ell(x|h)$ and $r(x|h)$ are kernels of known densities. In particular,

$$\ell(x|h) = \frac{2^h}{\Gamma(1)}\frac{h}{\sqrt{2\pi}}x^{-3/2}\exp\Big(-\frac{h^2}{2x}\Big),$$

is the kernel of an inverse Gamma distribution, $\text{IGa}(1/2, h^2/2)$, and

$$r(x|h) = \frac{(\pi/2)^{h/2} x^{h-1} e^{-\frac{\pi^2}{8}x}}{(h-1)!}$$

is the kernel of gamma distribution, $\text{Ga}(h, \pi^2/8)$. We can rewrite

$$\ell(x|h) = 2^h \text{IGa}(x|1/2, h^2/2)$$

to find

$$p(t|h) = 2^h \frac{\Gamma(1/2, (h^2/2)/t)}{\Gamma(1/2)}$$

where $\Gamma(a, b)$ is the upper incomplete gamma function, and we can rewrite

$$r(x|h) = (4/\pi)^h \text{Ga}(x|h, \text{rate} = \pi^2/8))$$

to find

$$q(t|h) = \left(\frac{4}{\pi}\right)^h \frac{\Gamma(h, (\pi^2/8)t)}{\Gamma(h)}.$$

Note that this provides a way to calculate $t(h)$, since we want to minimize $c(t|h) = p(t|h) + q(t|h)$. This is identical to choosing the truncation point $t(h)$ to be the point at which $\rho^L(x|h)$ and $\rho^R(x|h)$ intersect.

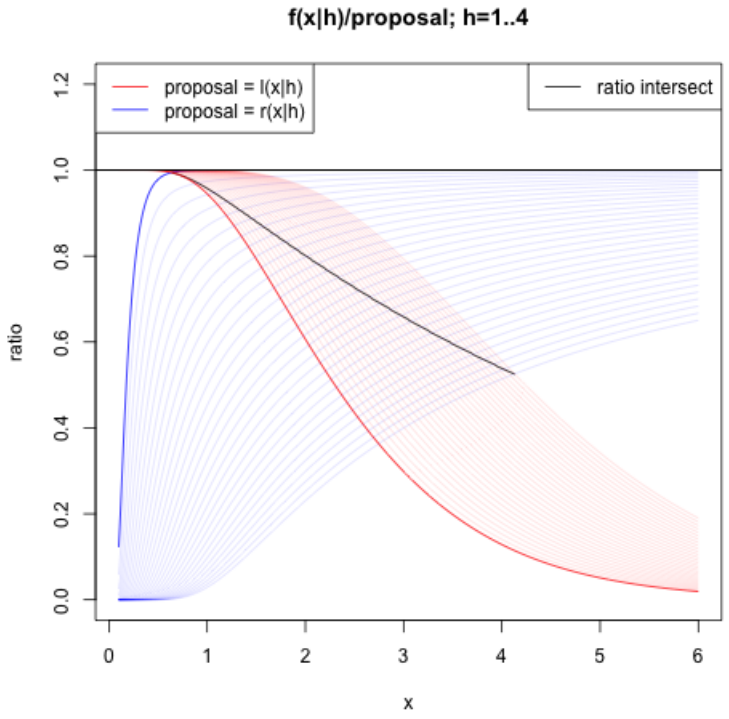
2.7.2 An Alternate $J^*(h, z)$ Sampler

Recall Fact 2.6.2, which says the density of $J^*(h, z)$ is

$$f(x|h, z) = \cosh^h(z) e^{-xz^2/2} f(x|h)$$

where $f(x|h)$ is given in (2.18). Following the general path put forth in the previous section, one finds that almost nothing changes. In particular, if we let $a_n^L(x|h, z) =$

Figure 2.2: A plot of the $f(x|h)/\ell(x|h)$ and $f(x|h)/r(x|h)$ for $h = 1.0$ to $h = 4.0$ by 0.1. The dark lines correspond to $h = 1$. The curve corresponding to ℓ increases monotonically while the curve corresponding to r decreases monotonically. The black line plots the point of intersection between the two curves as h changes.



$\cosh^h(z)e^{-xz^2/2}a_n^L(x|h)$ and let $S_n^L(x|h,z) = \sum_{i=0}^n (-1)^i a_n^L(x|h,z)$, then the analogous propositions, corollaries, and conjectures from the previous section still hold. In particular,

$$\frac{a_{n+1}^L(x|h)}{a_n^L(x|h)} = \frac{a_{n+1}^L(x|h,z)}{a_n^L(x|h,z)}$$

so Proposition 2.13, Corollary 2.14, and Corollary 2.15 hold with $a_n^L(x|h)$ replaced by $a_n^L(x|h,z)$, $S_n^L(x|h)$ replaced by $S_n^L(x|h,z)$, and $\ell(x|h)$ replaced by $\ell(x|h,z) = a_n^L(x|h,z)$. Additionally, nothing changes with regards the bounding kernel since

$$f(x|h) \le k(x|h) \iff f(x|h,z) \le k(x|h,z)$$

where

$$k(x|h,z) = \cosh^h(z)e^{-xz^2/2}k(x|h).$$

Hence the only major change is the form of the proposal density and the corresponding mixture representation. After adjusting, the left bounding kernel becomes

$$\ell(x|h,z) = \cosh^h(z)2^h \frac{h}{\sqrt{2\pi}} x^{-3/2} \exp\Big(-\frac{h^2}{2x} - \frac{xz^2}{2}\Big),$$

and the right bounding kernel becomes

$$r(x|h,z) = \cosh^h(z)\frac{(\pi/2)^{h/2}x^{h-1}}{(h-1)!}\exp\Big[-\Big(\frac{\pi^2}{8} + \frac{z^2}{2}\Big)x\Big].$$

Let

$$g(x|h,z) \propto k(x|h,z) = \begin{cases} \ell(x|h,z), & x < t(h) \\ r(x|h,z), & x \ge t(h), \end{cases}$$

and

$$p(t|h,z) = \int_0^t \ell(x|h,z)dx \;\text{ and }\; q(t|h,z) = \int_t^\infty r(x|h,z)dx.$$

Then one can represent $g(x|h,z)$ as the mixture

$$g(x|h,z) = \frac{p}{p+q}\frac{\ell(x|h,z)}{p} + \frac{q}{p+q}\frac{r(x|h,z)}{q}$$

and the normalizing constant of $k(x|h,z)$ is (suppressing the dependence on t)

$$c(h,z) = p(h,z) + q(h,z)$$

Thus, one can sample $J^*(h,z)$ by

1. Sample $X \sim g(x|h,z)$
2. Sample $U \sim \mathcal{U}(0, k(x|h))$.
3. Iteratively calculate the partial sums $S_n^L(x|h)$ until
   - $S_n^L(X|h)$ has decreased from $n-1$ to n, *and*
   - $U < S_n^L(X|h)$ for odd n or $S_n^L(X|h) < U$ for even n.

Note that the above procedure uses $k(x|h)$ and $S_n(x|h)$ instead of $k(x|h,z)$ and $S_n(x|h,z)$. This is because

$$\tilde{f}(x|h)/\tilde{g}(x|h) = \tilde{f}(x|h,z)/\tilde{g}(x|h,z)$$

and

$$\tilde{f}(x|h)/S_n^L(x|h) = \tilde{f}(x|h,z)/S_n^L(x|h,z).$$

Again, the kernels $\ell(x|h,z)$ and $r(x|h,z)$ are recognizable. The exponential term of $\ell(x|h,z)$ is

$$-\frac{z^2}{2x}\Big[(\frac{h}{z})^2 + x^2\Big].$$

Completing the square yields

$$-\frac{(z/h)^2h^2}{2x}\Big[(x-h/z)^2\Big]-zh;$$

so

$$\ell(x|h,z)=(1+e^{-2|z|})^h\frac{h}{\sqrt{2\pi x^3}}\exp\Big(-\frac{(z/h)^2h^2}{2x}\Big[(x-h/z)^2\Big]\Big),$$

which is the kernel of an inverse Gaussian distribution with parameters $\mu=h/z$ and $\lambda=h^2$. The right kernel is a gamma distribution with shape parameter h and rate parameter $\lambda_z=\pi^2/8+z^2/2$. Thus, the left hand is

$$\ell(x|h,z)=(1+e^{-2|z|})^h IG(x|\mu=h/z,\lambda=h^2)\ \text{ for } z>0$$

and

$$\ell(x|h,0)=2^h\mathrm{IGa}(x|1/2,h^2/2);$$

the right hand kernel is

$$r(x|h,z)=\Big(\frac{\pi/2}{\lambda_z}\Big)^h\mathrm{Ga}(x|h,\text{rate}=\lambda_z),\ \lambda_z=\pi^2/8+z^2/2;$$

and the respective weights are

$$p(t|h,z)=(2^he^{-zh})\Phi_{IG}(t|h/z,h^2),$$

$$p(t|h,0)=2^h\frac{\Gamma(1/2,(h^2/2)(1/t))}{\Gamma(1/2)},$$

and

$$q(t|h,z)=\Big(\frac{\pi/2}{\lambda_z}\Big)^h\frac{\Gamma(h,\lambda_z t)}{\Gamma(h)}.$$

Truncation Point

The normalizing constant $c(t|h,z)$ is

$$c(t|h,z) = \int_0^t \cosh^h(z)e^{-xz^2/2}\ell(x|h)dx + \int_t^\infty \cosh^h(z)e^{-xz^2/2}r(x|h)dx.$$

To minimize $c(t|h,z)$ over t, note that the critical points, which satisfy

$$\cosh^h(z)e^{-xz^2/2}\Big[\ell(x|h) - r(x|h)\Big] = 0,$$

are independent of z. Hence we only need to calculate the best $t = t(h)$ as a function of h.

2.7.3 Recapitulation

While the $J^*(1,z)$ sampler from §2.3 works well from binomial logistic regression, it cannot produce random variates from $J^*(h,z)$ in a single draw when $h \in \mathbb{N}\backslash\{1\}$. The sampler breaks completely when $h \notin \mathbb{N}$. In contrast, the method put forth in this section can produce draws from $J^*(h,z)$ for $h \geq 1$ if Conjecture 2.16 holds. We numerically verify this is the case for $h \in [1,4]$. In practice, to draw $J^*(h,z)$ when $h > 4$, we take sums independent J^* random variates like before. The new sampler is limited in two ways. First, the best truncation point t is a function of h, and must be calculated numerically. Second, the normalizing constant $c(h,z)$ grows as h increases. The former is not too troubling as one may precompute many $t(h)$ and then interpolate between values of h not specified. However, the latter is disturbing as $1/c(h,z)$ is the probability of accepting a proposal. Thus, as h increases the probability of accepting a proposal decreases. To address this deficiency, we devise yet another sampler.

2.8 An Approximate $J^*(b, z)$ Sampler

Daniels [1954] provides a method to construct approximations to the density of the mean of n independent and identically distributed random variables. More generally, Daniels procedure produces approximations to the density of $X(n)/n$ where $X(h)$ is an infinitely divisible family [Sato, 1999]. The approximation improves as n increases. This is precisely the scenario we are interested in addressing, as $J^*(n, z)$ is infinitely divisible and the two previously proposed samplers do not perform well when sampling $J^*(n, z)$, or equivalently $J^*(n, z)/n$, for large n.

2.8.1 The Saddle Point Approximation

The method of Daniels [1954] and variants thereof are known as saddlepoint approximations or the method of steepest decent. In addition to Daniels [1954], Murray [1974] provides an accessible explanation of the asymptotic expansion and approximation, including numerous helpful graphics. A more technical analysis may be found in the paper by Barndorff-Nielsen and Cox [1979] and the books by Butler [2007] and Jensen [1995]. McLeish [2010] provides several examples of simulating random variates following the approach of Lugannani and Rice [1980]. Below, we briefly summarize the basic idea behind the approximation following Daniels [1954].

Let $X(h)$ be an infinitely divisible family. Let $M(t)$ denote the moment generating function of $X(1)$, and let $K(t)$ denote its cumulant generating function:

$$M(t) = e^{K(t)} = \int_{-\infty}^{\infty} e^{tx} f(x) dx.$$

where $f(x)$ is the density of the random variable $X(1)$. Let $\bar{x}$ denote $X(n)/n$, which

can be thought of as the sample mean of n independent $X(1)$ random variables when n is an integer. The MGF of $\bar{x}$ is $M^n(t/n)$ and its Fourier inversion is

$$f_n(\bar{x}) = \frac{1}{2\pi}\int_{-\infty}^{\infty} M^n(it/n)e^{-it\bar{x}}dt = \frac{n}{2\pi}\int_{-\infty}^{\infty} M^n(it)e^{-int\bar{x}}dt$$

where f_n is the density of $X(n)/n$. The goal is to pick the path of this integral in a way that concentrates as much mass as possible at a single point. Changing variables to $T = it$ and phrasing this integral in terms of the cumulant generating function yields

$$f_n(\bar{x}) = \frac{n}{2\pi i}\int_{-\infty i}^{\infty i} e^{n[K(T)-T\bar{x}]}dT.$$

One can concentrate mass at $T_0 + 0i$ where T_0 is chosen to minimize

$$K(T) - T\bar{x} \text{ over } T \in \mathbb{R},$$

which will be a saddle point. Consequently, one may descend quickly in the directions perpendicular to the real axis at $T_0 + 0i$, which leads to an integral like

$$f_n(\bar{x}) = \frac{n}{2\pi i}\int_{T_0-\infty i}^{T_0+\infty i} e^{n(K(T)-T\bar{x})}dT,$$

though some care must be taken with the path of integration near T_0+0i. Performing an asymptotic expansion of $K(T)$ at T_0 and integrating yields the approximation of Daniels:

$$sp_n(\bar{x}) = \left(\frac{n}{2\pi}\right)^{1/2} K''(T_0)^{-1/2} e^{n[K(T_0)-T_0\bar{x}]};$$

note $T_0(x)$ solves

$$K'(T_0) - \bar{x} = 0. \tag{2.19}$$

Daniels [1954] (p. 639) provides conditions that ensure the approximation will hold, which in the case of the $J^*(1, z)$ distribution are

$$\lim_{u \to (\pi^2/8)^-} K_0'(u) = \infty \quad \text{and} \quad \lim_{u \to -\infty} K_0'(u) = 0$$

where $K_0(u) = \log \cos \sqrt{2u}$ is the cumulant generating function of $J^*(1)$. As seen in Fact 2.19, this is indeed the case.

2.8.2 Sampling the saddlepoint approximation

The saddlepoint approximation provides a good point-wise approximation of the density of $J^*(n, z)/n$. To make this useful for Pólya-Gamma data augmentation, we need to sample from the density proportional to $sp_n(x)$. (Henceforth we drop the bar notation for $\bar{x}$.) One general approach is to bound $\log sp_n(x)$ from above by piecewise linear functions, in which case the approximation will consist of a mixture of truncated exponentials. When the log-density is a concave functions, one is assured that such an approximation exists. Devroye provides several examples of how this may be used in practice, even for the case of arbitrary log-concave densities [Devroye, 1986, 2012].

Figure 2.3 shows an example of a piecewise linear envelope that bounds a log-concave density. One can construct such an envelope by picking points $\{x_i\}$ on the the graph of the density f, finding the tangent lines L_i at each point, and then constructing the function $e(x) = \min_i L_i(x)$, which corresponds to a piecewise linear function. It is a good idea to pick one of the points to be the mode of the density, since having $e(x) > \log f(x)$ may cause the proposal to be much larger than the density after exponentiation.

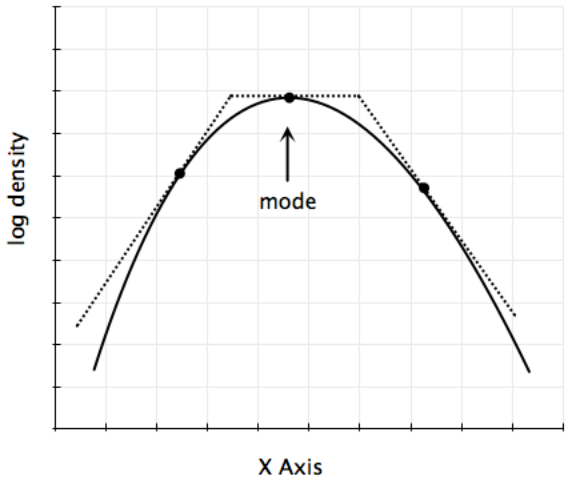


Figure 2.3: A log concave density bounded by a piecewise linear function.

We follow the piecewise linear envelope approach, though with a few modifications. In particular, we will bound the term $K(t)-tx$ found in the exponent of $sp_n(x)$ rather than the kernel itself using functions more complex than affine transforms. It will require some care to make sure that the subsequent envelope does not supersede $\log sp_n(x)$ too much. However, by working with $K(t) - tx$ directly, we avoid having to deal with the $K''(t)$ term in $sp_n(x)$, which will causes the mode of $sp_n(x)$ to shift as n changes.

Recall that t is implicitly a function of x that arises via the minimization of $K(t) - tx$ over t. This may be phrased in terms of convex duality via

$$\phi(x) = \min_{s \in \mathbb{R}} \Big\{ K(s) - sx \Big\} \tag{2.20}$$

where $K(t)$ is the cumulant generating function: $K(t)$ is strictly convex on dom $K = \{t : K(t) < \infty\}$ as $J^*(1, z)$ has a second moment [Jensen, 1995]. Using this notation,

we may write

$$sp_n(x) = \left(\frac{n}{2\pi}\right)^{1/2} K''(t(x))^{-1/2} e^{n\phi(x)}.$$

When needed, we will write $K_z(t)$ to denote the explicit dependence on z, though usually we will suppress the dependence on z. The connection to duality will help us find a good bound for $\phi(x)$; the following facts will be useful.

Fact 2.17. *Let K be the cumulant generating function of $J^*(1, z)$. Let $\phi(x)$ be the concave dual of K as in (2.20). Let*

$$t(x) = \operatorname*{argmin}_{s \in \mathbb{R}} \Big\{ K(s) - sx \Big\}.$$

Assume that when we write t we are implicitly evaluating it at x. Then

1. *$K(t)$ is strictly convex.*
2. *$K(t)$ is smooth.*
3. *$K'(t) = x$;*
4. *$\phi(x) = K(t) - tx$;*
5. *$\phi'(x) = -t$;*
6. *$\dfrac{dt}{dx}(x) = [K''(t)]^{-1}$;*
7. *As seen by item (3), $\phi'(x)$ is maximized when $t(x) = 0$. Thus,*

$$m = \operatorname*{argmax}_x \phi(x) \quad \textit{is attained when} \quad m = K'(0).$$

Proof. Barndorff-Nielsen [1978] shows that (1) holds so long as $J^*(1,z)$ has a second moment, which it does. The cumulant generating function $K(t) = -\log\cos\sqrt{2t}$ is smooth by composition of smooth functions so long as

$$\cos\sqrt{2t} = \begin{cases} \cos\sqrt{2t}, & t \geq 0 \\ \cosh\sqrt{2|t|}, & t < 0 \end{cases}$$

is smooth. For $t \neq 0$ this holds since cos and cosh are smooth and $\sqrt{2t}$ is smooth for $t \neq 0$. For $t = 0$, this follows from the Taylor expansion of cos and cosh. Items (3)-(7) are consequences of (1) and (2). □

Remark 2.18. *Sometimes it will be helpful to work with a shifted version of t: $u = t - z^2/2$. To reiterate, we will go between three different variables: x, t, and u characterized by the bijections*

1. *$x = K'(t)$ and*
2. *$u = t - z^2/2$.*

It will also be helpful to have the derivatives of K on hand and a few facts about x and u.

Fact 2.19. *Recall that $K(t) = \log\cosh(z) - \log\cos\sqrt{2u}$ is the cumulant generating function of $J^*(1,z)$. Its derivatives, with respect to t, are:*

1. $K'(t) = \dfrac{\tan\sqrt{2u}}{\sqrt{2u}}$;
2. $K''(t) = \dfrac{\tan^2(\sqrt{2u})}{2u} + \dfrac{1}{2u}\Big(1 - \dfrac{\tan\sqrt{2u}}{\sqrt{2u}}\Big)$.

Note that we are implicitly evaluating u at t as described in Remark 2.18. As shown above, $K'(t) = x$. Evaluating K'' at $t(x)$ yields

$$K''(t) = x^2 + \frac{1}{2u}(1 - x).$$

We may write $\frac{\tan\sqrt{s}}{\sqrt{s}}$ piecewise as

$$\frac{\tan\sqrt{s}}{\sqrt{s}} = \begin{cases} \frac{\tan\sqrt{s}}{\sqrt{s}}, & s > 0 \\ \frac{\tanh\sqrt{|s|}}{\sqrt{|s|}}, & s < 0 \\ 1, & s = 0. \end{cases}$$

The last fact can be seen by taking the Taylor expansion around $s = 0$. Thus, $u < 0 \iff x < 1$, $u > 0 \iff x > 1$, and $u = 0 \iff x = 1$.

This leads to the following two claims, which will help us bound the saddlepoint approximation. Notice that in each case, we adjust $\phi(x)$ to match the shape of the tails as suggested by Remark 2.7.

Lemma 2.20. *The function $\eta_r(x) = \phi(x) - (\log(x) - \log(x_c))$ is strictly concave for $x > 0$.*

Proof. Taking derivatives:

$$\eta_r'(x) = \phi'(x) - \frac{1}{x}$$

and

$$\eta_r''(x) = -\frac{dt}{dx}(x) + \frac{1}{x^2}.$$

Using Fact 2.17, this is negative if and only if

$$[K''(t)]^{-1} \geq \frac{1}{x^2} \iff x^2 \geq K''(t) \iff 0 \geq \frac{(1-x)}{2u}.$$

When $x > 1$, $u(x) > 0$, and $\eta_r''(x) < 0$. When $x < 1$, $u(x) < 0$, and $\eta_r''(x) < 0$. Continuity of K'' ensures that $\eta_r''(1) \leq 0$. □

Lemma 2.21. *The function* $\eta_l(x) = \phi(x) - \frac{1}{2}\left(\frac{1}{x_c} - \frac{1}{x}\right)$ *is strictly concave for* $x > 0$.

Proof. Taking derivatives:

$$\eta_\ell'(x) = \phi'(x) - \frac{1}{2x^2}$$

and

$$\eta_\ell''(x) = -\frac{dt}{dx}(x) + \frac{1}{x^3}.$$

Using Fact 2.17, this is negative if and only if

$$[K''(t)]^{-1} \geq \frac{1}{x^3} \iff x^3 \geq K''(t) \iff (x^2 + \frac{1}{2u})(x-1) \geq 0.$$

Again, we know that when $x > 1$, $u > 0$, and hence $\eta_l(x) < 0$. When $x < 1$ we need to show that $x^2 + 1/(2u) < 0$. This is equivalent to showing that

$$x^2 < -\frac{1}{2u} \iff 2ux^2 > -1,\ u < 0.$$

That is

$$\tan^2\sqrt{2u} > -1 \iff \tanh\sqrt{|2u|} > -1, \text{ for } u < 0,$$

which indeed holds. Thus, when $x < 1$, $\eta_l(x) < 0$. Again, continuity of K'' then ensures that $\eta_l''(1) \leq 0$. □

These two lemmas ensure the following claim.

Lemma 2.22. *Let*

$$\delta(x) = \begin{cases} \frac{1}{2}\left(\frac{1}{x_c} - \frac{1}{x}\right) & x \leq x_c, \\ \log(x) - \log(x_c), & x > x_c. \end{cases}$$

Then $\eta(x) = \phi(x) - \delta(x)$*, is continuous on* $\mathbb{R}$ *and concave on the intervals* $(0, x_c)$ *and* (x_c, ∞)*.*

We may create an envelope enclosing ϕ in the following way. See Figure 2.4 for a graphical interpretation.

1. Pick three points $x_\ell < x_c < x_r$ corresponding to left, center, and right.
2. Find the tangent lines L_ℓ and L_r that touch the graph of η at x_ℓ and x_r.
3. Construct an envelope of η using those two lines, that is

$$e(x) = \begin{cases} L_\ell(x), & x < x_c, \\ L_r(x), & x \geq x_c. \end{cases}$$

Then an envelope for $\phi(x)$ is

$$\phi(x) \leq e(x) + \delta(x).$$

Conjecture 2.23. $K''(t)/x^2$ *is increasing on* $x > 0$ *with* $\lim_{x\to 0^+} K''(t)/x^2 = 0$ *and* $\lim_{x\to\infty} K''(t)/x^2 = 1$ *and* $K''(t)/x^3$ *is decreasing on* $x > 0$ *with* $\lim_{x\to 0^+} K''(t)/x^3 = 1$ *and* $\lim_{x\to\infty} K''(t)/x^3 = 0$*.*

This can be seen by plotting these functions; however, we do not have a complete proof currently. Instead, we employ the following lemma.

Lemma 2.24. *Given $x_c \in (0,\infty)$, there are constants $\alpha_\ell, \alpha_r > 0$ such that $K''(t)$ satisfies*

$$1 \geq \frac{K''(t)}{x^3} \geq \alpha_\ell \quad \textit{for } x < x_c$$

and

$$1 \geq \frac{K''(t)}{x^2} \geq \alpha_r \quad \textit{for } x > x_c.$$

Proof. The upper bounds are verified in the proofs of Lemmas 2.21 and 2.20. For the lower bounds, recall that $K''(t(x)) > 0$ for $x \in I_M := [1/M, M]$ for any $M > 1$. Thus, $K''(t(x))$ is bounded from below on I_M. In addition, x^2 and x^3 are bounded on the same interval from above. Hence the ratios $K''(t)/x^3$ and $K''(t)/x^2$ are bounded from below on I_M and we only need to consider the tail behavior of these ratios.

Let $v(x) = 2u(x)$. When $x < 1$, $v < 0$, and $x^2|v| = \tanh^2\sqrt{|v|}$ the ratio

$$K''(t)/x^3 = \frac{1}{x} - \frac{1-x}{x(x^2|v|)} = \frac{1}{x} - \frac{1-x}{x\tanh^2\sqrt{|v|}}.$$

Employing the trigonometric identity $-\sinh^2 = 1 - \coth^2$ and writing out $x(v)$ yields

$$\frac{1}{\tanh^2\sqrt{|v|}} + \frac{1}{x}\Big(1 - \coth^2\sqrt{|v|}\Big) = \frac{1}{\tanh^2\sqrt{|v|}} - \frac{\sqrt{|v|}\cosh\sqrt{|v|}}{\sinh^3\sqrt{|v|}}.$$

As $v \to -\infty$ the first term converges to unity while the second term vanishes. Since v is an increasing function of x that diverges to $-\infty$ as $x \to 0^+$, for any $1 > \alpha_\ell > 0$, there is an $M > 1$ such that $K''(t)/x^3 > \alpha_\ell$ for $x < 1/M$.

Similarly, when $x > 1$, $v > 0$, and $x^2 v = \tan^2\sqrt{v}$ the ratio

$$K''(t)/x^2 = 1 + \frac{1-x}{x^2 v} = 1 + \frac{1-x}{\tan^2\sqrt{v}}.$$

The last term can be rewritten as

$$\frac{1-x}{\tan^2\sqrt{v}} = \frac{1}{\tan\sqrt{v}}\Big(\frac{1}{\tan\sqrt{v}} - \frac{1}{\sqrt{v}}\Big),$$

which converges to zero as $v \to (\pi/2)^{2^-}$. Since v is increasing in x and converges to $(\pi/2)^2$ as $x \to \infty$, for any $1 > \alpha_r > 0$, there is an $M > 1$ such that $K''(t)/x^2 > \alpha_r$ for $x > M$.

□

Lemma 2.22 and Lemma 2.24 give us the following proposition.

Proposition 2.25. *There exists constants $1 > \alpha_\ell, \alpha_r > 0$ such that the saddle point approximation of $J^*(n,z)/n$ is bounded by the envelope*

$$k(x|h,z) = \Big(\frac{n}{2\pi}\Big)^{1/2} \begin{cases} \alpha_\ell^{-1/2} e^{\frac{n}{2x_c}}\ x^{-3/2} \exp\Big(-\frac{n}{2x} + nL_\ell(x|z)\Big), & x < x_c \\ \alpha_r^{-1/2} x_c^n\ x^{n-1} \exp\Big(nL_r(x|z)\Big), & x > x_c, \end{cases}$$

where L_ℓ is the line touching η at x_ℓ and L_r is the line touching η at x_r. Further, L'_ℓ and L'_r are negative when $x_\ell \geq m = \operatorname{argmax}_x \phi(x)$.

Proof. Lemma 2.22 and Lemma 2.24 provide the envelope. It only remains to show that the slopes of L_ℓ and L_r are negative when $x_\ell \geq m$. Note that the concavity of ϕ ensures that $\phi'(x) \leq 0$ when $x \geq m$. Thus, in the left case, $L'_\ell(x_\ell) = \phi'(x_\ell) - \frac{1}{2x_\ell^2} < 0$. Similarly, in the right case, $L'_r(x_r) = \phi'(x_r) - \frac{1}{x_r} < 0$. □

Given the stipulation that $x_\ell \geq \operatorname{argmax}_x \phi(x)$, the left hand kernel, $k_\ell(x|h,z)$, is an inverse Gaussian kernel while the right hand kernel, $k_r(x|h,z)$, is a gamma kernel.

To see this let $\rho_\ell = -2L'_\ell(x)$ and $b_\ell = L_\ell(0)$; then the exponent of the left hand kernel is

$$nb_\ell - \frac{n\rho_\ell x}{2} - \frac{n}{2x} = \frac{-n\rho_\ell}{2x}\Big(\frac{1}{\rho_\ell} + x^2\Big) + nb_\ell.$$

Taking the first term and completing the square yields

$$\frac{-n\rho_\ell}{2x}\Big(x - \frac{1}{\sqrt{\rho_\ell}}\Big)^2 - n\sqrt{\rho_\ell}.$$

Thus

$$k_\ell(x|h,z) = \kappa_\ell\Big(\frac{n}{2\pi x^3}\Big)^{1/2}\exp\Big\{\frac{-n\rho_\ell}{2x}\Big(x - \frac{1}{\sqrt{\rho_\ell}}\Big)^2\Big\}$$

where

$$\kappa_\ell = \alpha_\ell^{-1/2} e^{\frac{n}{2x_c} + nb_\ell - n\sqrt{\rho_\ell}}$$

so k_ℓ is the kernel of an inverse Gaussian distribution with parameters $\mu = 1/\sqrt{\rho_\ell}$ and $\lambda = n$. For the right hand kernel let $\rho_r = -L'_r(x)$ and $b_r = L_r(0)$, which yields

$$k_r(x|h,z) = \kappa_r \frac{(n\rho_r)^n x^{n-1}}{\Gamma(n)} e^{-n\rho_r x}$$

where

$$\kappa_r = \Big(\frac{n}{2\pi\alpha_r}\Big)^{1/2} \frac{e^{nb_r}\Gamma(n)}{(n\rho_r)^n}$$

so k_r is the kernel of a Gamma distribution with shape n and rate $n\rho_r$. These two observations show that $g(x|h,z) \propto k(x|h,z)$ is a mixture, which can be sampled in a manner similar to the previous two algorithms.

We have yet to specify the points x_ℓ, x_c, or x_r. As mentioned at the outset, it is important to choose these points carefully so that the envelope does not exceed the target density by too much. Currently, we set x_ℓ to be the mode of ϕ. By picking

x_ℓ to match the maximum of ϕ we guarantee that the mode of $sp_n(x)$ matches the mode of $k(x|h,z)$ as $n \to \infty$. We could set $x_r = 1.2x_\ell$ and then chose x_c so that $L_\ell(x_c) = L_r(x_c)$, in which case the envelope e is continuous. When that is the case the following proposition holds. However, this requires a non-linear solve, so in practice we simply set $x_c = 1.1x_\ell$.

Proposition 2.26. *Suppose e is continuous. Let m be the maximum of $\phi(x)$. If $x_\ell = m$, then the envelope $e(x) + \delta(x)$ takes on its maximum at m as well. Further, as $n \to \infty$, the mode of the saddlepoint approximation converges to the mode of $k(x|h,z)$.*

Proof. Suppose m maximizes ϕ and $x_\ell = m$. Then

$$e'(x_\ell) + \delta'(x_\ell) = \phi'(x_\ell) = 0.$$

Since $e'(x_\ell) + \delta'(x_\ell)$ is strictly concave on $(0, x_c]$, x_ℓ must be the maximum of the left-hand portion of the envelope for ϕ. We will show that this is the only maximum by contradiction. Suppose the right-hand portion of the envelope of ϕ has a maximum at $y > x_c$. Since that portion is also strictly concave, we must have $\phi'(y) - \delta'(y) = 0 \implies \phi'(y) = \delta'(y)$. But $\phi'(y) < 0$ since $y > m$ and $\delta'(y) = 1/y > 0$, a contradiction.

To see that the modes of sp_n and $k(x|h,z)$ converge as $n \to \infty$, take the log of each. The log of the saddlepoint approximation is like

$$\phi(x) - \frac{1}{2n} \log K''(t(x))$$

while the log of the left hand kernel, where the maximum is, is like

$$e(x) + \delta(x) - \frac{3}{2n} \log x.$$

Since δ and ϕ are concave and decay faster than $\log x$ as $x \to 0^+$ and $\log x$ is increasing, we know that the argmax of each converges to m.

□

Collecting all of the above lemmas leads to the following approximate sampler of $J^*(n, z)$. Some preliminary notation: let $\phi_z(x)$ be the concave dual of $K_z(t)$; let $sp_n(x|z)$ be the saddle point approximation; and let m be the mode of ϕ_z: $m = (\tanh z)/z$.

- Preprocess.
    1. Let $x_\ell = m$, $x_c = 1.1x_\ell$, and $x_r = 1.2x_\ell$.
    2. Calculate the tangent lines of η at x_ℓ and x_r; $L_\ell(x|z)$ and $L_r(x|z)$ respectively.
    3. Construct the proposal $g(x|n, z) \propto k(x|n, z)$.
- Accept/reject.
    1. Draw $X \sim g(x|n, z)$.
    2. Draw $U \sim \mathcal{U}(0, k(X|n, z)$.
    3. If $U > sp_n(X|z)$, return to 1.
    4. Return nX.

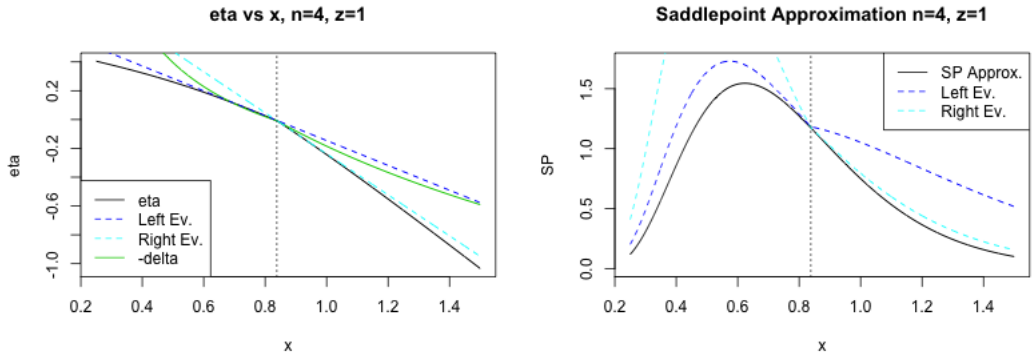


Figure 2.4: The saddlepoint approximation. The saddle point approximation is proportional to $[K''(t(x))]^{-0.5}\exp(n\phi(x))$. In the left plot, $\eta(x)$ is a solid black curve, which is bounded from above by an envelope of the dotted blue line on the left and the dotted cyan line on the right. The green line is -$\delta(x)$. On the right, the saddlepoint approximation in black, and the left and right envelopes are in blue and cyan respectively. This bound is a bit exaggerated since $n = 4$, which is rather small. The bounding envelope improves as n increases.

2.8.3 Recapitulation

The saddlepoint approximation sampler generates approximate $J^*(n, z)$ random variates when n is large, a regime that the previous two samplers handled poorly. The saddlepoint approximation sampler is similar to the previous two samplers in that the proposal is a mixture of an inverse Gaussian kernel and a gamma kernel. Hence the basic framework to simulate the approximation requires routines already developed in §2.3 and §2.7. We have identified that a good choice of x_ℓ is the mode of ϕ; however, we have not yet identified the optimal choices of x_c and x_r. The values of x_ℓ, x_c, and x_r depend on the tilting parameter z, but not the shape parameter n in $J^*(n, z)$. Thus, one could preprocess x_ℓ, x_r, and x_c for various values of z and then interpolate. We leave this task for another day.

2.9 Comparing the Samplers

We have a total of four $J^*(n, z)$ samplers available: the method from §2.3, which we call the Devroye approach, based upon sampling $J^*(1, z)$ random variates; the method from §2.7, which we call the alternate approach, that lets one directly draw $J^*(n, z)$ for $n \in [1, 4]$; the method from §2.8 using the saddlepoint approximation; and the method based upon Fact 2.6.5, where one simply truncates the infinite sum after, for instance, drawing 200 gamma random variables. Recall that to sample $J^*(n, z)$ using the $J^*(1, z)$ sampler, one sums n independent copies of $J^*(1, z)$. Similarly, to sample $J^*(n, z)$ when $n > 4$ using the alternate method, we sum an appropriate number of $J^*(b_i, z)$, $b_i \in (1, 4)$ so that $\sum_{i=1}^{m} b_i = n$.

We compare these methods empirically on a MacBook Pro with 2 GHz Intel Core i7 CPU and 8GB 1333 MHz DDR3 RAM. For a variety of (n, z) pairs, we record the time taken to sample 10,000 $J^*(n, z)$ random variates. Table 2.6 reports the best method for each (n, z) pair, along with the speed up over the Devroye approach as measured by the ratio of the time taken to draw samples using the Devroye method to the time taken to draw samples using the best method. The Devroye approach works well for $n = 1, 2$ while the alternate method works well for $n = 3, \ldots, 10$. The saddlepoint approximation works well for moderate to large n. These general observations do not change drastically across different z, though changing z can change the best sampler for fixed n. Based upon these observations, we may generate a hybrid sampler, which uses the Devroye method when $n = 1, 2$, the alternate method for $n \in (1, 13) \backslash \{1, 2\}$, the saddlepoint method when $13 \leq n \leq 170$, and a normal approximation for $n \geq 170$. The normal approximation is not strictly

necessary for large n, but the pre-built routines used to calculate the gamma function break down for $n \geq 170$. In this case, a simple fix is to calculate the mean and variance of the PG(n, z) distribution using the moment generating function from Fact 2.6, and then sample from a normal distribution by matching moments. The central limit theorem suggests that this is a reasonable approximation when n is sufficiently large.

While the alternate and approximate samplers accelerate $J^*(n, z)$ sampling, it remains to show that these new samplers improve the effective sampling rate, which is the main benchmark when comparing alternate posterior simulation techniques. Recall that Table 2.3 shows that the Pólya-Gamma method has a poor effective sampling rate compared to the method of Frühwirth-Schnatter et al. [2009] for negative binomial regression with moderate count sizes. The main bottleneck for the Pólya-Gamma technique was the generation of $J^*(n, z)$ random variates. As seen in 2.7, the hybrid sampler considerably improves the effective sampling rate for the data set named "More Counts" data set from 2.3. In particular, the run time for the Pólya-Gamma method is cut by at least half. Previously, the Pólya-Gamma technique lost to Frühwirth-Schnatter et al. [2009] for this data set; now, it wins. Thus, the new samplers do have a practical impact on the Pólya-Gamma data augmentation technique.

2.10 Recapitulation

The Pólya-Gamma data augmentation technique is useful when modeling proportions on the log-odds scale. Binary logistic regression and negative binomial regression are two prime examples that fit within this paradigm. As seen in Posterior

	Best Method						Speed-up over $J^*(1,z)$ sampler					
$n \setminus z$	0	0.1	0.5	1	2	10	0	0.1	0.5	1	2	10
1	DV	DV	DV	DV	DV	DV	1	1	1	1	1	1
2	DV	DV	AL	AL	AL	AL	1	1	1	1.08	1.08	1.22
3	DV	AL	AL	AL	AL	AL	1	1.26	1.25	1.29	1.64	1.78
4	AL	AL	AL	AL	AL	AL	1.21	1.5	1.58	1.47	1.93	2.75
10	SP	AL	AL	AL	AL	AL	1.34	1.36	1.3	1.35	1.7	2.14
12	SP	SP	SP	AL	AL	AL	1.64	1.54	1.54	1.52	1.94	2.56
14	SP	SP	SP	SP	SP	AL	1.86	1.72	1.77	1.7	1.92	2.26
16	SP	SP	SP	SP	SP	AL	2.06	1.87	2	1.93	2.21	2.57
18	SP	SP	SP	SP	SP	SP	2.27	2.07	2.17	2.15	2.46	2.42
20	SP	SP	SP	SP	SP	SP	2.51	2.25	2.35	2.36	2.69	2.74
30	SP	SP	SP	SP	SP	SP	3.68	3.36	3.57	3.36	3.92	4.05
40	SP	SP	SP	SP	SP	SP	4.68	4.41	4.57	4.48	4.99	5.51
50	SP	SP	SP	SP	SP	SP	5.83	5.16	5.55	5.55	6.11	6.78
100	SP	SP	SP	SP	SP	SP	11.07	10.4	10.66	10.44	12.22	10.45

Table 2.6: $J^*(n,z)$ benchmarks. For each method and each (n,z) pair the time taken to draw 10,000 samples was recorded and compared. The left portion of the table lists the best method for each (n,z) pair. The methods benchmarked include DV, the method from §2.3; AL, the method from §2.7; SP, the method from §2.8; and GA, an approximate draw using a truncated sum of 200 gamma random variates based upon Fact 2.6.5. Notice that the truncated sum method never wins. The DV method wins for small n; the AL method wins for modest n, and the SP method wins for medium and large n. The right hand portion of the table shows the ratio of the time taken to sample each (n,z) pair using DV to the time taken to sample using the best method.

More Counts data from Table 2.3: intercept = 3, $\bar{y} = 23.98$, $N = 400$							
Method	time	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	23.57	3127.85	3654.58	4380.93	132.68	155.02	185.83
FS	8.07	920.40	934.28	973.08	114.00	115.71	120.52

Table 2.7: A negative binomial example using the hybrid sampler. The data set is identical to that used in the "More Counts" data set from Table 2.3. Using the hybrid sampler, the Pólya-Gamma data augmentation approach wins whereas before it lost.

Calculation 2.3, the Pólya-Gamma approach is quite general and can be applied to any statistical model that has a logistic likelihood for the log odds $\{\psi_i = \beta(x_i)\}$ and a Gaussian process prior, $GP(m, K)$, for β. This framework subsumes static regression, mixed models, factor models, dynamic regression, and non-parametric regression. It is important to develop efficient schemes for posterior inference since such models arise in neuroscience, psychology, epidemiology, ecology, health care, political science, economics, and weather forecasting. The rate at which one can generate Pólya-Gamma random variates is a key factor in the efficiency of the Pólya-Gamma scheme; hence, building fast samplers is essential.

To that end, we have developed three procedures for generating random variates from the $J^*(h, z)$ family of distributions and, consequently, from the $\text{PG}(h, z)$ family of distributions. Each algorithm excels in various portions of the distribution's parametric space. Taken together, the three procedures comprise an efficient $\text{PG}(h, z)$ sampler. Prior to the development of these samplers, the Pólya-Gamma technique used the sum-of-gammas representation (Fact 2.6.5) to make approximate draws from $J^*(h, z)$. However, this method performs poorly and without the faster samplers presented in this thesis, the Pólya-Gamma technique would not out-perform other data augmentation approaches. With these samplers, the Pólya-Gamma technique outperforms O'Brien and Dunson [2004], Holmes and Held [2006], Frühwirth-Schnatter and Frühwirth [2007], Frühwirth-Schnatter and Frühwirth [2010], Gramacy and Polson [2012], and Fussl et al. [2013] for binary and binomial logistic regression. It out-performs Frühwirth-Schnatter et al. [2009] for negative binomial regression when working with small to moderate count sizes.

Beyond considerations of efficiency, the Pólya-Gamma technique is easy to implement and interpret. Unlike other data augmentation schemes, the PG technique requires one layer, as opposed to two layers, of latent variables, and hence requires keeping track of only two, as opposed to three, conditional distributions for Gibbs sampling. Further, unlike Metropolis-Hastings based approaches, it does not require one to select and then tune a proposal. Of the PG method's two complete conditionals, one is Gaussian and can be interpreted as the posterior of a normal linear model, familiar territory for must users. The other is simply a draw from the Pólya-Gamma distribution. Thus, the *only* real prerequisite for using the Pólya-Gamma technique is that one be able to simulate Pólya-Gamma random variates. This thesis provides such samplers. An accompanying R Package, `BayesLogit` [Polson et al., 2013a], implements these samplers and is freely available from the Comprehensive R Archive Network. The package includes routines for binary logistic regression and multinomial regression in addition to Pólya-Gamma random variate sampling.

Chapter 3

Forecasting High-Dimensional, Time-Varying Variance-Covariance Matrices with High-Frequency Data

Financial theory suggests that asset returns are driven by a few common sources of variation. Thus, many models decompose returns into a linear combination of common factors that are shared across all stocks and idiosyncratic factors that are unique to each asset. For instance, the Capital Asset Pricing Model (CAPM) [Sharpe, 1964] decomposes asset returns using a single common factor, which is taken to be the returns on the market-portfolio. Sharpe presents an economic argument; however, from a simplified statistical point of view we may consider his model as a collection of simple linear regressions over the $i = 1, \ldots, n$ assets for times $t = 1, \ldots, T$,

$$r_{it} = \beta_i r_{M,t} + \varepsilon_{it},\ \varepsilon_{it} \sim N(\alpha_i, \sigma_i^2).$$

One may add additional common factors by including additional regressors, for instance, market capitalization, earnings per price, or book value to market value. This information may be incorporated in various ways [Fama and French, 1993, Rosenberg and McKibben, 1973], but the basic idea is the same: regress asset returns on a set of *known* factors.

It is possible to follow the same line of reasoning but without directly specifying

what the factors represent, in which case the factors are *unknown* quantities that must be estimated from the data. A latent factor model, called factor stochastic volatility (FSVol), takes the form

$$r_{it} = \sum_{k=1}^{p} x_{ik} f_{kt} + \varepsilon_{it}, \varepsilon_{it} \sim N(0, \psi_{it})$$

for $i = 1, \ldots, n$ where $\{f_{it}\}_t, i = 1, \ldots, p$ are the common factors and $\{\varepsilon_{it}\}_t, i = 1, \ldots, n$ are the idiosyncratic factors. If we assume that the assets' mean daily return is zero, which is statistically justifiable, then both the common and idiosyncratic factors can be modeled as white noise processes with slowly changing variances. (White noise is sequence of uncorrelated, mean zero random variables [Hamilton, 1994].) Such a model is useful for forecasting the variance-covariance structure of asset returns. In particular, FSVol induces a specific structure on the daily (i.e. conditional upon parameters that change daily) covariance matrix of r_t,

$$\Sigma_t = X F_t X' + E_t \tag{3.1}$$

where F_t and E_t are diagonal matrices. Covariance matrices of this form have fewer degrees of freedom than arbitrary symmetric, positive definite matrices and thus the dimensionality of the problem is reduced, facilitating estimation and prediction.

Factor stochastic volatility is used to model a variety of financial time series and phenomenon, including foreign exchange returns [Pitt and Shephard, 1999, Aguilar and West, 2000, Lopes and Carvalho, 2007, Zhou et al., 2012, Nakajima and West, 2012], contagion across markets [Lopes and Migon, 2002], equity returns [Carvalho et al., 2011], and interest rates [Hays et al., 2012]. All of these assets have the

characteristic that a few shared components control the covariation. Until recently, such models were employed mostly for daily returns or returns calculated over an even longer period. But such models ignore a significant amount of data since financial assets are traded throughout the day.

In recent decades, the ability to record and process vast quantities of data have enabled statisticians and econometricians to make use of this intraday data for estimation and prediction of daily measures of variation and covariation. In particular, the theory of stochastic processes provides justification for constructing measures of variation and covariation for time series that are indexed on a continuum. Many financial assets are traded often enough that one may make use of these theoretical quantities to construct "high-frequency" statistics, also called realized measures. The high-frequency statistic we are most interested in is *realized covariance*, which is a high-frequency analog to the daily covariance matrix.

Koopman et al. [2005] have shown that high-frequency statistics provide superior estimates and forecasts of daily variance compared to several low-frequency alternatives in the univariate case. Liu [2009] has shown that many low-frequency methods for estimating and forecasting daily covariance matrices are inferior to forecasts generated by exponentially smoothing realized covariance matrices. His analysis considers a relatively high-dimensional setting, in which one wants to mimic a market portfolio using the 30 assets of the Dow Jones Industrial Average. However, Liu's analysis focuses on frequentist techniques for covariance estimation and he does not consider factor stochastic volatility.

We build upon Liu's work and have made three contributions to this discourse.

First, we show that forecasts generated by factor stochastic volatility are inferior to forecasts generated by exponentially smoothing realized covariance matrices. Second, to improve FSVol, we develop FSVol-like models that a incorporate information from realized covariance matrices. One such extension out-performs the original FSVol model; however, like the original model, that extension cannot compete with exponentially smoothing the high-frequency statistics. Third, we construct a model for matrix-variate data that produces identical forecasts to the aforementioned exponential smoothing procedure, thus wrapping an ad hoc technique in a statistical model. Though we do not follow Liu exactly, we share his goal of comparing models in an economically meaningful way. In particular, to compare FSVol, our extensions to FSVol, exponential smoothing, and our matrix-variate model, we construct one-day ahead portfolios for each approach and evaluate the performance of these portfolios over the course of several months.

The outline of this chapter is as follows. We review factor stochastic volatility and realized covariance (§3.1). We then develop extensions to factor stochastic volatility that incorporate information form high-frequency statistics and compare these extensions to exponentially smoothed realized covariance matrices (§3.2 and §3.4). After finding that these extensions are still inferior we build a matrix-variate model (§3.7).

3.1 Model Setup

3.1.1 Stochastic Volatility

The daily returns of financial assets display four prominent features: (1) the marginal mean of the returns is minuscule compared to the marginal variance; (2) the returns are uncorrelated; (3) the returns are heavy tailed; and (4) the variance of the returns appears to change slowly over time. These features suggest modeling daily returns as heteroscedastic white noise. Stochastic volatility [Taylor, 1982, Jacquier et al., 1994, Kim et al., 1998] is one attempt in this direction. In the basic stochastic volatility (SV) model, the returns are normally distributed given the log-variance h_t:

$$(x_t|h_t) \sim N(0, e^{h_t});$$

and the log-variance evolves as an AR(1) process [Hamilton, 1994]:

$$h_t = \mu + \phi(h_{t-1} - \mu) + \omega_t,\ \omega_t \sim N(0, W).$$

The parameter ϕ is usually close to 1 which induces a high-degree of autocorrelation in $\{h_t\}$ so that the log-variance meanders slowly about μ. One can extend this approach to a multivariate time series:

$$\begin{cases} x_{it} \sim N(0, e^{h_{it}}), & i = 1, \ldots, n; \\ h_t = \mu + \phi \odot (h_{t-1} - \mu) + \omega_t, & \omega_t \sim N(0, W); \end{cases}$$

where μ and ϕ are vectors, W is a covariance matrix, $\odot$ denotes the Hadamard product, and n is the dimension of x_t. We write $x_t \sim SV(\mu, \phi, W)$ when x_t is a stochastic volatility process. Unfortunately, the multivariate version of stochastic volatility cannot capture interesting correlation structures between daily returns as

the conditional and marginal covariance structure of x_t is diagonal:

$$\text{Var}(x_t|h_t)_{ij} = \begin{cases} \exp(h_{it}), & i = j \\ 0, & i \neq j \end{cases}$$

so

$$\text{Var}(x_t) = \text{Var}(\mathbb{E}(x_t|h_t)) + \mathbb{E}(\text{Var}(x_t|h_t)) = \mathbb{E}[\text{diag}\{\exp(h_{it})\}_{i=1}^{p}].$$

as $\mathbb{E}(x_t|h_t) = 0$.

3.1.2 Factor Stochastic Volatility

Factor stochastic volatility fuses stochastic volatility and factor models to produce time series with non-trivial covariance structures that still possess features (1) through (4) from the preceding section. A (static) factor model is a dimensionality reduction technique for modeling data jointly [Basilevsky, 1994], as opposed to conditionally. Suppose one has a collection of independent and identically distributed observations, $x_i \sim N(0, \Sigma)$, $i = 1, \ldots, T$ where $x_i \in \mathbb{R}^n$. The matrix Σ has $n(n+1)/2$ free parameters. When the there is little data relative to the degrees of freedom in Σ, that is when n is not much smaller than T, Σ is difficult to estimate. However, if one has reason to believe that the covariation in x_i is driven by a few common factors, then one can reduce the degrees of freedom in the system and improve estimation. In particular, if

$$x_i = Xf_i + \varepsilon_i,$$

where $f_i \sim N(0, I_p)$ and $\varepsilon_i \sim N(0, \Psi)$ are both independent and identically distributed, then the implied covariance structure of x_i is

$$\text{Var}(x_i) = XX' + \Psi.$$

When X is low rank the degrees of freedom are reduced, facilitating estimation of $\text{Var}(x_t)$. Factor stochastic volatility generalizes the factor model above by letting $\{f_i\}$ or $\{\varepsilon_i\}$ have some sort of inter-observation dependence. Specifically, $\{f_i\}$ and $\{\varepsilon_i\}$ are taken to be multivariate stochastic volatility processes.

Suppose $\{r_t\}_{t=1}^T$ is a collection of n-dimensional daily asset returns. Then r_t follows a factor stochastic volatility process, denoted $r_t \sim \text{FSV}$ [Aguilar, 1998, Lopes and West, 2004, Chib et al., 2006, 2009], if

$$\begin{cases} r_t = X f_t + \varepsilon_t \\ f_t \sim SV(\mu^f, \phi^f, W^f) \\ \varepsilon_t \sim SV(\mu^\varepsilon, \phi^\varepsilon, W^\varepsilon) \end{cases}$$

where X is an $n \times p$ dimensional matrix called the factor loadings, f_t is a p-dimensional stochastic volatility process, $\{f_{it}\}_t$ is called the ith factor, and ε_t is an n-dimensional stochastic volatility process called the idiosyncratic noise that has diagonal W^ε. The conditional variance is

$$\text{Var}(r_t | X, h_t^f, h_t^\varepsilon) = X F_t X' + \mathcal{E}_t \tag{3.2}$$

where F_t and $\mathcal{E}_t$ are diagonal with

$$F_{iit} = e^{h_{it}^f}, i = 1, \ldots, p, \quad \text{and} \quad \mathcal{E}_{iit} = e^{h_{it}^\varepsilon}, i = 1, \ldots, n.$$

Thus, in FSVol the covariation between assets is completely determined by the factor loadings X and the factor log-variances h_t^f. By modeling f_t and ε_t as stochastic volatility processes we replicate the characteristic white noise with slowly evolving variance found in financial asset returns.

3.1.3 Posterior Inference for FSVol

Posterior inference for factor stochastic volatility proceeds using standard Markov chain Monte Carlo techniques [Aguilar, 1998]. However, there are two points worth noting. First, one must constrain the factor loadings X for purposes of identification. In particular, one can transform f_t using a unitary matrix Γ, $\tilde{f}_t = \Gamma' f_t$, to produce a model indistinguishable from the original:

$$r_t = X f_t + \varepsilon_t = \tilde{X} \tilde{f}_t + \varepsilon_t,$$

where $\tilde{X} = X\Gamma'$. We follow Aguilar [1998] and constrain X to be unit lower triangular to identify the model.

Second, we use an approximate version of stochastic volatility to facilitate posterior inference. Consider a univariate stochastic volatility process $x_t \sim SV(\mu, \phi, W)$ and let $h_t = \log \mathrm{Var}(x_t)$, $t = 1, \ldots, T$ be the log-variance of x_t. The conditional distribution $(\{h_t\}_{t=0}^T | \{x_t\}_{t=1}^T, \mu, \phi, W)$ is not easy to simulate and hence we appeal to a standard data augmentation trick [Kim et al., 1998, Frühwirth-Schnatter, 2007]. In particular, one can transform x_t to y_t by $y_t = \log(x_t^2)$ to take

$$x_t \sim N(0, e^{h_t}) \;\text{ to }\; y_t \sim h_t + \log(\chi_1^2),$$

which makes y_t linearly related to the log-variance h_t. The innovation $\log(\chi_1^2)$ can be approximated as a discrete mixture of normals

$$\log(\chi_1^2) \simeq N(m_\gamma, v_\gamma)$$

where $\gamma \sim$ MN(1,w) and w, m, and v are known vectors. The approximation of

$x_t \sim SV(\mu, \phi, W)$ is then

$$\begin{cases} y_t = \log x_t^2 & \\ y_t = h_t + \nu_t & \nu_t \sim N(m_{\gamma_t}, v_{\gamma_t}), \\ \gamma_t \sim \text{MN}(1, w), \ i = 1, \ldots, n, & \\ h_t = \mu + \phi(h_{t-1} - \mu) + \omega_t, & \omega_t \sim N(0, W). \end{cases}$$

Conditional upon $\{\gamma_t\}$, $\{y_t\}$ is the response from a dynamic linear model, in which case $\{h_t\}$ may be sampled using forward filter backwards sampling [Frühwirth-Schnatter, 1994, Carter and Kohn, 1994]. The trick works for multivariate stochastic volatility as well.

3.1.4 Realized Covariance

The discrete time stochastic volatility and factor stochastic volatility models discussed above are appropriate when working with daily returns. However, prices are observed more than once per day. In fact, the price of a liquid stock is updated often enough that one may appeal to the theory of continuous time stochastic processes for insight. To that end, suppose $\{S_t\}_{t \geq 0}$ is a continuous time stochastic process representing the the price of a stock, where t is measured in days, and R_t is the cumulative log return of the stock: $R_t = \log S_t - \log S_0$. Define the δ-log-returns as $r_t(\delta) = R_t - R_{t-\delta}$. The day-$t$ *realized variance* of the log-returns process $\{R_t\}$ using an δ-spaced grid is

$$RV_t(\delta) = \sum_{\delta i \in (t-1, t]} r_{\delta i}^2(\delta).$$

In other words, the day-t realized variance is the sum of intraday squared returns. Though this is phrased in terms of a 24 hour day, we assume that the price of the

stock changes only when the markets are open and hence this can be thought of as the realized variance within the trading day.

One can devise reasonable models in which the realized covariance reconciles with the conditional variance of daily returns. For instance, Barndorff-Nielsen and Shephard [2004] show that if the log-returns are an Itō process

$$R_t = \int_0^t \alpha_t dW_t$$

where $\{\alpha_t\}$ is independent from $\{W_t\}$, then the day-t quadratic variation

$$\langle R \rangle_t - \langle R \rangle_{t-1} := \lim_{\delta \to 0} RV_t(\delta)$$

is identical to the daily variance σ_t^2, that is, the distribution of the daily returns conditional upon σ_t are

$$\begin{cases} r_t(1) = \sigma_t \varepsilon_t, & \varepsilon_t \sim N(0,1), \\ \sigma_t^2 = \lim_{\delta \to 0} RV_t(\delta). \end{cases}$$

While this derivation is specific to the modeling assumptions, Andersen et al. [2001] show empirically that this holds approximately.

When working with time series recorded at a high-frequency, one observes at the end of day t $RV_t(\delta)$ where δ is near 0; hence one expects, given the modeling assumptions above, that $RV_t(\delta) \simeq \sigma_t^2$. In practice, one must account for various forms of noise in the system, in which case the prices observed at the finest scale are perturbed versions of some "true" latent price. This fine-scale noise is attributed to market microstructure such as the bid-ask bounce, "refreshing" prices, and the discreteness of prices Zhang et al. [2005]. Naively summing intraday returns at the

highest possible frequency accumulates this noise and results in estimates that appear to diverge, or at least differ greatly from low-frequency estimates of the daily variance. Initially, statisticians used intraday returns at "safe" frequencies [Andersen et al., 2003] to avoid this problem. More complicated estimators of the daily quadratic variation use all intraday data, but adjust for the market microstructure noise [Ait-Sahalia et al., 2011, Barndorff-Nielsen et al., 2008].

The same theory carries over when working with multivariate processes to produce measures of covariation. In particular, the high-frequency analog of a daily covariance matrix is called *realized covariance* and is constructed like realized variance, but using the outer product, instead of the square, of intraday returns:

$$RC_t(\delta) = \sum_{\delta i \in (t-1,t]} r_{\delta i}(\delta) r_{\delta i}(\delta)'$$

where now R_t and $r_t(\delta) = R_t - R_{t-\delta}$ are vectors. $RC_t(\delta)$ converges to the daily multivariate quadratic variation, $\langle R \rangle_t - \langle R \rangle_{t-1}$. As before, market microstructure noise infects the fine-scale returns and one cannot naively sum the intraday outer product of returns to yield a reasonable approximation. To that end, we use of Barndorff-Nielsen et al.'s multivariate realized kernel [Barndorff-Nielsen et al., 2011], denoting the estimates of the daily quadratic covariation by RK_t and calling the collection $\{RK_t\}_{t=1}^T$ the realized kernels.

3.1.5 Exponential Smoothing Realized Kernels

Stochastic volatility and factor stochastic volatility are model based approaches to prediction. However, one may predict future responses using ad hoc forecasting

procedures as well. For time series, exponential smoothing is a popular technique for producing such forecasts. Exponential smoothing refers to a moving weighted average where the weights decay geometrically, $w_i \propto \lambda^i$ [Montgomery et al., 1990]. One can recursively construct an approximation to this weighted average, S_t, by averaging the most recent weighted average with a new observation. One-step ahead forecasts can be made by predicting that the value of interest tomorrow will be the weighted average of today. In the case of realized kernels, that is

$$\begin{cases} S_t = \lambda S_{t-1} + (1-\lambda) RK_t \\ \hat{\Sigma}_t = S_{t-1}. \end{cases} \tag{3.3}$$

The quantity S_t is "adapted" to the data in the sense that it only uses information acquired up to and including time t; further, $\hat{\Sigma}_t$ is a proper prediction because it only uses information acquired prior to time t. Since there is no statistical model, one must find alternate ways to pick the parameter λ. A general approach is to pick some measure of discrepancy between the forecasted value and the observed value and then minimize the average empirical discrepancy over an in-sample set.

One note on terminology is in order. Exponential smoothing refers to a weighted average of past observations to produce a single point estimate or point forecast. "Smoothing" in this sense refers to averaging. In the context of state-space models, "smoothing" may refer to a retrospective distribution, such as $p(\theta_{t-k}|D_t)$ where $\{\theta_t\}$ are the hidden states and $D_t = \{y_1, \ldots, y_t\}$ is the data up till time t, or some joint retrospective distribution such as $p(\{\theta_t\}_{i=1}^t|D_t)$.

3.2 Extensions to Factor Stochastic Volatility

As mentioned in the introduction, it is common to model the returns of many assets using a few explicit or implicit factors [Sharpe, 1964, Fama and French, 1993, Rosenberg and McKibben, 1973, Pitt and Shephard, 1999, Aguilar and West, 2000, Lopes and Carvalho, 2007, Zhou et al., 2012, Nakajima and West, 2012, Lopes and Migon, 2002, Carvalho et al., 2011, Hays et al., 2012]. Simultaneously, high-frequency statistics have been shown to be useful in univariate and multivariate forecasts of volatility [Koopman et al., 2005, Liu, 2009]. Thus, we would like to find hybrid models that incorporate information from the realized kernel while preserving a factor structure and its corresponding interpretation. Here, we construct such models, taking information from realized kernels and inserting it into the factor stochastic volatility model as exogenous data.

3.2.1 Factor "Decomposition"

To motivate our approach, we consider a factor decomposition of covariance matrices. Recall that the conditional structure of the covariance matrix in FSVol is given by (3.2):

$$\mathrm{Var}(r_t|h_t^f, h_t^\varepsilon, X) = X F_t X' + \mathcal{E}_t$$

where $F_t = \mathrm{Var}(f_t|h_t^f)$ and $\mathcal{E}_t = \mathrm{Var}(\varepsilon_t|h_t^f)$. If we make the simplifying assumption that intraday returns $r_t(\delta)$ (as defined in §3.1.4) are independent and identical normal distributions with mean zero and constant variance given $(h_t^f, h_t^\varepsilon, X)$, then

$$\Sigma_t = \mathrm{Var}(r_t|h_t^f, h_t^\varepsilon, X) = \mathrm{Var}\Big(\sum_{\delta i \in (t-1,t]} r_{\delta i}(\delta) \Big| h_t^f, h_t^\varepsilon, X \Big)$$

and the distribution of the realized covariance is

$$\sum_{\delta i \in (t-1,t]} r_{\delta i}(\delta) r_{\delta i}(\delta)' \sim W_m(1/\delta, \Sigma_t).$$

Thus we should have

$$RK_t \simeq RC_t(\delta) \sim W_m(1/\delta, X F_t X' + \mathcal{E}_t),$$

in which case one may estimate the values X, F_t, and $\mathcal{E}_t$ by maximum likelihood using data RK_t. Denote these point estimates by $\hat{X}_t$, $\hat{F}_t$, and $\hat{\mathcal{E}}_t$ for day t (see Ch. 4 from Basilevsky [1994]). The quantities $\hat{X}_t$, $\hat{F}_t$, and $\hat{\mathcal{E}}_t$ suffer from modeling error, since intraday returns are not iid, as well as sampling error. Accounting for both sources of error, the estimators are some perturbed versions of the true values. For instance, we could say that

$$\hat{X}_t = X + \text{noise}$$

and $\hat{F}_t = F_t \cdot \text{noise}$ where "noise" denotes some unspecified distribution. The latter is additive on the log-scale,

$$\log \hat{F}_{iit} = \log F_{iit} + \text{noise} = h_{it}^f + \text{noise}$$

which is be convenient for modeling purposes. Recall that X and h_t^f determine the conditional covariance structure of the returns r_t and hence we may benefit by incorporating $\log \hat{F}_{iit}$ and $\hat{X}_t$ into factor stochastic volatility models to track h_{it} and X respectively. In this way, we follow a two stage approach to avoid considering a complicated likelihood by first modeling RK_t to extract $\tilde{X}_t$ and $\tilde{F}_t$ and then using those point estimates as exogenous data for factor stochastic volatility models.

3.2.2 Factor Log-Variances

As suggested by the factor decomposition of RK_t above, $v_{it} = \log \hat{F}_{iit}$, should be related to the log-variances h_t^f. We may incorporate this new data by adding an additional observation of h_t^f, to the stochastic volatility model for f_t,

$$\begin{cases} f_{it} = N(0, e^{h_{it}}), & i = 1, \ldots, p, \\ v_t = h_t^f + \eta_t, & \eta_t \sim N(a, B) \\ h_t^f = \mu + \phi \odot (h_{t-1}^f + \mu) + \omega_t, & \omega_t \sim N(0, W). \end{cases}$$

In general, one need not take v_t to be $\{\log \hat{F}_{iit}\}_i$. Instead, one might take v_t to be some other source of information that informs the log-variances. For instance, v_t could be the first p eigenvalues of RK_t.

3.2.3 Factor Loadings

In classic FSVol, it is difficult, though possible, to let the factor loadings vary in time [Lopes and Carvalho, 2007]. The realized kernel offers a direct route less sensitive dynamic factor loadings. As before suppose one has data $\tilde{X}_t$ that informs the factor loadings X, or letting the loadings evolve in time, X_t. One can incorporate this new information by changing the factor stochastic volatility model so that

$$\begin{cases} r_t = X_t f_t + \varepsilon_t, & f_t, \varepsilon_t \sim SV \\ \tilde{x}_t = x_t + \eta_t, & \eta_t \sim N(0, B) \\ x_t = x_{t-1} + \omega_t & \omega_t \sim N(0, V^\theta). \end{cases}$$

where $\tilde{x}_t = \text{vecl } \tilde{X}_t$, $x_t = \text{vecl } X_t$, and vecl vectorizes the lower triangular portion of a matrix. The matrix valued data need not come from the factor decomposition above, for instance, one might take the first few columns of the Cholesky decomposition of RK_t to inform Θ_t.

Alcoa (AA)	American Express (AXP)	Boeing (BA)	Bank of America (BAC)	Caterpillar (CAT)
Cisco (CSCO)*	Chevron (CVX)	Du Pont (DD)	Disney (DIS)	General Electric (GE)
Home Depot (HD)	Hewlett-Packard (HPQ)	IBM (IBM)	Intel (INTC)*	Johnson & Johnson (JNJ)
JP Morgan (JPM)	Kraft (KFT)	Coca-Cola (KO)	McDonald's (MCD)	3M (MMM)
Merck (MRK)	Microsoft (MSFT)*	Pfizer (PFE)	Proctor & Gamble (PG)	AT&T (T)
Traveler's (TRV)	United Technologies (UTX)	Verizon (VZ)	Walmart (WMT)	Exxon Mobil (XOM)

Table 3.1: The thirty stocks which make up the data set. The asterisk denotes companies whose primary exchange is the NASDAQ. All other companies trade primarily on the NYSE. Taken from Windle and Carvalho [2012].

3.3 Data, Evaluation, and Computation

To assess the models presented in §3.2.2 and §3.2.3 we compare their predictive performance to the classic factor stochastic volatility model and to exponentially smoothed realized kernels.

3.3.1 Data

The data set follows the 30 stocks in the Dow Jones Industrial Average (as of October, 2010; see Table 3.1) for $T = 927$ days of trading beginning on February 27, 2007 and ending on October 29, 2010. The daily returns $\{r_t\}$ are calculated using the open-to-close log return, $r_{it} = \log P_{it,\text{close}} - \log P_{it,\text{open}}$, where $i \in 1, \ldots, n$ corresponds to the ith stock and $t \in 1, \ldots, 927$ corresponds to the t-th trading day. We construct the realized kernels using intraday transaction data provided by the Trades and Quotes (TAQ) database, which records the tick-by-tick prices at which a security was bought or sold[1]. Details on how the data was cleaned and how the realized kernels were constructed can be found in Appendix 5.

[1]Wharton Research Data Services (WRDS) was for gathering and processing data used for these benchmarks. This service and the data available thereon constitute valuable intellectual property and trade secrets of WRDS and/or its third-party suppliers

3.3.2 Evaluation

We compare factor stochastic volatility-like models and forecasts based on exponentially smoothing realized kernels using two measures. The primary metric is the empirical standard deviation of the one-step ahead minimum variance portfolio returns, which assesses a model's ability to hedge risk among a class of similarly risky assets. The one-step ahead minimum variance portfolio is

$$\pi_t = \underset{\|\xi\|_1=1}{\operatorname{argmin}} \operatorname{Var}[\xi' r_t | D_{t-1}]$$

where $D_{t-1} = \{r_1, \ldots, r_{t-1}\}$ is the data observed up to and including time $t-1$. When $\mathbb{E}(r_t|\Sigma_t) = 0$, as is the case with the FSVol-like models, the one-day ahead minimum variance portfolio can be calculated as

$$\pi_t = \underset{\|\xi\|_1=1}{\operatorname{argmin}} \xi \hat{\Sigma}_t \xi \text{ where } \hat{\Sigma}_t = \mathbb{E}[\Sigma_t | D_{t-1}].$$

In the case of FSVol with static factor loadings, $\Sigma_t = X F_t X' + \mathcal{E}_t$. In the case of dynamic factor loadings, $\Sigma_t = \Theta_t F_t \Theta_t' + \mathcal{E}_t$. In the case of exponential smoothing, $\hat{\Sigma}_t = S_{t-1}$ as in (3.3). The aggregate loss is then the empirical standard deviation of the minimum variance portfolios

$$L_1(\{\hat{\Sigma}_t\}, \{r_t\}) = \operatorname{sd}\{\pi_t' r_t\}_{t=1}^T, \tag{3.4}$$

where $\pi_t = \underset{\|\xi\|_1=1}{\operatorname{argmin}} \xi \hat{\Sigma}_t \xi$. An alternative measure of performance is the log-(Gaussian)-likelihood,

$$L_2(\{\hat{\Sigma}_t\}, \{r_t\}) = \sum_{t=1}^{T} -\frac{1}{2} \log |\hat{\Sigma}_t| - \frac{1}{2} r_t' \Sigma_t^{-1} r_t. \tag{3.5}$$

We treat the predictive log-likelihood as a check upon the primary metric.

3.3.3 Prediction

To to generate one-step ahead forecasts for exponential smoothing we split the data set into an initialization set, $t = 1, \ldots, 50$; an in-sample set, $t = 51, \ldots, 100$; and an out-of-sample set, $t = 101, \ldots, 920$. The initialization set is used to pick S_{50} to set up the exponential smoother. Next, we pick λ to minimize the loss L_1 over the in-sample set using predictions generated from (3.3). The parameter λ is then fixed to exponentially smooth the out-of-sample set using (3.3).

Estimation and prediction for the FSVol-like models is more complicated. Both FSVol and the FSVol-like models may be decomposed into convenient conditional densities for Gibbs sampling [Aguilar, 1998]. Some constrained parameters, such as ϕ^f and ϕ^ε require a Metropolis-within-Gibbs step. We want to produce forecasts

$$\hat{\Sigma}_t = \mathbb{E}[X F_t X' + \mathcal{E}_t | D_{t-1}]$$

for $t = 101, \ldots, 920$ where F_t and $\mathcal{E}_t$ are diagonal with $F_{iit} = e^{h^f_{it}}, i = 1, \ldots, p$ and $\mathcal{E}_{iit} = e^{h^\varepsilon_{it}}, i = 1, \ldots, n$. To calculate a single point estimate we must simulate the joint distribution

$$p(X, h^f_t, h^\varepsilon_t | D_{t-1})$$

which may be produced using the posterior distribution

$$p(X, h^f_{t-1}, \mu^f, \ldots, h^\varepsilon_{t-1}, \mu^\varepsilon, \ldots | D_{t-1}).$$

To compute a one-step ahead forecast of $(\Sigma_t | D_{t-1})$ for $t \in 101, \ldots, 920$, one must re-run our Gibbs sampler at each time step $t \in 101, \ldots, 920$. Thus *for each* factor-like model with 1, 2, or 3 factors we generate 1500 samples, discarding the first 500 as

burn-in, 820 times over. These posterior simulations may also be used to examine the adapted estimates $\mathbb{E}[\Sigma_t|D_t]$. We implement the Gibbs samplers in C++. It takes on the order of a minute to produce 1500 samples for a single t and thus takes several days to produce forecasts for all of the days and models we consider. [2]

3.4 Model Comparison

Table 3.2 shows that the forecasts produced by exponentially smoothing realized kernels are superior to the forecasts produced by FSVol and the extensions to FSVol described in §3.2.2 and §3.2.3. This bolsters the result found in Liu [2009] that exponentially smoothing realized kernels is superior to more complicated models that only make use of daily data. However, as seen in Table 3.2, the portfolio constructed using FSVol averages only 0.0633% more standard deviation per day than the portfolio constructed using realized kernels. Thus, for every $10,000 invested daily, one takes on $6.33 in extra risk by using the factor stochastic volatility model instead of the realized kernel forecasting procedure when predicting the one-step ahead covariance matrix.

Among the extensions to factor stochastic volatility, only the dynamic factor loadings model that uses information from the Cholesky decomposition of RK_t outperforms factor stochastic volatility. In that model, we take $\tilde{X}_t$ from §3.2.3 to be the first p columns from L_t where $L_t D_t L_t' = RK_t$ is the Cholesky decomposition p is the

[2] One pays a price in the time taken to implement complied code. We have written over 20,000 lines of code to implement, test, and coordinate the C++, R, Bash, and Perl routines needed to run the MCMC simulations.

number of factors.

Table 3.2 also includes the performance of each method when using the adapted estimates $\mathbb{E}[\Sigma_t|D_t]$ for each loss function. The labeling "Realized Kernel - Random Walk" means that both the time t estimate and the time $t+1$ forecast are equal to RK_t. It appears that RK_t is a good proxy for the daily covariance matrix as it has the best log-likelihood and a decent portfolio performance. Among the FSVol models, one finds that only the dynamic factor loadings models improve upon classic factor stochastic volatility.

In general, one benefits by working with the whole realized kernel; but, exponential smoothing, the method used here to produce forecasts of daily covariance using the whole realized kernel, is ad hoc and lacks a notion of likelihood. A equivalent forecast generated by a statistical model would be preferable since it would include an accompanying notion of uncertainty. Below, we tackle this problem, building upon the model of Uhlig [1997] to wrap exponential smoothing within a statistical model, but first take a closer look at the robustness of exponentially smoothing realized kernels.

3.5 Robustness of Exponential Smoothing

The empirical comparisons above engender two criticisms. First, the data set considered heretofore follows the share prices of 30 large companies, for which stocks are traded frequently. Stocks that trade frequently are called liquid while stocks that trade infrequently are called illiquid. The realized kernel estimator relies on synchronizing the price vector of a collection of assets using the asset that is slowest

Method	Factors	Adapted Estimate		1-Step Ahead Forecast	
		MVP	LLH	MVP	LLH
Factor Stochastic Volatility	1	0.008837	97382	0.010213	94910
	2	0.008504	97824	0.009923	95155
	3	0.008400	98187	0.010035	**95302**
Log Variances - Eigenvalues	1	0.009275	97361	0.010588	94948
	2	0.009262	97762	0.010637	95177
	3	0.009359	97998	0.010810	95280
Loadings - Cholesky	1	0.007567	97567	0.009968	94120
	2	**0.007490**	98280	**0.009783**	94606
	3	0.007729	**98443**	0.009931	94714
Log Variances - Factor "Decomp."	1	0.009572	97199	0.010964	94798
	2	0.009027	97633	0.010449	95050
	3	0.009116	98002	0.010766	95264
Loadings - Factor "Decomp"	1	0.008121	97787	0.010079	94778
	2	0.007795	97520	0.010343	94028
	3	0.007834	97449	0.010161	94592
Realized Kernel - Random Walk		**0.007836**	**101384**	0.010602	91966
Exponential Smoothing		0.008782	98096	**0.009290**	**96675**
Uhlig-like Model		**0.007836**	**101384**	**0.009615**	94047

Table 3.2: The covariance estimation and prediction benchmarks. For each $t = 101, \ldots, 920$, we calculate an adapted estimate $\mathrm{Var}[r_t|D_t]$ and a one-step ahead forecast $\mathrm{Var}[r_t|D_{t-1}]$ of the day t covariance matrix; D_t is the data up to time t. For the FSVol-like models we re-estimate all of the parameters, in addition to all of the hidden states, for each t. For exponential smoothing, we use the in-sample period of $t = 51, \ldots, 100$ to pick the smoothing parameter λ which we then took as fixed, i.e. as part of the data set D_t, for $t = 101, \ldots, 920$. The entry labeled "Realized Kernel - Random Walk" estimates the day t covariance matrix using the day t realized kernel and forecasts the day t covariance matrix using the day $t-1$ realized kernel. For exponential smoothing, the adapted estimate is the exponentially smoothed realized kernels, S_t, while the one-step ahead forecast is the day $t-1$ weighted average S_{t-1}. See equation (3.3). The column labeled MVP reports the empirical standard deviation of the minimum variance portfolios and the column labeled LLH reports the log-likelihood, each calculated with both the adapted estimates and 1-step ahead forecasts. The realized kernel provides the best adapted estimate while the smoothed realized kernel provides the best 1-step ahead forecast. The row labeled Uhlig-like model refers to the estimates and forecasts produced using the model described in §3.7 with parameters (k, n, λ) determined by constrained maximum likelihood, also described in §3.7. This table and the accompanying caption is taken from Windle and Carvalho [2012].

to update. Thus, if one of the stocks within a portfolio is traded infrequently, the quality of the realized kernel estimate will degrade. However, this illiquidity will not affect factor stochastic volatility so long as the illiquid stock is traded at least once per day. Consequently, it may be the case that exponentially smoothing realized kernels does not work as well as FSVol when one includes an infrequently traded asset. Second, exponential smoothing relies on a single parameter to generate forecasts of matrix-variate data; but it may be inappropriate to use a single parameter when the collection of assets is not similar or when one considers many assets.

To check how infrequently traded assets impact forecasts, we artificially induce illiquidity in the data from §3.3. Instead of using an entire day's worth in intraday price data to construct realized kernels, we use prices observed at 5, 10, 15, 30, and 60 minute intervals. As before, an in-sample period is used to select the smoothing parameter for each set of realized kernels and an out-of-sample period is used to measure its performance. The plot on the left of Figure 3.1 shows choice of smoothing parameter for each collection while the plot on the right shows the resulting empirical standard deviation of the minimum variance portfolios. Exponential smoothing still beats FSVol when the realized kernels are constructed using prices observed every 5, 10, or 15 minutes. At 30 and 60 minutes FSVol out performs exponential smoothing. Thus, factor stochastic volatility may beat exponentially smoothed realized kernels in extreme circumstances.

To check if exponential smoothing is too parsimonious, we expand the number of assets under consideration to 96 equities[3], selected, in part, so that the new data

[3]The ticker symbols for the assets used are AA, ABT, AFL, AIG, ALL, APA, APC, AXP, BA,

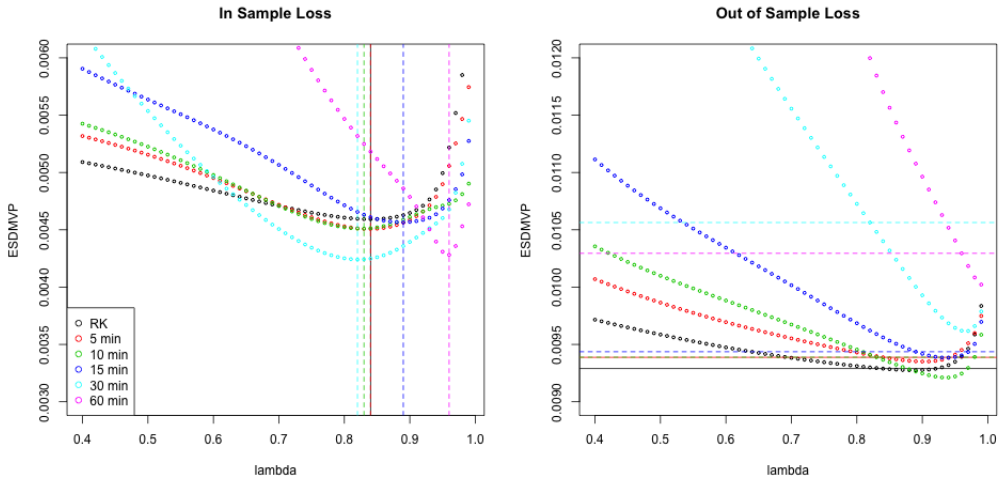


Figure 3.1: Illiquidity benchmarks. We construct the realized kernels for the 30 assets from §3.3 using prices sampled periodically every 5, 10, 15, 30, and 60 minutes. The vertical line in the left-hand plot is the in-sample choice of λ. The horizontal line in the right-hand plot is the ESDMVP for the portfolio for the out-of-sample period using the in-sample choice of λ. The full realized kernel performs best out-of-sample, though the 5 and 10 minute estimates are not far behind. The best FSVol like model had an ESDMVP of 0.00978, which is higher than the portfolios constructed using the 5, 10, or 15 minute realized covariances. Taken from Windle and Carvalho [2012].

covers the same time period as the original data set. For the entire collection of assets, a new set of realized kernels is constructed $RK_t, t = 1, \ldots, 920$. To study the effect of portfolio size and portfolio composition on the smoothing parameter and the resulting empirical measure of loss, we repeat the same procedure as described in §3.3. In particular, a portfolio of size N assets is selected at random; the best smoothing parameter from (3.3) is chosen by minimizing the loss over the in-sample period, which in this case is $t = 51, \ldots, 150$; and the out-of-sample loss produced via (3.3) is calculated. This procedure is repeated 200 times for each $N = 30, \ldots, 90$. The left plot of Figure 3.2 shows that the smoothing parameter is stable, while the right plot shows that the average empirical standard deviation *decreases* as the number of assets increases; thus, exponential smoothing is improves as the number of assets increases.

3.6 Exponential Smoothing and Volatility Models

The straight-forward nature of exponential smoothing compels us to explain that, in fact, similar forecasting procedures are found in popular univariate volatility models and consequently, devising a multidimensional model that reproduces exponential smoothing is a natural step forward, despite its apparent simplicity. Thus, to clarify the mechanics of volatility forecasting and to show that encapsulating expo-

BAC, BAX, BBT, BEN, BK, BMY, C, CAH, CAT, CCL, CL, COP, CVS, CVX, D, DD, DE, DHR, DIS, DOW, DUK, EMC, EMR, EXC, FDX, GD, GE, GIS, HAL, HD, HON, HPQ, IBM, ITW, JNJ, JPM, K, KMB, KO, LLY, LMT, LOW, MCD, MDT, MMM, MO, MRK, MRO, NEM, NKE, NOC, OXY, PEP, PFE, PG, PNC, PX, RTN, SLB, SO, STI, STT, SYY, T, TGT, TWX, TXN, UNH, UNP, USB, UTX, VZ, WAG, WFC, WMT, XOM, AAPL, AMAT, AMGN, COST, CSCO, DELL, INTC, MSFT, ORCL, QCOM, YHOO.

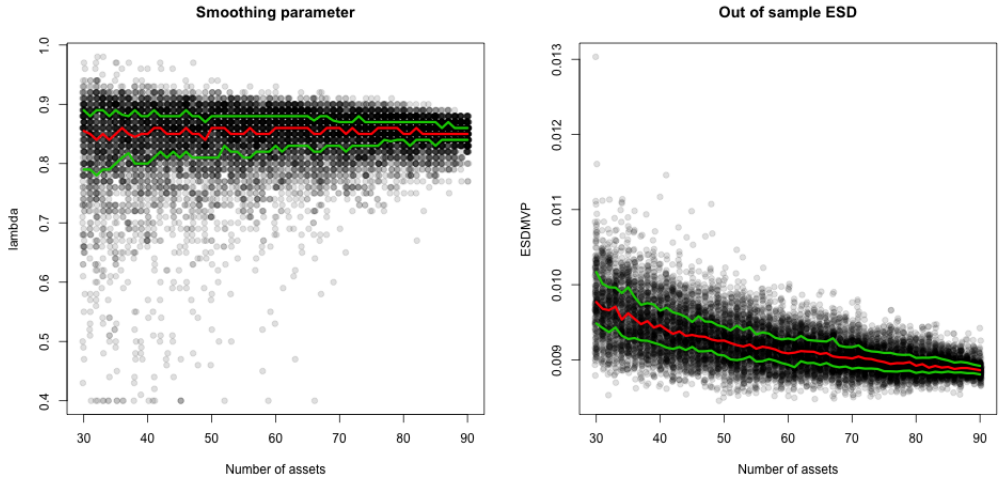


Figure 3.2: Portfolio composition. A plot of the smoothing parameter λ selected by minimizing in-sample loss of N stocks selected at random within a pool of 96 assets is on the left while the subsequent out-of-sample loss is on the right. In this case we take the in-sample period to be $t = 51, \ldots, 150$ and the out-of-sample period to be $t = 151, \ldots, 920$. The red line is the median ESDMVP for each N and the green lines are the first and third quartiles. Taken from Windle and Carvalho [2012].

nential smoothing in a model has precedent, we explain how the two most common models in univariate volatility forecasting, GARCH(1,1) and stochastic volatility, are exponential smoothing with mean reversion.

Briefly recapitulating exponential smoothing, consider the univariate time series $\{y_t\}_{t=1}^T$ and suppose that one has observed data $D_t = \{y_1, \ldots, y_t\}$. To forecast one step into the future using exponential smoothing, pick a smoothing parameter $\lambda \in (0, 1)$ and construct the exponentially weighted average

$$\hat{y}_{t+1} \propto \sum_{i=0}^{t-1} \lambda^k y_{t-i}.$$

As $\lambda \to 0$ the weighted average approaches a random walk forecast $\hat{y}_{t+1} = y_t$, which is the point-wise forecast produced when the dynamics of y_t follow $y_t = y_{t-1} + \omega_t,\ \omega_t \sim N(0, W)$. As $\lambda \to 1$ the weighted average approaches a white noise forecast $\hat{y}_{t+1} = \sum_{i=0}^{t-1} \frac{1}{t}\ y_{t-i}$, which is the point-wise forecast produced when the dynamics of y_t follow $y_t \sim N(\mu, W)$. In between these two extremes the one-step ahead forecasts balances the most recent observations with those further into the past. The generalized autoregressive conditional heteroskedasticity (GARCH) model [Bollerslev, 1986] and stochastic volatility (SV) model [Taylor, 1982] are extensions of this approach.

3.6.1 GARCH(1,1) and Stochastic Volatility Forecasts

Suppose that $(P_t)_{t=1}^T$ are a sequence of asset prices, such as the daily closing price of some equity, and that $r_t = \log P_t - \log P_{t-1}, t = 1, \ldots, T$ are the log-returns of those prices.

The GARCH(1,1) model specifies that

$$\begin{cases} r_t = \sigma_t \varepsilon_t, \quad \varepsilon_t \sim N(0,1), \\ \sigma_t^2 = \alpha + \beta \sigma_{t-1}^2 + \gamma r_{t-1}^2. \end{cases}$$

Expanding the recursion for σ_t^2 produces

$$\sigma_t^2 = \alpha \sum_{i=0}^{k} \beta^i + \gamma \sum_{i=0}^{k} \beta^i r_{t-i-1}^2 + \beta^{k+1} \sigma_{t-k-1}^2.$$

In the limit this becomes

$$\sigma_t^2 = (1-\gamma')\mu + \gamma' \sum_{i=0}^{\infty} w_i r_{t-i-1}^2 \tag{3.6}$$

where $\mu = \alpha/(1-\beta-\gamma)$, $w_i = (1-\beta)\beta^i$, $\alpha' = \alpha/(1-\beta)$, and $\gamma' = \gamma/(1-\beta)$, which shows that the forecast for σ_t^2 will average a long-term mean and the exponentially weighted moving average of past squared returns. Thus, instead of starting with the GARCH(1,1) *model*, one could start with the *forecast* (3.6) and then pick the parameters α, β, and γ to minimize the loss function $\sum_{t=1}^{T} N(r_t|0,\sigma_t^2)$, where σ_t^2 comes from (3.6), given the data $\{r_t\}_{t=\infty}^{T}$. This procedure produces identical point estimates to those one would get by picking α, β, and γ by maximum likelihood using the GARCH model. In this way, GARCH(1,1) is like an exponential smoothing forecasting procedure in which the measure of loss is a Gaussian likelihood.

Stochastic volatility has a similar interpretation, but instead of smoothing squared returns, one smooths the log of square returns, that is one takes the exponentially weighted geometric average of squared returns to produce a point forecast. Again, suppose that $\{r_t\}$ are the returns of some financial asset. Recall that Taylor's model [Taylor, 1982] is

$$\begin{cases} r_t \sim N(0, e^{h_t}), \\ h_t = \alpha + \phi h_{t-1} + \omega_t, \quad \omega_t \sim N(0, W) \end{cases}$$

where one now tracks the log variance h_t. As seen in §3.1.3, the transformation $y_t = \log(r_t^2)$ leads to

$$\begin{cases} y_t = h_t + \nu_t, \\ h_t = \alpha + \phi h_{t-1} + \omega_t, \quad \omega_t \sim N(0, W). \end{cases}$$

The distribution of ν_t should be $\log \chi_1^2$; however, let us approximate this using $\nu_t \sim N(0, V)$, in which case the model above becomes a dynamic linear model (DLM) and hence has closed form filtering and forecasting distributions [Harrison and West, 1997]. In particular, Given $h_{t-1} \sim N(m_{t-1}, C_{t-1})$ the DLM recursions are

	Mean	Variance
$h_t \mid D_{t-1}$	$a_t := \alpha + \phi m_{t-1}$	$R_t := \phi^2 C_{t-1} + W$
$y_t \mid D_{t-1}$	$f_t := a_t$	$Q_t := R_t + V$
$h_t \mid D_t$	$m_t := (I - A_t)(\alpha + \phi m_{t-1}) + A_t y_t$	$C_t = R_t - A_t R_t$

where $A_t = R_t Q_t^{-1}$. One can show that

$$a_t = \alpha + \phi(I - A_{t-1})a_{t-1} + \phi A_{t-1} y_{t-1},$$

and further that, given to noise ratio $r = W/V$,

$$\frac{1}{A_t} = 1 + \frac{1}{\phi^2 A_{t-1} + r}.$$

Expanding the recursion for a_t one finds that

$$a_t = \sum_{i=0}^{k-1} M_i^{t-1} \alpha + \sum_{i=0}^{k-1} M_i^{t-1} \phi A_{t-i-1} y_{t-i-1} + M_k^{t-1} a_{t-k}$$

where

$$M_i^t = \prod_{j=i}^{1} \phi(I - A_t) \text{ for } i > 0 \text{ and } M_0^t = 1.$$

As A_t converges to A, the limit in k is approximately

$$a_t = \alpha \sum_{i=0}^{\infty} \psi^i + \sum_{i=0}^{\infty} \psi^i (\phi A) y_{t-i-1}, \text{ where } \psi = \phi(1 - A).$$

We can rewrite this as

$$a_t = (1 - \gamma')\mu + \gamma' \sum_{i=0}^{\infty} w_i \log(r_{t-i-1}^2) \tag{3.7}$$

where $\mu = \alpha'/(1 - \gamma')$, $\alpha' = \alpha/(1 - \psi)$, $\gamma' = (\phi - \psi)/(1 - \psi)$, and $w_i = (1 - \psi)\psi^i$. Thus, the forecast for h_t is an average of a long-term mean and an exponentially weighted average of the past log squared returns and the point forecast $\exp(a_t)$ of the variance $\text{Var}(r_t|D_{t-1})$ is like a geometric exponentially weighted moving average of past log squared returns.

3.7 Model Based Exponential Smoothing

As seen in §3.4, exponentially smoothing realized kernels produces good forecasts, but exponential smoothing lacks the richness of a statistical model; for instance, the forecasts produced using exponential smoothing do not include a description of how forecast errors are distributed. We are interested in filling this gap for large, dynamic covariance matrices, just as GARCH(1,1) and SV fill this gap for univariate time series. The following discussion follows Windle and Carvalho [2012].

Exponential smoothing covariance matrices dates back to at least Quintana and West [1987], where it is called variance discounting. Subsequently, Shephard [1994] provided a model-based justification for univariate variance discounting while Uhlig [1997] provided a model-based justification for variance discounting when the

response is a vector. More recent proposals include the Wishart autoregressive process of Gourieroux et al. [2009], the multivariate stochastic volatility models of Philipov and Glickman [2006], the inverse Wishart autoregressive process of Fox and West [2011], and the HEAVY models of Noureldin et al. [2011]. Fox and West remark that one may employ the techniques of Pitt and Walker [2005] to construct matrix variate processes. One may indirectly model covariance matrices either through the Cholesky decomposition Chiriac and Voev [2010] or the matrix logarithm Bauer and Vorkink [2011]. The approach entertained here differs from the above models in its simplicity.

We draw inspiration from the state-space model of Uhlig [1997] in which the hidden states correspond to precision matrices and the observations correspond to vectors of asset returns. The filtered estimates and one-step ahead forecasts produced by this model are exponentially weighted averages of the the outer product of past returns. We aim to use the same machinery to produce filtered estimates and one-step ahead forecasts that are exponentially weighted averages of realized covariance matrices. Prado and West [2010] explore a similar option; though, their model constrains the smoothing parameter λ found in (3.3) so that $\lambda > (m-2)/(m-1)$ where m is the order of the matrix-variate data. When m is large this forces λ to be very close to 1, which may be unreasonable. Our model, in contrast, allows for more freedom in the smoothing parameter.

3.7.1 The Model

The following extension to the work of Uhlig [1997] handles relatively high-dimensional covariance matrices and wraps the exponential smoothing forecasting

procedure in a model. As noted above, Uhlig's original model encased exponential smoothing techniques [Quintana and West, 1987] in a statistical model. His model smooths rank-1 covariance matrices of order m via

$$\begin{cases} Y_t \sim W_m(1, \mathcal{P}_t^{-1}) \\ \mathcal{P}_t = U_{t-1}' \Psi_t U_{t-1}/\lambda, \quad \Psi_t \sim \beta_m\Big(\frac{\nu-1}{2}, 1/2, I\Big) \\ U_{t-1} = \text{chol } \mathcal{P}_{t-1}. \end{cases}$$

where $Y_t = r_t r_t'$ and r_t are a vector of asset returns. W_m is the Wishart distribution and β_m is the multivariate beta distribution, which we describe below. One may exponentially smooth symmetric, positive definite matrices of rank k, including full rank matrices, via

$$Y_t \sim W_m(k, (k\mathcal{P}_t)^{-1}) \tag{3.8}$$

$$\mathcal{P}_t = U_{t-1}' \Psi_t U_{t-1}/\lambda, \ \Psi_t \sim \beta_m\Big(\frac{n}{2}, \frac{k}{2}, I\Big), \tag{3.9}$$

$$U_{t-1} = \text{chol } \mathcal{P}_{t-1}$$

where $n > m-1$ and $k \in \big\{(m-1, \infty) \cup \{1, \ldots, m-1\}\big\}$. The evolution from the distribution of $(\mathcal{P}_{t-1}|D_{t-1})$ to the distribution of $(\mathcal{P}_t|D_{t-1})$ proceeds in closed form as justified by the following theorem, taken from Windle and Carvalho [2012], which combines theorems found in Uhlig [1997], Muirhead [1982], and Díaz-García [2003]. Regarding notation, we will write $S_{m,i}^+$ for the set of symmetric, non-negative definite matrices of order m and rank i with the convention that $S_{m,i}^+$ consists of those matrices with rank $\lceil i \rceil \wedge m$ when i is a real number. When considering full-rank matrices we just write S_m^+. We write $S_{m,i}^+(A)$ as the set of $X \in S_{m,i}^+$ such that $A - X \in S_m^+$.

Definition 3.1. *Let k be a positive integer less than m or a real number greater than*

$m-1$ *and* $n > m-1$. *The multivariate beta distribution* $\beta_m(k/2, n/2)$, *is*

$$U \sim \frac{\Gamma_m\left[\frac{1}{2}(k+n)\right]}{\Gamma_m(\frac{1}{2}k)\Gamma_m(\frac{1}{2}n)}(\det U)^{(k-m-1)/2}\left[\det(I-U)\right]^{(n-m-1)/2}dU$$

for $U \in S_m^+(I)$ *when* $k > m-1$ *and*

$$U \sim \frac{\Gamma_m\left[\frac{1}{2}(k+n)\right]}{\Gamma_m(\frac{1}{2}k)\Gamma_m(\frac{1}{2}n)}(\det D)^{(k-m-1)/2}\left[\det(I-U),\right]^{(n-m-1)/2}dU$$

where $U = HDH' \in S_{m,k}^+(I)$, H *is a matrix of orthonormal columns of order* $m \times k$, *and* D *is a diagonal matrix of order* $k \times k$, *when* k *is a positive integer less than* m. *When* k *is a positive integer less than* m *we define*

$$V \sim \beta_m(n/2, k/2)$$

by $V = I - U$ *and* $U \sim \beta_m(k/2, n/2)$. *We define*

$$V \sim \beta_m(k/2, n/2, S)$$

by $V = T'UT$ *where* $T'T = S$ *is the Cholesky factorization of* S *and* $U \sim \beta_m(k/2, n/2)$.

Theorem 3.2. *Let* k *be a positive integer less than* m *or a real number greater than* $m-1$ *and let* $n > m-1$. *The bijection from* $S_{m,k}^+ \times S_m^+$ *to* $S_m^+ \times S_{m,k}^+(I)$ *defined by* $(A, B) \mapsto (S, U)$ *via*

$$\begin{cases} S = A + B \\ U = (T^{-1})'AT^{-1} \end{cases}$$

where $T'T = S$ *and* T *is the upper triangular Cholesky factor of* S, *or inversely*

$$\begin{cases} A = T'UT, \\ B = T'(I-U)T, \end{cases}$$

defines a change of variables from

$$A \sim W_m(k, \Sigma) \perp B \sim W_m(n, \Sigma) \tag{3.10}$$

to

$$S \sim W_m(n+k, \Sigma) \perp U \sim \beta_m(k/2, n/2). \tag{3.11}$$

Further, the conditional distribution of $(S|B)$ *and* $(B|S)$ *are*

$$S \mid B = W_m(k, \Sigma) + B, \tag{3.12}$$

and

$$B \mid S = T'(I-U)T = \beta_m(n/2, k/2, S). \tag{3.13}$$

One may forward filter and backwards sample the joint distribution $(\mathcal{P}_{1:T}|D_T)$ [Windle and Carvalho, 2012], but we are interested in 1-step ahead forecasts and hence the distribution of $(Y_t|D_{t-1})$, which only requires understanding how $(\mathcal{P}_{t-1}|D_{t-1})$ and $(\mathcal{P}_t|D_{t-1})$ change with time. In particular, given the information set D_{t-1}, suppose that

$$\mathcal{P}_{t-1} \sim W_m(n+k, \Sigma_{t-1}^{-1}) \perp \Psi_t \sim \beta_m(n/2, k/2),$$

which is similar to equation (3.11). This implies that

$$Z_t \sim W_m(k, \Sigma_{t-1}^{-1}) \perp \lambda\mathcal{P}_t \sim W_m(n, \Sigma_{t-1}^{-1})$$

as seen in equation (3.10) and tells us how the distribution of $\mathcal{P}_{t-1}$ transitions to the distribution of $\mathcal{P}_t$. We may update the distribution $(\mathcal{P}_t|D_{t-1}) \sim W_m(n, \Sigma_{t-1}^{-1}/\lambda)$ with

the data Y_t:

$$\begin{aligned} p(\mathcal{P}_t|D_{t-1}, Y_t) &\propto p(Y_t|\mathcal{P}_t)p(\mathcal{P}_t|D_{t-1}) \\ &\propto |\mathcal{P}_t|^{(k-m-1)/2}|\mathcal{P}_t|^{(n-m-1)/2}\exp\Big(-\frac{1}{2}\text{ tr }\Big[Y_t k\mathcal{P}_t + \mathcal{P}_t\lambda\Sigma_{t-1}\Big]\Big) \\ &\propto |\mathcal{P}_t|^{(\nu-m-1)/2}\exp\Big(-\frac{1}{2}\text{ tr }\mathcal{P}_t\Big[kY_t + \lambda\Sigma_{t-1}\Big]\Big) \\ &\propto W_m(\nu, (\lambda\Sigma_{t-1} + kY_t)^{-1}),\ \nu = k + n. \end{aligned}$$

We repeat this process to generate the distributions for $(\mathcal{P}_t|D_{t-1})$ and $(\mathcal{P}_t|D_t)$ recursively.

The evolution may be summarized as in Windle and Carvalho [2012]. Suppose Σ_0 is initialized to some prior value. Proceeding inductively, one has:

- Time $t-1$ "posterior:"

$$\mathcal{P}_{t-1} \mid D_{t-1} \sim W_m(\nu, \Sigma_{t-1}^{-1}).$$

- Evolution, i.e. time t "prior:"

$$\mathcal{P}_t \mid D_{t-1} = \lambda^{-1}W_m(n, \Sigma_{t-1}^{-1}).$$

- Observation:

$$Y_t \mid \mathcal{P}_t \sim W_m(k, (k\mathcal{P}_t)^{-1}).$$

  This ensures that $\mathbb{E}[Y_t|\mathcal{P}_t] = \mathcal{P}_t^{-1}$.

- Update, i.e. time t "posterior:"

$$\mathcal{P}_t \mid D_t \sim W_m(\nu, (\lambda\Sigma_{t-1} + kY_t)^{-1}).$$

One only needs the recursion

$$\Sigma_t = \lambda\Sigma_{t-1} + kY_t$$

to keep track of the parameter Σ_t. The following moments are taken from Windle and Carvalho [2012] as well.

- "Posterior" mean of hidden variance:

$$\mathbb{E}[\mathcal{P}_{t-1}^{-1}|D_{t-1}] = \frac{\Sigma_{t-1}}{\nu - m - 1}.$$

- "Prior" mean of hidden variance:

$$\mathbb{E}[\mathcal{P}_{t}^{-1}|D_{t-1}] = \frac{\lambda\Sigma_{t-1}}{\nu - k - m - 1}.$$

- Forecasted mean of hidden variance:

$$\mathbb{E}[Y_t \mid D_{t-1}] = \mathbb{E}[\,\mathbb{E}[Y_t \mid \mathcal{P}_t, D_{t-1}] \mid D_{t-1}] = \mathbb{E}[\mathcal{P}_t^{-1} \mid D_{t-1}].$$

If we chose

$$\lambda = \frac{\nu - k - m - 1}{\nu - m - 1} \tag{3.14}$$

then the time t "prior" mean of the hidden variance is equal to the time $t-1$ "posterior" mean. In terms of λ and k the constraint is

$$\nu - m - 1 = k(1-\lambda)^{-1}.$$

In the limit, i.e. for sufficiently large N,

$$\Sigma_t \simeq k\sum_{i=0}^{N} \lambda^i Y_{t-i},$$

and thus,

$$\mathbb{E}[Y_t|D_{t-1}] \simeq (1-\lambda)\sum_{i=0}^{N}\lambda^i Y_{t-1-i}$$

In other words, the 1-step ahead point forecasts $\mathbb{E}[Y_t|D_{t-1}]$ are an exponentially weighted average of past observations $\{Y_s\}_{s=1}^{t-1}$, just like exponential smoothing.

Degenerate Evolution

One drawback to this model is that the evolution of $\mathcal{P}_t$ degenerates to a non-recognizable distribution when one does not update the information set with new observations. In particular, suppose that one has the information set D_{t-1} and consider evolving $(\mathcal{P}_t|D_{t-1})$ to $(\mathcal{P}_{t+1}|D_{t-1})$. Using Theorem 3.2, one would go from

$$\mathcal{P}_t|D_{t-1} \sim W_m(n, \Sigma_{t-1}^{-1}/\lambda) \;\perp\; \Psi_t \sim \beta_m(n/2, k/2).$$

to

$$\mathcal{P}_{t+1} \sim W_m(n-k, \Sigma_{t-1}^{-1}/\lambda^2).$$

Thus as we evolve $\mathcal{P}_t$ *without updating* the information set D_{t-1} the distribution loses k degrees of freedom. Iterating many steps into the future, $(\mathcal{P}_{t+j}|D_{t-1})$ eventually becomes an unrecognizable distribution.

3.7.2 Estimating n, k, and λ

Windle and Carvalho [2012] show that one may take a joint draw from the complete conditional $(\{\mathcal{P}_s\}_{s=0}^{T}|D_T, n, k, \lambda)$. However, the complete conditionals for n, k, and λ do not have convenient forms and hence one must appeal to a Metropolis-Hastings step or some other method for picking these parameters. To that end, it

is more convenient to consider the marginal posterior $(n, k, \lambda | D_T)$. In Proposition 3.3 below (taken from Windle and Carvalho [2012]) we show that one may marginalize the hidden states $\{\mathcal{P}_t\}$ to produce the conditional distributions $(Y_t | D_{t-1}, n, k, \lambda)$. These distributions can be used to factor the joint distribution $(Y_{1:T} | D_0)$ where $D_0 = \{\Sigma_0, n, k, \lambda\}$ as

$$p(Y_{1:T} | D_0) = \Big[\prod_{t=1}^{T} p(Y_t | D_{t-1}) \Big]. \tag{3.15}$$

One may use this factorization to pick (n, k, λ) by maximum likelihood or to calculate the log-posterior of $(n, k | D_T)$ for Metropolis-Hastings sampling. Its distribution is related to the multivariate beta distributions of Olkin and Rubin [1964].

Proposition 3.3. *Suppose that* $k, n > m$, $(Y_t | \mathcal{P}_t) \sim W_m(k, (k\mathcal{P}_t)^{-1})$, *and* $(\mathcal{P}_t | D_{t-1}) \sim W_m(n, V_t^{-1})$. *Then the density for* $(Y_t | D_{t-1})$ *is*

$$\beta'_m(k/2, n/2, V_t/k) := \frac{\Gamma_m(\frac{\nu}{2})}{\Gamma_m(\frac{n}{2})\Gamma_m(\frac{k}{2})} \frac{|Y_t|^{(k-m-1)/2} |V_t/k|^{n/2}}{|V_t/k + Y_t|^{\nu/2}}.$$

Proof. To see this consider the joint density $p(Y_t | \mathcal{P}_t) p(\mathcal{P}_t | D_{t-1})$. It is

$$\frac{|k\mathcal{P}_t|^{k/2}}{2^{km/2}\Gamma_m(\frac{k}{2})} |Y_t|^{(k-m-1)/2} \exp\Big\{ \frac{-1}{2} \text{ tr } k\mathcal{P}_t Y_t \Big\}$$
$$\frac{|V_t|^{n/2}}{2^{nm/2}\Gamma_m(\frac{n}{2})} |\mathcal{P}_t|^{(n-m-1)/2} \exp\Big\{ \frac{-1}{2} \text{ tr } V_t \mathcal{P}_t \Big\}$$

which is

$$\frac{|Y_t|^{(k-m-1)/2}}{2^{km/2}\Gamma_m(\frac{k}{2})} \frac{|V_t|^{n/2}}{2^{nm/2}\Gamma_m(\frac{n}{2})} \, k^{km/2} |\mathcal{P}_t|^{(\nu-m-1)/2} \exp\Big\{ \frac{-1}{2} \text{ tr } \big(V_t + kY_t\big)\mathcal{P}_t \Big\}.$$

The latter terms are the kernel for a Wishart distribution. Integrating the kernel for $\mathcal{P}_t$ produces

$$\frac{2^{\nu m/2}\Gamma_m(\frac{\nu}{2})}{|V_t + kY_t|^{\nu/2}}.$$

Hence the distribution of $Y_t|D_{t-1}$ is

$$\frac{\Gamma_m(\frac{\nu}{2})k^{km/2}}{\Gamma_m(\frac{n}{2})\Gamma_m(\frac{k}{2})}\frac{|Y_t|^{(k-m-1)/2}|V_t|^{n/2}}{|V_t+kY_t|^{\nu/2}}.$$

Factoring the k in the denominator gives us

$$\frac{\Gamma_m(\frac{\nu}{2})}{\Gamma_m(\frac{n}{2})\Gamma_m(\frac{k}{2})}\frac{|Y_t|^{(k-m-1)/2}|V_t/k|^{n/2}}{|V_t/k+Y_t|^{\nu/2}}.$$

□

Proposition 3.3 may be used to select the parameters n, k, and λ. Recall that the data set described in §3.3 consists of realized kernels RK_t constructed for $t = 1, \ldots, 920$ days and that this data set has been partitioned into a initialization set $RK_t, t = 1, \ldots, 50$, a in-sample set $RK_t, t = 51, \ldots, 100$, and an out-of-sample set $RK_t, t = 101, \ldots, 920$. To select n, k, and λ by maximum likelihood, we first set Σ_{50} by exponentially smoothing RK_t for $t = 1, \ldots, 50$ with smoothing parameter 0.9, a common default value, to generate S_{50} as in (3.3) and then letting $\Sigma_{50} = kS_{50}$. Given Σ_{50} and the constraint (3.14) one can calculate the likelihood for the in-sample set, $\ell(n, k|Y_{51:100}, \Sigma_{50})$, using Proposition 3.3. Figure 3.3 shows the level sets for this likelihood. The maximum is at $(n, k) = (43.61, 70.66)$, implying a smoothing parameter $\lambda = 0.476$, which somewhat smaller than one would expect. To compare this model-based method with the previous approaches, we fix λ, n, and k from the maximum likelihood estimate and compute the empirical loss for the out-of-sample set using (3.3). As seen Table 3.2 these forecasts out-perform the FSVol-like models; however, the matrix-variate model-based approach under-performs the forecasts generated by picking the exponential smoothing parameter by minimizing the primary measure of loss.

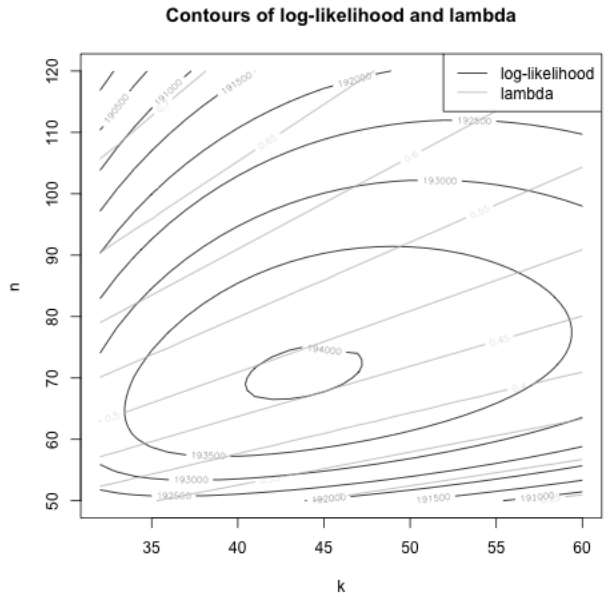


Figure 3.3: The matrix-variate, state-space model's marginal log-likelihood. The log-likelihood of $(n, k|\{RK_t\}_{t=51}^{100}, \Sigma_{50})$ where Σ_{50} has been initialized by exponential smoothing and λ is fixed by constraint (3.3). Proposition 3.3 shows how to compute the log-likelihood. Taken from Windle and Carvalho [2012].

3.7.3 Connection to IGARCH

Returning to the discussion from §3.6, one may reconcile the Uhlig-like, matrix-variate state-space model with an integrated GARCH model. In particular, consider the IGARCH model for daily returns:

$$\begin{cases} r_t \sim N(0, \sigma_t^2) \\ \sigma_t^2 = \lambda \sigma_{t-1}^2 + (1-\lambda) r_t^2. \end{cases}$$

Like §3.6, σ_t^2 is the rolling, exponentially weighted average of past square returns. The factorization $p(r_{1:T}|D_0) = \prod_{t=1}^{T} p(r_t|D_{t-1})$ lets one calculate the likelihood of λ. Similarly, after marginalizing $\mathcal{P}_t$ in the Uhlig-like model, an integrated GARCH representation emerges:

$$\begin{cases} Y_t \sim \beta'_m(\frac{k}{2}, \frac{n}{2}, V_t^*) \\ V_t^* = \lambda(V_{t-1}^* + Y_{t-1}), \end{cases}$$

so that

$$V_t^* = \sum_{i=1}^{N} \lambda^i Y_{t-1}$$

is a rolling, exponentially weighted sum of past realized kernels. The same factorization $p(Y_{1:T}|D_0) = \prod_{t=1}^{T} p(Y_t|D_{t-1})$ with constraint (3.14) lets one estimate (k, n) and hence λ. Thus, both models generate exponentially weighted sums whose smoothing parameters can be selected via a likelihood of the form

$$\sum_{t=1}^{T} \log p(Y_t|D_{t-1}, \lambda).$$

3.8 Recapitulation

Koopman et al. [2005] and Liu [2009] show that high-frequency statistics are a useful source of data for forecasting variation and covariation of financial asset

returns. These observations prompted us to examine (1) whether factor stochastic volatility is inferior to simple forecasting procedures that use high-frequency statistics and (2) whether one can incorporate information from high-frequency statistics into factor-like models. To both inquiries the result is affirmative. As seen in §3.4, exponential smoothing realized kernels produces better forecasts than factor stochastic volatility models, which only make use of daily returns. Given that observation, we embarked on the task of improving factor stochastic volatility models using high-frequency data. We found that one can do this by altering FSVol so that the factor loadings change in time and track information from the realized kernel. Such a model preserves the factor interpretation of asset returns, a desirable feature of any financial model. However, the dynamic factor loadings model still performs poorly compared to smoothing realized kernels.

All of these conclusions are drawn using a financially meaningful measure of loss, the empirical standard deviation of the one-step ahead minimum variance portfolios. One of the main applications of forecasting covariance matrices is the construction of optimal portfolios for an investor with certain, predefined attitudes towards risk and return. Our primary measure of loss fits within this framework, but does not depend on the mean of the assets' returns, a desirable property as estimating the mean is difficult. As a tool of portfolio construction, we have shown that exponentially smoothing realized kernels is better than factor stochastic volatility. Though, it is worth noting that the magnitude by which factor stochastic volatility loses is not too great, an observation relevant for users of these models, whose resources will vary. An investor may study our results and find that the extra cost of acquiring the

high-frequency data to construct realized kernels outweighs any advantage gained in improvements in prediction of covariance matrices. However, if one does have access to this data we have shown that exponentially smoothing realized kernels produces better portfolios. Further, we have shown that the performance of these portfolios is robust to to the inclusion of infrequently traded assets and to the specific composition and size of the collection of assets considered, important points for volatility forecasting in practice.

Despite these successes, exponential smoothing is an ad hoc approach, and it is be preferable to encase the procedure in a statistical model. To that end, we have built a matrix-variate state-space model, in the spirit of Uhlig [1997], that has closed form evolution and filtering equations, essential attributes when working with high-dimensional objects. Further, the hidden states of the model can be marginalized to estimate the two remaining parameters.

Appendix

Appendix 1

Pseudocode

Algorithm 1 $IG(\mu, \lambda)$ Sampler [Devroye, 1986].

$Y \sim N(0,1)$.
$W \leftarrow \mu(1 + \frac{\mu Y}{2\lambda})$.
$X \leftarrow W - \sqrt{W^2 - \mu^2}$.
$U \sim \mathcal{U}(0,1)$.
if $U > \mu/(\mu + X)$ **then**
    $X \leftarrow \mu^2/X$.
end if
Return X.

Algorithm 2 $\text{IGa}(1/2, \text{scale} = s)\mathbf{1}_{(0,t)}$ Sampler [Devroye, 1986].

$R \leftarrow t/s$.
$E \sim \mathcal{N}(0,1)\mathbf{1}_{(1/\sqrt{R},\infty)}$.
$X = s/E^2$.
Return X.

Algorithm 3 $IG(\mu, \lambda)\mathbf{1}_{(0,t)}$ Sampler.

$z \leftarrow \frac{1}{\mu}$. If you pass z directly then $z = 0$ is a $\text{IGa}(1/2, \lambda)\mathbf{1}_{(0,t)}$ draw.
repeat
    $X \sim \text{IGa}(1/2, \lambda/2)\mathbf{1}_{(0,t)}$.
    $\alpha \leftarrow \exp(\frac{-\lambda z^2}{2} X)$
    $U \sim \mathcal{U}$
until $U \leq \alpha$

Algorithm 4 Hybrid $IG(\mu, \lambda)\mathbf{1}_{(0,t)}$ Sampler.

Let τ be some pre-specified constant, e.g. $\tau = \mu$.
$X \leftarrow X + 1$.
if $t < \tau$ **then**
  $X \sim \text{IG}(\mu, \lambda)\mathbf{1}_{(0,t)}$ from Algorithm 3.
else
  repeat
    $X \sim \text{IG}(\mu, \lambda)$
  until $X < t$
end if

Algorithm 5 $\text{Ga}(\text{shape}, \text{rate})\mathbf{1}_{(t,\infty)}$ Sampler [Philippe, 1997, Dagpunar, 1978].

$a \leftarrow$ shape; $b \leftarrow t$ rate.
$c_0 \leftarrow 0.5((b-a) + \sqrt{(b-a)^2 + 4b})/b$.
repeat
  $X \sim b + \mathcal{E}(c_0)$
  $\log \rho = (a-1)\log(X) - X(1-c_0)$
  $\log M = (a-1)\log((a-1)/(1-c_0)) - (a-1)$
until $\log \mathcal{U}(0,1) \leq \log \rho - \log M$.
Return tX/b.

Algorithm 6 $J^*(1, z)$ Sampler.

Input: z, a positive real number
Define: $\texttt{pigauss}(t \mid \mu, \lambda)$, the CDF of the inverse Gaussian distribution
Define: $a_n(x)$, the piecewise-defined coefficients in (1) and (2).
$t \leftarrow 0.64$, $K \leftarrow \pi^2/8 + z^2/2$
$p \leftarrow \frac{\pi}{2K} \exp(-Kt)$
$q \leftarrow 2 \exp(-|z|)\, \texttt{pigauss}(t \mid \mu = 1/z, \lambda = 1.0)$
repeat
  Generate $U, V \sim \mathcal{U}(0, 1)$
  if $U < p/(p + q)$ **then**
    (Truncated Exponential)
    $X \leftarrow t + E/K$ where $E \sim \mathcal{E}(1)$
  else
    (Truncated Inverse Gaussian)
    $\mu \leftarrow 1/z$
    if $\mu > t$ **then**
      repeat
        Generate $1/X \sim \chi^2_1 \mathbf{1}_{(t,\infty)}$
      until $\mathcal{U}(0, 1) < \exp(-\frac{z^2}{2} X)$
    else
      repeat
        Generate $X \sim IG(\mu, 1.0)$
      until $X < t$
    end if
  end if
  $S \leftarrow a_0(X)$, $Y \leftarrow VS$, $n \leftarrow 0$
  repeat
    $n \leftarrow n + 1$
    if n is odd **then**
      $S \leftarrow S - a_n(X)$; **if** $Y < S$, **then return** X
    else
      $S \leftarrow S + a_n(X)$; **if** $Y > S$, **then break**
    end if
  until FALSE
until FALSE

Algorithm 7 $J^*(h)$ Sampler from §2.7.

Let a_n be the coefficients from the inverse Gamma representation.
$p \leftarrow p_r(t)/(p_\ell(t) + p_r(t))$.
if $\texttt{runif}(1) < p$ **then**
    $X \leftarrow \text{Ga}(h, \text{rate} = \pi^2/8)\mathbb{I}_{(t,\infty)}$.
else
    $X \leftarrow \text{IGa}(1/2, h^2/2)\mathbb{I}_{(0,t)}$.
end if
$S \leftarrow S_0(X)$.
$Y \leftarrow \texttt{runif}(0, \tilde{g}(X))$.
$decreasing, done \leftarrow \texttt{FALSE}$.
$prev \leftarrow S$.
$n \leftarrow 0$
while !$done$ **do**
    $a.n \leftarrow a_n(X)$.
    $decreasing \leftarrow a.n < prev$.
    if n is odd **then**
        $S \leftarrow S - a_n(X)$.
        $done \leftarrow (Y \leq S)$ and $decreasing$
    else
        $S \leftarrow S + a_n(X)$.
        $done \leftarrow (Y > S)$ and $decreasing$
    end if
end while
Accept if $Y \leq S$.

Algorithm 8 $J^*(h, z)$ Sampler from §2.7.

Let a_n be the coefficients from the inverse **gamma** representation.
Everything here is conditional upon z and h.
$p \leftarrow p_r(t, z, h)/(p_\ell(t, z, h) + p_r(t, z, h))$.
$\lambda_z = \pi^2/8 + z^2/2$.
if $\texttt{runif}(1) < p$ **then**
    $X \leftarrow \text{Ga}(h, \text{rate} = \lambda_z)\mathbb{I}_{(t,\infty)}$.
else
    $X \leftarrow \text{IGauss}(\mu = h/z, h^2)\mathbb{I}_{(0,t)}$.
end if
$S \leftarrow S_0(X)$.
$Y \leftarrow \texttt{runif}(0, \tilde{g}(X))$.
$decreasing, done \leftarrow \texttt{FALSE}$.
$prev \leftarrow S$.
$n \leftarrow 0$
while !$done$ **do**
    $a.n \leftarrow a_n(X)$.
    $decreasing \leftarrow a.n < prev$.
    if n is odd **then**
        $S \leftarrow S - a_n(X)$.
        $done \leftarrow (Y \leq S)$ and $decreasing$
    else
        $S \leftarrow S + a_n(X)$.
        $done \leftarrow (Y > S)$ and $decreasing$
    end if
end while
Accept if $Y \leq S$.

Algorithm 9 $J^*(h,z)$ Saddlepoint Sampler from §2.8.

Setup:
Pick x_ℓ, x_c, and x_r.
Let κ_ℓ, κ_r, ρ_ℓ, ρ_r, and $k(x|h,z)$ be as in Proposition 2.25.
Let $sp_h(x|z)$ be the saddle point approximation.
Accept/Reject:
$w_\ell \leftarrow \kappa_\ell \Phi_{IG}(x_c, \mu = 1/\rho_\ell, \lambda = h)$.
$w_r \leftarrow \kappa_r(1 - \Phi_{Ga}(x_c, \text{shape} = h, \text{rate} = h\rho_r))$.
$w \leftarrow w_\ell + w_r$.
$go \leftarrow 1$.
while go **do**
  if $\mathcal{U}(0,w) \leq w_\ell$ **then**
    $X \leftarrow \text{IG}(\mu = 1/\sqrt{\rho_\ell}, \lambda = h)\mathbf{1}_{(0,x_c)}$.
    $K \leftarrow k(X|h,z)$. ▷ $k(x|h,z)$ is defined piecewise.
  else
    $X \leftarrow \text{Ga}(\text{shape} = h, \text{rate} = h\rho_r)\mathbf{1}_{(x_c,\infty)}$
    $K \leftarrow k(X|h,z)$.
  end if
  $S \leftarrow sp_h(X|z)$
  $go \leftarrow \mathcal{U}(0,K) > S$.
end while

Appendix 2

Truncated Inverse Gaussian Acceptance Rate

We employ two methods to simulate a truncated inverse Gaussian random variate, $Y \sim \text{IG}(\mu, \lambda)\mathbf{1}_{(0,t)}$. One approach is to use rejection sampling. In that case, the probability of accepting a proposal $X \sim \text{IG}(\mu, \lambda)$ is

$$\Phi_{IG}(t|h/z, h^2).$$

Another approach is to use accept/reject sampling with an $X \sim \text{IG}(1/2, h^2/2)$ proposal. The inverse Gaussian kernel is

$$k_f(x) = e^{-zh}x^{-3/2}\exp\Big(-\frac{z^2}{2x}(x-h/z)^2\Big) = x^{-3/2}\exp\Big(-\frac{xz^2}{2}-\frac{h^2}{2x}\Big).$$

while the kernel of the proposal is

$$k_g(x) = x^{-3/2}\exp\Big(-\frac{h^2}{2x}\Big).$$

The ratio of the target to the proposal is

$$\frac{k_f(x)}{k_g(x)} = e^{-xz^2/2},$$

which we may maximize over $x \in (0,t)$ to see that $c = 1$ ensures that

$$k_f(x) \leq ck_g(x)$$

on $(0, t)$ and so the probability of accepting a proposal is

$$\mathbb{E}_g[e^{-xz^2/2}] = \frac{e^{-zh}}{\Phi_{\text{IGa}}(t; 1/2, h^2/2)} \Phi_{IG}(t, h/z, h^2)$$

Thus, it is better to use the accept/reject approach if

$$e^{-zh} \geq 1 - \Phi_{\text{Ga}}(1/t, 1/2, \text{rate} = h^2/2).$$

Appendix 3

Binary Logistic Regression and Mixed Model Benchmarks

Descriptions of the data sets and methods used for the binary logistic regression and binary logistic mixed model benchmarks can be found in the technical supplement to Polson et al. [2013b]. The descriptions are reproduced here for convenience.

3.1 Data Sets

Nodal: part of the `boot` R package [Canty and Ripley, 2012]. The response indicates if cancer has spread from the prostate to surrounding lymph nodes. There are 53 observations and 5 binary predictors.

Pima Indian: There are 768 observations and 8 continuous predictors. It is noted on the UCI website that there are many predictor values coded as 0, though the physical measurement should be non-zero [Machine Learning Repository, 2012d]. We have removed all of those entries to generate a data set with 392 observations. The marginal mean incidence of diabetes is roughly 0.33 before and after removing the data.

Heart: The response represents either an absence or presence of heart disease. There are 270 observations and 13 attributes of which 6 are categorical or binary and

1 is ordinal. The ordinal covariate has been stratified by dummy variables [Machine Learning Repository, 2012c].

Australian Credit: The response represents either accepting or rejecting a credit card application [Machine Learning Repository, 2012a]. The meaning of each predictor has been removed to protect the propriety of the original data. There are 690 observations and 14 attributes, of which 8 are categorical or binary. There were 37 observations with missing attribute values. These missing values have been replaced by the mode of the attribute in the case of categorical data and the mean of the attribute for continuous data. This data set is linearly separable and results in some divergent regression coefficients, which are kept in check by the prior.

German Credit: The response represents either a good or bad credit risk [Machine Learning Repository, 2012b]. There are 1000 observations and 20 attributes including both continuous and categorical data. We benchmark two scenarios. In the first, the ordinal covariates have been given integer values and have not been stratified by dummy variable, yielding a total of 24 numeric predictors. In the second, the ordinal data has been stratified by dummy variables, yielding a total of 48 predictors.

3.2 Methods

All of these routines are implemented in R, though some of them make calls to C. In particular, the independence Metropolis samplers do not make use of any

non-standard calls to C, though their implementations have very little R overhead in terms of function calls; the Pólya-Gamma method calls a C routine to sample the Pólya-Gamma random variates, but otherwise only uses R. We think this is fair since other basic random variate generators in R call compiled code. As a check upon our independence Metropolis sampler we include the independence Metropolis sampler of Rossi et al. [2005], which may be found in the `bayesm` package [Rossi, 2012], though their sampler uses a t_6 proposal while ours uses a normal proposal. The suite of routines in the `binomlogit` package [Fussl, 2012] implement the techniques discussed in Fussl et al. [2011]. One routine provided by the `binomlogit` package coincides with the technique described in Frühwirth-Schnatter and Frühwirth [2010] for the case of binary logistic regression. A separate routine implements the latter and uses a single call to C. Gramacy and Polson's R package, `reglogit`, also calls external C code [Gramacy, 2012]. For every data set the regression coefficient was given a diffuse $N(0, 0.01I)$ prior, except when using Gramacy and Polson's method, in which case it was given a $\exp(\sum_i |\beta_i/100|)$ prior per the specifications of the `reglogit` package. The following is a short description of each method along with its abbreviated name.

PG: The Pólya-Gamma technique.

FS: The method of Frühwirth-Schnatter and Frühwirth [2010].

IMN: Independence Metropolis with a normal proposal. We calculate the posterior mode and the Hessian at the mode to pick the mean and variance of the proposal.

IMT: Independence Metropolis with a t_6 proposal from the R package `bayesm` [Rossi, 2012]. One calculates the posterior mode and the Hessian at the mode to pick the mean and scale matrix of the proposal.

OD: The method of O'Brien and Dunson [2004]. Strictly speaking, this is not logistic regression; it is binary regression using a student's T cumulative distribution function as the inverse link function. Their approach reduces to the t-link approach of Albert and Chib [1993] when not correcting by Metropolis-Hastings. To speed up the sampling we do not use the correction step.

Fussl: Work by Fussl et al. [2013] that extends the technique of Frühwirth-Schnatter and Frühwirth [2010]. A convenient representation is found that relies on a discrete mixture of normals approximation. From the R package `binomlogit` [Fussl, 2012].

MHGB: Similar to the discrete mixture of normals approach in Fussl et al. [2013], but instead of using a discrete mixture of normals, use a single normal to approximate the error term and correct using Metropolis-Hastings. MHGB stands for Metropolis-Hastings within Gibbs for binomial logistic regression. From the R package `binomlogit`.

MHG1: This routine is identical to MHGB, but is restricted to binary logistic regression. From the R package `binomlogit`.

MHGH: Like the discrete mixture of normals approach found in Fussl et al. [2013], but the specific sampling procedure is determined by the ratio y_i/n_i. From the R package `binomlogit`.

GP: The method of Gramacy and Polson [2012]. They employ another data augmentation scheme that uses only a single layer of latents. This routine uses a double exponential prior. The scale of this prior is set to agree with the scale of the normal prior used in all other cases above. From the R package `reglogit` [Gramacy, 2012].

3.3 Binary Logistic Regression Benchmarks

The following eight tables comprise the binary logistic benchmarks reported in §2.5. "ARate" refers the Metropolis-Hastings acceptance rate. When a sampler does not use a Metropolis-Hastings step, ARate is set to 1.0. The tables report the time taken to generate 10,000 samples along with the the minimum, median, and maximum effective sample sizes and effective sampling rates of those 10,000 samples. These numbers have been averaged over 10 batches for each data set.

Nodal data: $N = 53$, $P = 6$								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	2.98	1.00	3221.12	4859.89	5571.76	1081.55	1631.96	1871.00
IMN	1.76	0.66	1070.23	1401.89	1799.02	610.19	794.93	1024.56
IMT	1.29	0.64	3127.79	3609.31	3993.75	2422.49	2794.69	3090.05
OD	3.95	1.00	975.36	1644.66	1868.93	246.58	415.80	472.48
FS	3.49	1.00	979.56	1575.06	1902.24	280.38	450.67	544.38
Fussl	2.69	1.00	1015.18	1613.45	1912.78	376.98	598.94	710.30
MHGB	1.41	0.62	693.34	1058.95	1330.14	492.45	751.28	943.66
MHG1	1.30	0.61	671.76	1148.61	1339.58	518.79	886.78	1034.49
MHGH	3.06	1.00	968.41	1563.88	1903.00	316.82	511.63	622.75
GP	17.86	1.00	2821.49	4419.37	5395.29	157.93	247.38	302.00

Diabetes: $N = 392$, $P = 9$								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	5.65	1.00	3255.25	5444.79	6437.16	576.14	963.65	1139.24
IMN	2.21	0.81	3890.09	5245.16	5672.83	1759.54	2371.27	2562.59
IMT	1.93	0.68	4751.95	4881.63	5072.02	2456.33	2523.85	2621.98
OD	6.63	1.00	1188.00	2070.56	2541.70	179.27	312.39	383.49
FS	6.61	1.00	1087.40	1969.22	2428.81	164.39	297.72	367.18
Fussl	6.05	1.00	1158.42	1998.06	2445.66	191.52	330.39	404.34
MHGB	3.82	0.49	647.20	1138.03	1338.73	169.41	297.98	350.43
MHG1	2.91	0.48	614.57	1111.60	1281.51	211.33	382.23	440.63
MHGH	6.98	1.00	1101.71	1953.60	2366.54	157.89	280.01	339.18
GP	88.11	1.00	2926.17	5075.60	5847.59	33.21	57.61	66.37

Heart: $N = 270$, $P = 19$								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	5.56	1.00	2097.03	3526.82	4852.37	377.08	633.92	872.30
IMN	2.24	0.39	589.64	744.86	920.85	263.63	333.19	413.03
IMT	1.98	0.30	862.60	1076.04	1275.22	436.51	543.95	645.13
OD	6.68	1.00	620.90	1094.27	1596.40	93.03	163.91	239.12
FS	6.50	1.00	558.95	1112.53	1573.88	85.92	171.04	241.96
Fussl	5.97	1.00	604.60	1118.89	1523.84	101.33	187.49	255.38
MHGB	3.51	0.34	256.85	445.87	653.13	73.24	127.28	186.38
MHG1	2.88	0.35	290.41	467.93	607.80	100.70	162.25	210.79
MHGH	7.06	1.00	592.63	1133.59	1518.72	83.99	160.72	215.25
GP	65.53	1.00	1398.43	2807.09	4287.55	21.34	42.84	65.43

Australian Credit: $N = 690$, $P = 35$								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	12.78	1.00	409.98	3841.02	5235.53	32.07	300.44	409.48
IMN	3.42	0.22	211.48	414.87	480.02	61.89	121.53	140.59
IMT	3.92	0.00	8.27	10.08	26.95	2.11	2.57	6.87
OD	14.59	1.00	28.59	988.30	1784.77	1.96	67.73	122.33
FS	15.05	1.00	36.22	1043.69	1768.47	2.41	69.37	117.53
Fussl	14.92	1.00	29.34	991.32	1764.40	1.97	66.44	118.27
MHGB	8.93	0.19	13.03	222.92	435.42	1.46	24.97	48.76
MHG1	7.38	0.19	13.61	220.02	448.76	1.85	29.83	60.84
MHGH	18.64	1.00	28.75	1040.74	1817.85	1.54	55.84	97.53
GP	162.73	1.00	95.81	2632.74	4757.04	0.59	16.18	29.23

German Credit: $N = 1000$, $P = 25$								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	15.37	1.00	3111.71	5893.15	6462.36	202.45	383.40	420.44
IMN	3.58	0.68	2332.25	3340.54	3850.71	651.41	932.96	1075.47
IMT	4.17	0.43	1906.23	2348.20	2478.68	457.11	563.07	594.30
OD	17.32	1.00	1030.53	2226.92	2637.98	59.51	128.59	152.33
FS	18.21	1.00	957.05	2154.06	2503.09	52.55	118.27	137.43
Fussl	18.13	1.00	955.41	2150.59	2533.40	52.68	118.60	139.70
MHGB	10.60	0.29	360.72	702.89	809.20	34.03	66.30	76.33
MHG1	8.35	0.29	334.83	693.41	802.33	40.09	83.04	96.08
MHGH	22.15	1.00	958.02	2137.13	2477.10	43.25	96.48	111.84
GP	223.80	1.00	2588.07	5317.57	6059.81	11.56	23.76	27.08

German Credit Full: $N = 1000$, $P = 49$								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	22.30	1.00	2803.23	5748.30	6774.82	125.69	257.75	303.76
IMN	4.72	0.41	730.34	1050.29	1236.55	154.73	222.70	262.05
IMT	6.02	0.00	5.49	14.40	235.50	0.91	2.39	39.13
OD	25.34	1.00	717.94	2153.05	2655.86	28.33	84.96	104.80
FS	26.44	1.00	727.17	2083.48	2554.62	27.50	78.80	96.62
Fussl	26.91	1.00	755.31	2093.68	2562.11	28.06	77.80	95.21
MHGB	14.66	0.13	132.74	291.11	345.12	9.05	19.86	23.54
MHG1	12.45	0.13	136.57	290.13	345.22	10.97	23.31	27.73
MHGH	35.99	1.00	742.04	2075.41	2579.42	20.62	57.67	71.67
GP	243.41	1.00	2181.84	5353.41	6315.71	8.96	21.99	25.95

Synthetic, orthogonal predictors: $N = 150$, $P = 10$								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	3.83	1.00	6140.81	7692.04	8425.59	1604.93	2010.44	2201.04
IMN	1.87	0.78	3009.10	4114.86	4489.16	1609.67	2200.72	2397.94
IMT	1.54	0.64	3969.87	4403.51	4554.04	2579.84	2862.12	2960.05
OD	4.88	1.00	2325.65	3030.71	3590.09	476.36	620.74	735.29
FS	4.46	1.00	2162.42	2891.85	3359.98	484.91	648.41	753.38
Fussl	3.79	1.00	2207.30	2932.21	3318.37	583.11	774.58	876.59
MHGB	2.10	0.53	1418.07	1791.71	2030.70	676.70	854.94	968.96
MHG1	1.72	0.53	1386.35	1793.50	2022.31	805.40	1042.20	1174.97
MHGH	4.34	1.00	2170.71	2887.57	3364.68	500.67	666.18	776.37
GP	38.53	1.00	5581.31	7284.98	8257.91	144.85	189.07	214.32

Synthetic, factor predictors: $N = 500$, $P = 20$								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	8.70	1.00	1971.61	2612.10	2837.41	226.46	300.10	325.95
IMN	2.52	0.42	826.94	966.95	1119.81	327.98	382.96	443.65
IMT	2.59	0.34	1312.67	1387.94	1520.29	507.54	536.84	588.10
OD	9.67	1.00	428.12	573.75	652.30	44.28	59.36	67.48
FS	9.85	1.00	459.59	585.91	651.05	46.65	59.48	66.09
Fussl	9.51	1.00	422.00	564.95	639.89	44.39	59.43	67.31
MHGB	5.35	0.33	211.14	249.33	281.50	39.46	46.58	52.59
MHG1	4.17	0.32	201.50	239.50	280.35	48.37	57.51	67.30
MHGH	11.18	1.00	452.50	563.30	644.73	40.46	50.37	57.65
GP	114.98	1.00	748.71	1102.59	1386.08	6.51	9.59	12.06

3.4 Binary Logistic Mixed Model Data Sets

Synthetic: A synthetically generated data set with 5 groups, 100 observations within each group, and a single fixed effect.

Polls: Voting data from a Presidential campaign [Gelman and HIll, 2006]. The response indicates a vote for or against former President George H.W. Bush. There are 49 groups corresponding to states. Some states have few observations, necessitating a model that shrinks coefficients towards a global mean to get reasonable estimates. A single fixed effect corresponding to race is included. Entries with missing data were deleted to yield a total of 2015 observations.

Xerop: The Xerop data set from the `epicalc` R package [Chongsuvivatwong, 2012], examines if vitamin A deficiencies contribute to respiratory infections. Multiple observations of each individual were made. The data is grouped by individual id yielding a total of 275 random intercepts. A total of 5 fixed effects are included in the model—age, sex, height, stunted growth, and season—corresponding to an 8 dimensional regression coefficient after expanding the season covariate using dummy variables.

Appendix 4

Dynamic Binary Logistic Regression Benchmarks

4.1 Data Sets

Tokyo Rainfall: A data set found in Kitagawa [1987] that counts the days on which it rained over a two-year period. The data set includes a leap year.

Synth Low: 500 synthetic binary responses with 2 loosely correlated predictors.

Synth High: 500 synthetic binary responses with 2 highly correlated predictors.

4.2 Benchmarks

Dataset: Tokyo. Prior: $\phi = 1, W \sim IGa(150, 15)$. $T = 366$.								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	21.72	1.00	4352.80	7735.08	10081.46	200.43	356.18	464.22
Fussl	20.60	1.00	1627.61	3649.46	5428.93	79.03	177.19	263.60
CUBS	233.98	0.47	652.86	761.98	910.62	2.79	3.26	3.89
Dataset: Synth Low. Prior $\phi_i = 0.95, W_i = 0.172$. $T = 500, P = 2$.								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	28.34	1.00	8309.00	9395.80	9894.45	293.16	331.50	349.09
Fussl	29.08	1.00	5299.68	7646.07	9800.60	182.26	262.96	337.05
CUBS	609.04	0.68	2424.95	3930.01	5168.38	3.98	6.45	8.49
Dataset: Synth High. Prior: $\phi_i = 0.95, W_i = 0.172$. $T = 500, P = 2$.								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	28.46	1.00	8611.53	9441.44	9894.56	302.60	331.76	347.68
Fussl	29.19	1.00	6204.70	7879.60	9656.77	212.54	269.91	330.78
CUBS	610.43	0.61	2562.38	4291.96	6932.42	4.20	7.03	11.36
Dataset: Synth Low. Prior $\phi_i \sim N(0.95, 0.01), W_i \sim IGa(10, 1)$. $T = 500, P = 2$.								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	31.03	1.00	1348.88	7063.34	9914.37	43.46	227.65	319.53
Fussl	31.81	1.00	1213.67	5643.54	9060.83	38.15	177.40	284.82
CUBS	627.81	0.65	125.22	576.81	1695.55	0.20	0.92	2.70
Dataset: Synth High. Prior $\phi_i \sim N(0.95, 0.01), W_i \sim IGa(10, 1)$. $T = 500, P = 2$.								
Method	time	ARate	ESS.min	ESS.med	ESS.max	ESR.min	ESR.med	ESR.max
PG	31.02	1.00	686.61	4802.36	9523.97	22.13	154.81	307.01
Fussl	31.84	1.00	479.20	3774.41	8505.58	15.05	118.55	267.15
CUBS	618.25	0.55	181.61	541.04	1617.91	0.29	0.87	2.61

Table 4.1: The minimum, median, and maximum effective sample sizes and effective sampling rates calculated for dynamic binary and binomial logistic regression for the Pólya-Gamma technique, the method of Fussl et al. [2013], and the method of Ravines et al. [2006].

Appendix 5

Realized Kernel Construction

Our construction of the realized kernels is based upon Barndorff-Nielsen et al. [2009, 2011]. Barndorff-Nielsen et al.'s model, which takes into account market microstructure noise, is

$$X_{t_i} = Y_{t_i} + U_{t_i}$$

where $\{t_i\}_{i=1}^n$ are the times at which the m-dimensional vector of log stock prices, $\{X_t\}_{t\geq 0}$, are observed, $\{Y_t\}_{t\geq 0}$ is the latent log stock price, and $\{U_{t_i}\}_{t=1}^n$ are errors introduced by market microstructure. The challenge is to construct estimates of the quadratic variation of $\{Y_t\}$ with the noisy data $\{X_{t_i}\}_{i=1}^n$. They do this using a kernel approach,

$$K(X_t) = \sum_{h=-H}^{H} k\Big(\frac{h}{H}\Big)\Gamma_h$$

where

$$\Gamma_h(X_t) = \sum_{j=h+1}^{n} x_j x'_{j-h}, \text{ for } h \geq 0,$$

with $x_j = X_{s_j} - X_{s_{j-1}}$ and $\Gamma_h = \Gamma'_{-h}$ for $h < 0$. The kernel $k(x)$ is a weight function and lives within a certain class of functions. While this provides a convenient formula for calculating realized kernels, the choice of weight function and proper bandwidth H requires some nuance. Barndorff-Nielsen et al. [2011] discuss both issues. We follow their suggestion, using the Parzen kernel for the weight function and picking

H as the average of the collection of bandwidths $\{H_i\}_{i=1}^m$ one calculates for each asset individually. Before addressing either of those issues one must address the practical problem of cleansing and synchronizing the data.

Clean the data : The data was cleaned using the following rules.

- Retrieve prices from only one exchange. For most companies we used the NYSE, but for Cisco, Intel, and Microsoft we used FINRA's Alternative Display Facility.
- If there are several trades with the same time stamp, which is accurate up to seconds, then the median price across all such trades is taken to be the price at that time.
- Discard a trade when the price is zero.
- Discard a trade when the correction code is not zero.
- Discard a trade when the condition code is a letter other than 'E' or 'F'.

Synchronize Prices : Regarding synchronization, prices of different assets are not updated at the same instant in time. To make use of the statistical theory for constructing the realized measures one must decide how to "align" prices in time so that they appear to be updated simultaneously. Barndorff-Nielsen et al. suggest constructing a set of refresh times (τ_j) which corresponds to a "last most recently updated approach." The first refresh time τ_1 is the first time at which all asset prices have been updated. The subsequent refresh times are inductively defined so that τ_n is the first time at which all assets prices have

been updated since τ_{n-1}. After cleansing and refreshing the data, one is left with the collection (X_{τ_j}) from which the realized kernels will be calculated.

Jitter End Points : For their asymptotic results to hold Barndorff-Nielsen et al. suggest jittering the first and last observations (X_{τ_j}). We do this by taking the average of the first two observations and relabeling the resulting quantity as the first observation and taking the average of the last two observations and labeling the resulting quantity as the last observation.

Calculate Bandwidths :

We follow Barndorff-Nielsen et al. [2009] when calculating each H_i individually using the time series $\{X_t^{(i)}\}$ before it has been synchronized or jittered. For the moment, we assume that i is fix and suppress it from the notation. In particular, for each asset i the bandwidth H is estimated as

$$\hat{H} = c^*(\hat{\xi}^2)^{2/5} n^{3/5}$$

where $c^* = 0.97$ for the Parzen kernel, n is the number of observations, and the estimate of ξ^2 is

$$\hat{\xi}^2 = \hat{\omega}^2 / \widehat{IV}.$$

$\widehat{IV}$ is the realized variance sampled on a 20 minute grid. $\hat{\omega}^2$ is an estimate of the variance of U_{τ_i} and is given by

$$\hat{\omega}^2 = \frac{1}{q} \sum_{k=1}^{q} \hat{\omega}_k^2 \quad \text{with} \quad \hat{\omega}_k^2 = \frac{RV_{dense}^{(k)}}{2n_{(k)}}.$$

The quantity $RV^{(k)}_{dense}$ is the sum of square increments taken at a high frequency.

$$RV^{(k)}_{dense} = \sum_{j=0}^{n_k-1} x_j^{(k)2},\ x_j^k = (X_{qj+k} - X_{q(j-1)+k}), k = 1, \ldots, q.$$

and n_k is the number of observations elements in $\{x_j^k\}_{j=1}^{n_k}$. For each time series we chose $q = \lfloor n/195 \rfloor$, which is the average number of ticks on that day per two minute period [Barndorff-Nielsen et al., 2009].

Calculate Realized Kernel Estimate :

At the end of the day one has $\{X_{\tau_j}\}$ and H. Using these quantities the realized covariation matrix is estimated using the kernel method described at the beginning of the section.

Bibliography

O. Aguilar. *Latent Structure in Bayesian Multivariate Time Series Models*. PhD thesis, Duke University, 1998.

O. Aguilar and M. West. Bayesian dynamic factor models and portfolio allocation. *Journal of Business and Economic Statistics*, 18(3):338–357, July 2000.

Y. Ait-Sahalia, P. A. Mykland, and L. Zhang. Ultra high frequency volatility estimation with dependent microstructure noise. *Journal of Econometrics*, 160(1): 160–175, January 2011.

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

T. G. Andersen, T. Bollerslev, F. X. Diebold, and H. Ebens. The distribution of realized stock return volatility. *Journal of Financial Econometrics*, 61:43–76, 2001.

T. G. Andersen, T. Bollerslev, F. X. Diebold, and P. Labys. Modeling and forecasting realized volatility. *Econometrica*, 71(2):579–625, March 2003.

D. F. Andrews and C. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(36):99–102, 1974.

O. Barndorff-Nielsen. *Information and Exponential Families*. John Wiley & Sons, 1978.

O. Barndorff-Nielsen and D. R. Cox. Edgeworth and saddle-point approximations with statistical applications. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41:279–312, 1979.

O. E. Barndorff-Nielsen and N. Shephard. Econometric analysis of realized covariation: High frequency based covariance, regression, and correlation in financial economics. *Econometrica*, 72(3):885–925, May 2004.

O. E. Barndorff-Nielsen, P. R. Hansen, A. Lunde, and N. Shephard. Designing realized kernels to measure the ex post variation of equity prices in the presence of noise. *Econometrica*, 76:1481–1536, 2008.

O. E. Barndorff-Nielsen, P. R. Hansen, A. Lunde, and N. Shephard. Realized kernels in practice: Trades and quotes. *Econometrics Journal*, 12(3):C1–C32, 2009.

O. E. Barndorff-Nielsen, P. R. Hansen, A. Lunde, and N. Shephard. Multivariate realized kernels: Consistent positive semi-definite estimators of the covariation of equity prices with noise and non-synchronous trading. *Journal of Econometrics*, 162:149–169, 2011.

A. Basilevsky. *Statistical Factor Analysis and Related Methods*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, Inc., 1994.

G. H. Bauer and K. Vorkink. Forecasting multivariate realized stock market volatility. *Journal of Econometrics*, 160:93–101, 2011.

P. Biane, J. Pitman, and M. Yor. Probability laws related to the Jacobi theta and Riemann zeta functions, and brownian excursions. *Bulletin of the American Mathematical Society*, 38:435–465, 2001.

P. Billingsley. *Probability and Measure*. John Wiley & Sons, second edition, 1986.

T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307–327, 1986.

R. W. Butler. *Saddlepoint Approximations with Applications*. Cambridge University Press, 2007.

A. Canty and B. Ripley. *boot: Bootstrap R (S-Plus) Functions*, 1.3-4 edition, 2012.

B. P. Carlin, N. G. Polson, and D. S. Stoffer. A Monte Carlo approach to non-normal and nonlinear state-space modeling. *Journal of the American Statistical Association*, 87:493–500, 1992.

C. K. Carter and R. Kohn. On gibbs sampling for state space models. *Biometrika*, 81:541–533, 1994.

C. M. Carvalho, H. F. Lopes, and O. Aguilar. Dynamic stock selection strategies: A structured factor model framework. In *Bayesian Statistics 9*. Oxford University Press, 2011.

G. Casella and R. L. Berger. *Statistical Inference*. Duxbury Advanced Series, 2002.

S. Chib, F. Nardari, and N. Shephard. Analysis of high dimensional stochastic volatility models. *Journal of Econometrics*, 134:341–371, 2006.

S. Chib, Y. Omori, and M. Asai. Multivariate stochastic volatility. In T. Andersen, R. Davis, J.-P. Kreiss, and T. Mikosch, editors, *Handbook of Financial Time Series*, pages 365–400. Springer-Verlag, 2009.

R. Chiriac and V. Voev. Modelling and forecasting multivariate realized volatility. *Journal of Applied Econometrics*, 2010.

V. Chongsuvivatwong. *epicalc: Epidemiological calculator*, 2012. URL `http://CRAN.R-project.org/package=epicalc`. R package version 2.15.1.0.

R. V. Churchill and J. W. Brown. *Complex Variables and Applications*. McGraw-Hill, 1984.

Z. Ciesielski and S. J. Taylor. First passage times and sojourn density for Brownian motion in space and the exact Hausdorff measure of the sample path. *Trans. Amer. Math. Soc.*, 103:434–450, 1962.

J. Dagpunar. Sampling of variates from a truncated gamma distribution. *Journal of Statistical Computation and Simulation*, 8:59–64, 1978.

H. E. Daniels. Saddlepoint approximations in statistics. *Annals of Mathematical Statistics*, 25:631–650, 1954.

L. Devroye. *Non-uniform random variate generation*. Springer-Verlag, 1986.

L. Devroye. On exact simulation algorithms for some distributions related to Jacobi theta functions. *Statistics & Probability Letters*, 79:2251–2259, 2009.

L. Devroye. Random variate generation for the generalized inverse gaussian distribution. Available on Devroye's website., November 2012. URL `http://luc.devroye.org/devs.html`.

J. A. Díaz-García. A note on sirivastava's paper "singular wishart and multivariate beta distributions": Jacobians. Technical report, Universidad Autónoma Agraria Antonio Narro, 2003.

E. F. Fama and K. R. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33:3–56, 1993.

E. B. Fox and M. West. Autoregressive models for variance matrices: Stationary inverse Wishart processes. Technical report, Duke University, July 2011.

S. Frühwirth-Schnatter. Data augmentation and dynamic linear models. *Journal of Time Series Analysis*, 15:183–202, 1994.

S. Frühwirth-Schnatter. *Finite Mixture and Markov Switch Models*. Springer, 2007.

S. Frühwirth-Schnatter and R. Frühwirth. Auxiliary mixture sampling with applications to logistic models. *Computational Statistics and Data Analysis*, 51:3509–3528, 2007.

S. Frühwirth-Schnatter and R. Frühwirth. Data augmentation and MCMC for binary and multinomial logit models. In *Statistical Modelling and Regression Structures*, pages 111–132. Springer-Verlag, 2010. Available from UT library online.

S. Frühwirth-Schnatter, R. Frühwirth, L. Held, and H. Rue. Improved auxiliary mixture sampling for hierarchical models of non-Gaussian data. *Statistics and Computing*, 19:479–492, 2009.

A. Fussl. *binomlogit: Efficient MCMC for Binomial Logit Models*, 2012. URL `http://CRAN.R-project.org/package=binomlogit`. R package version 1.0.

A. Fussl, S. Frühwirth-Schnatter, and R. Frühwirth. Efficient MCMC estimation of binomial logit models, September 2011. URL `http://www.stat.tugraz.at/Statistiktage2011/PFussl.pdf`.

A. Fussl, S. Frühwirth-Schnatter, and R. Frühwirth. Efficient MCMC for binomial logit models. Currently under revision, 2013.

D. Gamerman. Sampling from the posterior distribution in generalized linear mixed models. *Statistics and Computing*, 7:57–68, 1997.

D. Gamerman. Markov chain monte carlo for dynamic generalised linear models. *Biometrika*, 85(1):215–227, March 1998.

A. Gelman and J. HIll. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, 2006.

J. Geweke and H. Tanizaki. Bayesian estimation of state-space models using the Metropolis-Hastings algorithm wtihin gibbs sampling. *Computational Statistics and Data Analysis*, 37:151–170, 2001.

S. J. Godsill, A. Doucet, and M. West. Monte carlos smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99:156–168, 2004.

C. Gourieroux, J. Jasiak, and R. Sufana. The Wishart autoregressive process of multivariate stochastic volatility. *Journal of Econometrics*, 150:167–181, 2009.

R. B. Gramacy. *reglogit: Simulation-based Regularized Logistic Regression*, 2012. URL `http://CRAN.R-project.org/package=reglogit`. R package version 1.1.

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

J. D. Hamilton. *Time Series Analysis*. Princeton University Press, 1994.

J. Harrison and M. West. *Bayesian Forecasting and Dynamic Models*. Springer Verlag, 1997.

S. Hays, H. Shen, and J. Z. Huang. Functional dynamic factor models with applications to yield curve forecasting. *Annals of Applied Statistics*, 6(3):870–894, 2012.

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

E. Jacquier, N. G. Polson, and P. E. Rossi. Bayesian analysis of stochastic volatility models. *Journal of Business and Economic Statistics*, 12:371–389, 1994.

J. L. Jensen. *Saddlepoint Approximations*. Oxford Science Publications, 1995.

J. T. Kent. Eigenvalue expansions for diffusion hitting times. *Z. Wahrscheinlichkeitstheorie verw. Gebiete*, 52:309–319, 1980.

S. Kim, N. Shephard, and S. Chib. Stochastic volatility: Likelihood inference and comparison with ARCH models. *The Review of Economic Studies*, 65(3):361–393, Jul. 1998.

G. Kitagawa. Non-Gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82:1032–1041, 1987.

S. J. Koopman, B. Jungbackera, and E. Hol. Forecasting daily variability of the S&P 100 stock index using historical, realised and implied volatility measurements. *Journal of Empirical Finance*, 12:445–475, 2005.

Q. Liu. On portfolio optimization: How and when do we benefit from high-frequency data? *Journal of Applied Econometrics*, 24:560–582, 2009.

H. F. Lopes and C. M. Carvalho. Factor stochastic volatility with time varying loadings and Markov switching regimes. *Journal of Statistical Planning and Inference*, 137:3082–3091, 2007.

H. F. Lopes and H. S. Migon. *Case Studies in Bayesian Statistics*, chapter Comovements and Contagion in Emergent Markets: Stock Indexes Volatilities, pages 287–302. Springer, 2002.

H. F. Lopes and M. West. Bayesian model assessment in factor analysis. *Statistica Sinica*, 14:41–67, 2004.

R. Lugannani and S. Rice. Saddle point approximations for the distribution of the sum of independent random variables. *Applied Probability*, 12:475–490, 1980.

Machine Learning Repository. Statlog (australian credit approval) data set, 2012a. URL http://archive.ics.uci.edu/ml/datasets/Statlog+(Australian+Credit+Approval).

Machine Learning Repository. Statlog (german credit data) data set, 2012b. URL http://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data).

Machine Learning Repository. Statlog (heart) data set, 2012c. URL http://archive.ics.uci.edu/ml/datasets/Statlog+(Heart).

Machine Learning Repository. Pima indians diabetes data set, 2012d. URL http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes.

D. McFadden. The measurement of urban travel demand. *Journal of Public Economics*, 3:303–328, 1974.

D. McLeish. Simulating random variables using moment generating functions and saddlepoint approximations. Technical report, University of Waterloo, 2010.

D. C. Montgomery, L. A. Johnson, and J. S. Gardiner. *Forecasting and Time Series Analysis*. McGraw-Hill, Inc., second edition, 1990. Original edition released in 1976.

R. J. Muirhead. *Aspects of Multivariate Statistical Theory*. Wiley, 1982.

J. Murray. *Asymptotic Analysis*. Clarendon Press, 1974.

J. Nakajima and M. West. Latent threshold dynamic factor process modelling: Case study in spatio-temporal eeg analysis. Technical Report 12-zz, Duke University, 2012.

D. Noureldin, N. Shephard, and K. Sheppard. Multivariate high-frequency-based volatility (HEAVY) models. *Applied Econometrics*, 27(6):907–933, 2011.

S. M. O'Brien and D. B. Dunson. Bayesian multivariate logistic regression. *Biometrics*, 60:739–746, 2004.

I. Olkin and H. Rubin. Multivariate beta distributions and independence properties of the Wishart distributions. *Annals of Mathematical Statistics*, 35(1):261–269, 1964.

L. L. Pennisi. *Elements of Complex Variables*. Holt, Rinehart and Winston, 1976.

A. Philipov and M. E. Glickman. Multivariate stochastic volatility via Wishart processes. *Journal of Business and Economic Statistics*, 24:313–328, July 2006.

A. Philippe. Simulation of right and left truncated gamma distributions by mixtures. *Statistics and Computing*, 7:173–181, 1997.

M. K. Pitt and N. Shephard. Time varying covariances: a factor stochastic volatility approach. *Bayesian Statistics*, 6:547–570, 1999.

M. K. Pitt and S. G. Walker. Constructing stationary time series models using auxiliary variables with applications. *Journal of the American Statistical Association*, 100(470):554–564, 2005.

M. Plummer, N. Best, K. Cowles, and K. Vines. CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, 2006. URL `http://CRAN.R-project.org/doc/Rnews/`.

N. G. Polson, J. G. Scott, and J. Windle. Bayeslogit, March 2013a. URL `http://cran.r-project.org/web/packages/BayesLogit/index.html`. R Package for simulating Polya-Gamma random variates.

N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using Pólya-gamma latent variables, February 2013b. URL `http://arxiv.org/abs/1205.0310`.

R. Prado and M. West. *Time Series: Modeling, Computation, and Inference*, chapter Multivariate DLMs and Covariance Models, pages 263–319. Chapman & Hall/CRC, 2010.

J. M. Quintana and M. West. An analysis of international exchange rates using multivariate DLMs. *The Statistician*, 36:275–281, 1987.

C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

R. R. Ravines, H. S. Migon, and A. M. Schmidt. An efficient sampling scheme for generalized dynamic models. Working Paper, January 2006.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2005.

B. Rosenberg and W. McKibben. The prediction of systematic and specific risk in common stocks. *The Journal of Financial and Quantitative Analysis*, 8:317–333, 1973.

P. E. Rossi. *bayesm: Bayesian Inference for Marketing/Micro-econometrics*, 2012. URL `http://CRAN.R-project.org/package=bayesm`. R package version 2.2-5.

P. E. Rossi, G. M. Allenby, and R. McCulloch. *Bayesian Statistics and Marketing*. John Wiley & Sons, 2005.

K.-I. Sato. *Lévy Processes and Infinitely Divisible Distributions*. Cambridge University Press, 1999.

W. F. Sharpe. Capital asset prices: a theory of market equilibrium under conditions of risk. *The Journal of Finance*, 19:425–442, 1964.

N. Shephard. Local scale models: State space alternative to integrated garch processes. *Journal of Econometrics*, 60:181–202, 1994.

N. Shephard and M. K. Pitt. Likelihood analysis of non-Gaussian measurement time series. *Biometrika*, 84:653–667, 1997.

S. J. Taylor. *Financial Returns Modelled by the Product of Two Stochastic Processes–a Study of Daily Sugar Prices 1961-1979*, pages 203–226. Amersterdam: North-Holland., 1982.

H. Uhlig. Bayesian vector autoregressions with stochastic volatility. *Econometrica*, 65(1):59–73, Jan. 1997.

R. W. M. Wedderburn. Quasi-likelihood functions, generalized linear models, and the Gauss-Newton method. *Biometrika*, 61:439–447, 1974.

M. West, J. Harrison, and H. S. Migon. Dynamics generalized linear models and bayesian forecasting. *Journal of the American Statistical Association*, 80(389):73–83, March 1985.

J. Windle and C. M. Carvalho. Forecasting high-dimensional, time-varying covariance matrices for portfolio selection. The University of Texas at Austin, 2012.

L. Zhang, P. A. Mykland, and Y. Ait-Sahalia. A tale of two time scales: Determining integrated volatility with noisy high-frequency data. *Journal of the American Statistical Association*, 100(472):1394–1411, 2005.

X. Zhou, J. Nakajima, and M. West. Dynamic dependent factor models: Improving forecasts and portfolio decisions in financial time series. Technical Report 2012-09, Duke University, 2012. URL `http://ftp.stat.duke.edu/WorkingPapers/11-16.html`. Under review at: International Journal of Forecasting.