Data augmentation for models based on rejection sampling

Vinayak Rao, Lizhen Lin, and David Dunson
Department of Statistical Science, Duke University, USA
{VAR11, LIZHEN, DUNSON}@STAT.DUKE.EDU

Abstract

We present a data augmentation scheme to perform Markov chain Monte Carlo inference for models where data generation involves a rejection sampling algorithm. Our idea is to instantiate the rejected proposals preceding each data point, and we show that this can be done easily and efficiently. The resulting joint probability over observed and rejected variables can be much simpler than the marginal distribution over the observed variables, which often involve intractable integrals. Our algorithm is an instance of a growing body of work on exact Markov chain Monte Carlo inference for doubly-intractable distributions and we consider two such problems. The first is a Bayesian analysis of the matrix Langevin distribution on the Stiefel manifold, and the second Bayesian inference for a nonparametric Gaussian process density model. Our experiments demonstrate superior performance over state-of-the-art sampling algorithms for such problems.

Keywords: Bayesian inference; Density estimation; Doubly intractable; Gaussian process; Matrix Langevin; Markov Chain Monte Carlo; Rejection sampling; Stiefel manifold.

1. Introduction

Rejection sampling allows us to sample from a probability density $p(x)$ by constructing an upper bound to $p(x)$, and accepting or rejecting samples from a density proportional to the bounding envelope. The envelope usually is much simpler than $p(x)$, with the number of rejections, and thus the algorithm’s efficiency, determined by how closely it matches the true density.

In typical applications, the probability density of interest is indexed by a parameter $\theta$, and we write it as $p(x \mid \theta)$. A Bayesian analysis places a prior on $\theta$, and given observations from the likelihood $p(x \mid \theta)$, studies the posterior over $\theta$. An intractable likelihood, often with a normalization constant depending on $\theta$, precludes straightforward Markov chain Monte Carlo inference over $\theta$: calculating a Metropolis-Hastings acceptance ratio involves evaluating the ratio of two such likelihoods, and is itself intractable. This class of problems is called ‘doubly intractable’ (Murray et al. 2006), and existing approaches require the ability to draw exact samples from $p(x \mid \theta)$, or to obtain positive unbiased estimates of $p(x \mid \theta)$.
We describe a different approach that is applicable when \( p(x \mid \theta) \) has an associated rejection sampling algorithm. Our idea is to instantiate the rejected proposals that preceded each observation, resulting in an augmented state-space on which we run a Markov chain. Including the rejected proposals as auxiliary variables eliminates any intractable terms, and allows the application of standard techniques. Importantly, we show that conditioned on the observations, it is straightforward to independently sample the number and values of the rejected proposals: this just requires running the rejection sampler to generate as many acceptances as there are observations, with all rejected proposals kept. The ability to produce a conditionally independent draw of these variables is important when posterior updates of some parameters are intractable, while others are simple. In such a situation, we introduce the rejected variables only when we need to carry out the intractable updates, after which we discard them and carry out the simpler updates.

2. Rejection sampling

Consider a probability density \( p(x \mid \theta) = f(x, \theta) / Z(\theta) \) on some space \( \mathbb{X} \), with the parameter \( \theta \) taking values in \( \Theta \). We assume that the normalization constant \( Z(\theta) \) is difficult to evaluate, so that naively sampling from \( p(x \mid \theta) \) is not easy. We also assume there exists a second, simpler density \( q(x \mid \theta) \geq f(x, \theta) / M \) for all \( x \), and for some positive \( M \).

Rejection sampling generates samples distributed as \( p(\cdot \mid \theta) \) by first proposing samples from \( q(\cdot \mid \theta) \). A draw \( y \) from \( q(\cdot \mid \theta) \) is accepted with probability \( f(y, \theta) / \{ Mq(y \mid \theta) \} \).

Let there be \( r \) rejected proposals preceeding an accepted sample \( x \), and denote them by \( \mathcal{Y} = \{y_1, \cdots, y_r\} \) where \( r \) itself is a random variable. Write \(|\mathcal{Y}| = r\), so that the joint probability is

\[
p(\mathcal{Y}, x) = \prod_{i=1}^{\left|\mathcal{Y}\right|} q(y_i \mid \theta) \left\{ 1 - \frac{f(y_i, \theta)}{Mq(y_i \mid \theta)} \right\} q(x \mid \theta) \left\{ \frac{f(x, \theta)}{Mq(x \mid \theta)} \right\} = \frac{f(x, \theta)}{M} \prod_{i=1}^{\left|\mathcal{Y}\right|} \left\{ q(y_i \mid \theta) - \frac{f(y_i, \theta)}{M} \right\}.
\] (1)

It is well known that this procedure recovers samples from \( p(x \mid \theta) \), so that the expression above has the correct marginal distribution over \( x \) (Robert and Casella (2005)). Later, we will need to sample the rejected variables \( \mathcal{Y} \) given an observation \( x \) drawn from \( p(\cdot \mid \theta) \). Simulating from \( p(\mathcal{Y} \mid x, \theta) \) involves the following two steps in Algorithm 1.

Algorithm 1 relies on the following proposition about the conditional \( p(\mathcal{Y} \mid x, \theta) \).

**Proposition 1** The set of rejected samples \( \mathcal{Y} \) preceding an accepted sample \( x \) is independent of \( x \): \( p(\mathcal{Y} \mid \theta, x) = p(\mathcal{Y} \mid \theta) \). We can thus assign \( x \) the set \( \hat{\mathcal{Y}} \) of another sample, \( \hat{x} \).
Algorithm 1 Algorithm to sample from $p(Y \mid x, \theta)$

**Input:** A sample $x$, and the parameter value $\theta$

**Output:** The set of rejected proposals $Y$ preceding $x$

1: Draw sample $y_i$ independently from $q(\cdot \mid \theta)$ until a point $\hat{x}$ is accepted.
2: Discard $\hat{x}$, and treat the preceding rejected proposals as $Y$.

3. Bayesian Inference

3.1 Sampling by introducing rejected proposals of the rejection sampler

Given observations $X = \{x_1, \ldots, x_n\}$, and a prior $p(\theta)$, Bayesian inference typically uses Markov chain Monte Carlo to sample from the intractable posterior $p(\theta \mid X)$. We partition $\theta$ into $(\theta_1, \theta_2)$ so that the normalization constant factors as $Z(\theta) = Z_1(\theta_1)Z_2(\theta_2)$, with the first term $Z_1$ simple to evaluate, and $Z_2$ intractable. Consequently, updating $\theta_1$ with $\theta_2$ fixed is straightforward; in fact there are situations where we can place a conjugate prior on $\theta_1$. Inference over $\theta_2$ is a doubly-intractable problem.

We assume that $p(x \mid \theta)$ has an associated rejection sampling algorithm with proposal density $q(x \mid \theta) \geq f(x, \theta)/M$. For the $i$th observation $x_i$, write the preceding set of rejected samples as $Y_i = \{y_{i1}, \ldots, y_{i|Y_i|}\}$. The joint density of all samples, both rejected and accepted, is then

$$P(x_1, Y_1, \ldots, x_n, Y_n) = \prod_{i=1}^{n} \frac{f(x_i, \theta)}{M} \prod_{j=1}^{|Y_i|} \left\{ q(y_{ij} \mid \theta) - \frac{f(y_{ij}, \theta)}{M} \right\}.$$ 

This does not involve any intractable terms, so that standard techniques can be applied to update $\theta$. To introduce the rejected proposals $Y_i$, we simply follow Algorithm 1: draw proposals from $q(\cdot \mid \theta)$ until we have $n$ acceptances, with the $i$th batch of rejected proposals forming the set $Y_i$.

The ability to produce conditionally independent draws of $Y$ is important when some components of $\theta$ have simple conditional posterior distributions. For instance, if there exists a conjugate prior $p_1(\theta_1)$ on $\theta_1$ for the likelihood $p(x \mid \theta_1, \theta_2)$, then introducing the rejected proposals $Y_i$ breaks this conjugacy. The resulting complications in updating $\theta_1$ can slow down mixing of the Markov chain, especially when $\theta_1$ is high dimensional. A much cleaner solution is to sample $\theta_1$ from its conditional posterior $p(\theta_1 \mid X, \theta_2)$, introducing the auxiliary variables only when needed to update $\theta_2$. After updating $\theta_2$, they can then be discarded. Algorithm 2 describes this.

3.2 Related work

One of the simplest, and most widely applicable Markov chain Monte Carlo algorithms for doubly-intractable distributions is the exchange sampler of Murray et al. (2006). Simplifying an earlier idea by Møller et al. (2006), this algorithm effectively amounts to the
Algorithm 2 An iteration of the Markov chain for posterior inference over $\theta = (\theta_1, \theta_2)$

**Input:** The observations $X$, and the current parameter values $(\theta_1, \theta_2)$

**Output:** New parameter values $(\hat{\theta}_1, \hat{\theta}_2)$

1: Run Algorithm 1 $|X|$ times, keeping all the rejected proposals $\mathcal{Y} = \bigcup_{i=1}^{|X|} \mathcal{Y}_i$.
2: Update $\theta_2$ with a Markov kernel having $p(\hat{\theta}_2 \mid X, \mathcal{Y}, \theta_1)$ as stationary distribution.
3: Discard the rejected proposals $\mathcal{Y}$.
4: Sample $\hat{\theta}_1$ from its posterior $p(\theta_1 \mid X, \theta_2)$.

following: given the current parameter $\theta_{\text{curr}}$, propose a new parameter $\theta_{\text{new}}$ according to some proposal distribution. Additionally, generate a dataset of $n$ ‘pseudo-observations’ $\{\hat{x}_i\}$ from $p(x \mid \theta_{\text{new}})$. The exchange algorithm then proposes swapping the parameters associated with datasets. [Murray et al. (2006)] show that all intractable terms cancel out in the resulting acceptance probability, and that the resulting Markov chain has the correct stationary distribution.

While the exchange algorithm is applicable whenever one can sample from the likelihood $p(x \mid \theta)$, it does not exploit the mechanism used to produce these samples. When the latter is a rejection sampling algorithm, each pseudo-observation is preceeded by a sequence of rejected proposals. These are all discarded, and only the accepted proposals are used to evaluate the new parameter $\theta_{\text{new}}$. By contrast our algorithm explicitly instantiates these rejected proposals, so that they can be used to make good proposals. In our experiments, we use a Hamiltonian Monte Carlo sampler on the augmented space and exploit gradient information to make nonlocal moves with high probability of acceptance. For reasonable acceptance probabilities under the exchange sampler, one must make local updates to $\theta$, or resort to complicated annealing schemes.

Another framework for doubly intractable distributions is the pseudo-marginal approach of [Andrieu and Roberts (2009)]. The idea here is that even if we cannot exactly evaluate the acceptance probability, it is sufficient to use a positive, unbiased estimate: this will still result in a Markov chain with the correct stationary distribution. In our case, instead of requiring an unbiased estimate, we require an upper bound $M$ to the normalization constant $Z(\theta)$. Additionally, like the exchange sampler, the pseudo-marginal method provides a mechanism to evaluate a proposed parameter $\theta_{\text{new}}$; how to make good proposals is less obvious.

Related to our ideas is a sampler from [Adams et al. (2009)], see also Section 6. Their problem also involved inferences on the parameters governing the output of a rejection sampling algorithm. Like us, they proceeded by augmenting the state space to include the rejected proposals $\mathcal{Y}$. However, rather than generating independent realizations of $\mathcal{Y}$ when needed, [Adams et al. (2009)] outlined a set of Markov transition operators to perturb the current configuration of $\mathcal{Y}$, while maintaining the correct stationary distribution. With prespecified probabilities, they proposed adding a new variable to $\mathcal{Y}$, deleting a variable from $\mathcal{Y}$ and perturbing the value of an existing element in $\mathcal{Y}$. These local updates to $\mathcal{Y}$
can slow down Markov chain mixing, require the user to specify a number of parameters, and also involve calculating Metropolis-Hastings acceptance probabilities for each local step. Furthermore, the Markov nature of their updates require them to maintain the rejected proposals at all times, complicating inferences over other parameters. Our algorithm is much simpler and cleaner.

4. Convergence properties

Write the Markov transition density of our chain as \( k(\hat{\theta} \mid \theta) \), and the \( m \)-fold transition density as \( k^m(\hat{\theta} \mid \theta) \). For simplicity, we suppress the fact that these depend on the data \( X \). The Markov chain is said to be uniformly ergodic if there exist constants \( C \) and \( \rho \) such that for all \( m \) and \( \theta \),

\[
\int_{\Theta} |p(\theta \mid X) - k^m(\hat{\theta} \mid \theta)|d\theta \leq C\rho^m.
\]

The term to the left is twice the total variation distance between the desired posterior and the state of the Markov chain initialized at \( \theta \) after \( m \) iterations. Small values of \( \rho \) imply faster mixing.

The following minorization condition is sufficient for uniform ergodicity (Jones and Hobert, 2001): there exists a probability density \( h(\hat{\theta}) \) and a \( \delta > 0 \) such that for all \( \theta, \hat{\theta} \in \Theta \),

\[
k(\hat{\theta} \mid \theta) \geq \delta h(\hat{\theta}).
\]  

When this holds, the mixing rate \( \rho \leq 1 - \delta \), so that a large \( \delta \) implies rapid mixing.

Our Markov transition density first introduces the rejected proposals \( \mathcal{Y}_i \), and then conditionally updates \( \theta \). The set \( \mathcal{Y}_i \) preceding the \( i \)th observation takes values in the union space \( \mathbb{U} \equiv \bigcup_{r=0}^{\infty} \mathbb{X}^r \). The output of the rejection sampler, including the \( i \)th observation, lies in the product space \( \mathbb{U} \times \mathbb{X} \) with density given by equation (1), so that any \((\mathcal{Y}, x) \in (\mathbb{U} \times \mathbb{X})\) has probability

\[
p(\mathcal{Y}, x \mid \theta) = \frac{f(x, \theta)}{M} \lambda(dx) \prod_{i=1}^{\mid\mathcal{Y}\mid} \left\{ q(y_i \mid \theta) - \frac{f(y_i, \theta)}{M} \right\} \lambda(dy_i).
\]

Here, \( \lambda \) is the measure with respect to which the densities \( f \) and \( q \) are defined, and it is easy to see that the above quantity integrates to 1. From Bayes’ rule, the conditional density over \( \mathcal{Y} \) is

\[
p(\mathcal{Y} \mid x, \theta) = \frac{1}{M} \prod_{i=1}^{\mid\mathcal{Y}\mid} \left\{ q(y_i \mid \theta) - \frac{f(y_i, \theta)}{M} \right\} \lambda(dy_i).
\]

The fact that the right hand side does not depend on \( x \) is an alternate proof of Proposition 1. This density characterizes the data augmentation step of our sampling algorithm. In practice, we need as many draws from this density as there are observations.
The second step of sampling from \( p(\hat{\theta} \mid Y, X, \theta) \) depends on the problem at hand. We simplify matters by assuming we can sample from \( p(\hat{\theta} \mid Y, X) \) independently of the old \( \theta \); this is the classical data augmentation algorithm. We also assume that the proposal density \( q(\cdot \mid \theta) \) is uniformly bounded from above and below by finite, positive quantities \( B \) and \( b \), and that the probability of rejection for the rejection sampler is bounded below by a positive \( r \). Letting \( \beta = br/B \), we can easily verify that \( p(\hat{\theta} \mid Y, X) \geq \beta Y p(\hat{\theta} \mid X) \). We can now state our result.

**Theorem 2** Assume that the proposal density \( q(x \mid \theta) \) has uniform lower and upper bounds \( b \) and \( B \), and the probability of rejection is bounded below by \( r \), with \( r, b > 0 \). Further assume we can sample from the conditional \( p(\hat{\theta} \mid Y, X) \). Then our data augmentation algorithm is uniformly ergodic with mixing rate \( \rho \) upper bounded by

\[
1 - \frac{1}{(M(1 - \beta) - \beta)^n}.
\]

where \( \beta = br/B \).

Despite our simplifying assumptions, our theorem has a number of useful implications. Firstly, the smaller \( M \) is, the lower the bound is, suggesting faster mixing. Recall that \( M \) is a measure of how well our proposal distribution \( q(\cdot \mid \theta) \) matches the distribution of interest \( f(\cdot \mid \theta) \). The theorem suggests that the closer the match, the faster the chain mixes. This makes intuitive sense: a poor proposal density will result in more rejected proposals \( Y \), and will increase the coupling between successive \( \theta \)’s in the Markov chain. The parameter \( r \) has a similar effect. Also, the larger the number of observations \( n \) is, the more slowly the chain mixes. Again, from the construction of our chain this makes sense. We suspect this property holds for most exact samplers for doubly-intractable distributions, though we are unaware of any such result.

Even without assuming we can sample from \( p(\hat{\theta} \mid Y, X) \), our ability to sample \( Y \) independently means that the marginal chain over \( \theta \) is still Markovian. By contrast, existing approaches (Adams et al. (2009); Walker (2011)) only produce dependent updates in the complicated auxiliary space: they sample from \( p(\hat{Y} \mid \theta, Y, X) \) by making local updates to \( Y \). Consequently, these chains are Markovian only in the complicated augmented space, and the marginal processes over \( \theta \) have long-term dependencies. Besides affecting mixing, this can also complicate analysis.

In the following sections we apply our sampling algorithm to two problems, one involving Bayesian inference for the matrix Langevin distribution, and the second, the Gaussian process density sampler of Adams et al. (2009).

### 5. Bayesian inference for the matrix Langevin distribution

#### 5.1 The Matrix Langevin distribution on the Stiefel manifold

The Stiefel manifold \( V_{p,d} \) is the space of all \( d \times p \) orthonormal matrices, that is, \( d \times p \) matrices \( X \) such that \( X^T X = I_p \), where \( I_p \) is the \( p \times p \) identity matrix. When \( p = 1 \), this is the \( d - 1 \) hypersphere \( S^{d-1} \), and when \( p = d \), this is the space of all the orthonormal matrices \( O(d) \). Probability distributions on the Stiefel manifold play an important role
in statistics, signal processing and machine learning, with applications ranging from studies of orientations of orbits of comets and asteroids to principal components analysis to the estimation of rotation matrices. The simplest such distribution is the matrix Langevin distribution, an exponential-family distribution whose density with respect to the invariant Haar volume measure (Edelman et al., 1998) is
\[
p_{\text{ML}}(X \mid F) = \exptr(F^T X) / Z(F).
\]
Here \( \exptr \) is the exponential-trace, and \( F \) is a \( d \times p \) matrix. The normalization constant
\[
Z(F) = F_1(\frac{1}{2} d, \frac{1}{4} F^T F)
\]
is the hypergeometric function with matrix arguments, evaluated at \( \frac{1}{4} F^T F \) (Chikuse, 2003). Let \( F = G \kappa H^T \) be the singular value decomposition of \( F \), where \( G \) and \( H \) are \( d \times p \) and \( p \times p \) orthonormal matrices, and \( \kappa \) is a positive diagonal matrix. We parametrize \( p_{\text{ML}} \) by \((G, \kappa, H)\), and one can think of \( G \) and \( H \) as orientations, with \( \kappa \) controlling the concentration in directions determined by these orientations. Large values of \( \kappa \) imply concentration along the associated directions, while setting \( \kappa \) to zero gives the uniform distribution on the Stiefel manifold. It can be shown (Khatri and Mardia, 1977) that
\[
0 F_1(\frac{1}{2} d, \frac{1}{4} F^T F) = 0 F_1(\frac{1}{2} d, \frac{1}{4} \kappa^T \kappa),
\]
so that this depends only on \( \kappa \). We write it as \( Z(\kappa) \).

In our Bayesian analysis, we place independent priors on \( \kappa, G \) and \( H \). The latter two lie on the Stiefel manifolds \( V_{p,d} \) and \( V_{p,p} \), and we place matrix Langevin priors \( p_{\text{ML}}(\cdot \mid F_0) \) and \( p_{\text{ML}}(\cdot \mid F_1) \) on these: we will see below that these are conditionally conjugate. We place independent Gamma\((a_0, b_0)\) priors on the diagonal elements of \( \kappa \). However, the difficulty in evaluating the normalization constant \( Z(\kappa) \) makes posterior inference over \( \kappa \) doubly intractable. Thus, in a 2006 University of Iowa PhD thesis, Camano-Garcia keeps \( \kappa \) constant, while Hoff (2009b) uses a first-order Taylor expansion of the intractable term to run an approximate sampling algorithm. Below, we show how fully Bayesian inference can be carried out over this quantity as well.

### 5.2 A rejection sampling algorithm

We first describe a rejection sampling algorithm from Hoff (2009a) to sample from \( p_{\text{ML}} \). For simplicity, assume \( H \) is the identity matrix. In the general case, we simply rotate the resulting draw by \( H \), since if \( X \sim p_{\text{ML}}(\cdot \mid F) \), then \( XH \sim p_{\text{ML}}(\cdot \mid FH^T) \). At a high level, the algorithm sequentially proposes vectors from the matrix Langevin on the unit sphere: this is also called von Mises-Fisher distribution and is easy to simulate (Wood, 1994). The mean of the \( r \)th vector is column \( r \) of \( G, G_{[r]} \), projected onto the nullspace of the earlier vectors, \( N_r \). This sampled vector is then projected back onto \( N_r \) and normalized, and the process is repeated \( p \) times. Call the resulting distribution \( p_{\text{seq}} \); for more details, see Algorithm 3 and Hoff (2009a).
Algorithm 3 Proposal distribution $p_{\text{seq}}(\cdot \mid G, \kappa)$ for matrix Langevin distribution (Hoff 2009a)

\textbf{Input:} Parameters $G, \kappa$; write $G_{[i]}$ for column $i$ of $G$, and $\kappa_i$ for element $(i, i)$ of $\kappa$

\textbf{Output:} An output $X \in V_{p,d}$; write $X_{[i]}$ for column $i$ of $X$

1: Sample $X_{[1]} \sim p_{\text{ML}}(\cdot \mid \kappa_1 G_{[1]}).$
   For $r \in \{2, \cdots p\}$

   (a) Construct $N_r$, an orthogonal basis for the nullspace of \{X_{[1]}, \cdots X_{[r-1]}\}.

   (b) Sample $z \sim p_{\text{ML}}(\cdot \mid \kappa_r N_r^T G_{[r]}), and

   (c) Set $X_{[r]} = z^T N_r / \|z^T N_r\|.$

It can be seen that $p_{\text{seq}}$ is a density on the Stiefel manifold taking the form

$$p_{\text{seq}}(X \mid G, \kappa) = \left\{ \prod_{r=1}^{p} \frac{\|\kappa_r N_r^T G_{[r]} \|^2 / (d-r-1)/2}{\Gamma(d-r+1)/2} \right\} \text{etr}(\kappa G^T X),$$

(3)

where $I_k(\cdot)$ is the modified Bessel function of the first kind. Write $D(X, \kappa, G)$ for the reciprocal of the term in braces. Since $I_k(x)/x^k$ is an increasing function of $x$, and $\|N_r^T G_{[r]}\| \leq \|G_{[r]}\| = 1$, we have the following bound $D(X, \kappa, G)$:

$$D(X, \kappa, G) \leq \prod_{r=1}^{p} \frac{\Gamma(d-r+1)/2 \|\kappa_r\|^2 (d-r-1)/2}{\|\kappa_r\|^2 (d-r-1)/2} = D(\kappa)$$

This implies $\text{etr}(\kappa G^T X) \leq D(\kappa) p_{\text{seq}}(X \mid G, \kappa)$, allowing the following rejection sampler: draw a sample $X$ from $p_{\text{seq}}(\cdot)$, and accept with probability $D(X, \kappa, G)/D(\kappa)$. The accepted proposals come from $p_{\text{ML}}(\cdot \mid G, \kappa)$, and for samples from $p_{\text{ML}}(\cdot \mid G, \kappa, H)$, postmultiply these by $H$.

5.3 Posterior sampling

Given a set of $n$ observations \{X_i\}, and writing $S = \sum_{i=1}^{n} X_i$, we have:

$$p(G, \kappa, H \mid X_i) \propto \text{etr}(H \kappa G^T S)p(H)p(G)p(\kappa)/Z(\kappa)^n.$$ 

At a high level, our approach is a Gibbs sampler that sequentially updates $H, G$ and $\kappa$. The first two steps are straightforward, while the third will require our data augmentation scheme.

1. Updating $G$ and $H$: With a matrix Langevin prior on $H$, the posterior is

$$p(H \mid X_i, \kappa, G) \propto \text{etr}\left\{ (S^T G \kappa + F_0)^T H \right\}.$$
This is just the matrix Langevin distribution over rotation matrices, and one can sample from this following Section 5.2. From here onwards, we will rotate the observations by $H$, allowing us to ignore this term. Redefining $S$ as $SH$, the posterior over $G$ is also a matrix Langevin:

$$p(G \mid X_i, \kappa) \propto \text{etr} \left\{ (S\kappa + F_i)^T G \right\} .$$

2. Updating $\kappa$: Here, we exploit the rejection sampler scheme of the previous section, and instantiate the rejected proposals using Algorithm 1. From Section 5.2, the joint probability is

$$p(\{X_i, Y_i\} \mid G, \kappa) = \frac{\text{etr} \left\{ \kappa G^T \left( S + \sum_{j=1}^{n} Y_{ij} \right) \right\}}{D(\kappa)^{1+|\mathcal{Y}|}} \prod_{i=1}^{n} \prod_{j=1}^{|\mathcal{Y}|} \frac{D(\kappa) - D(Y_{ij}, G, \kappa)}{D(Y_{ij}, G, \kappa)} . \quad (4)$$

All terms in the expression above can be evaluated easily, allowing a simple Metropolis-Hastings algorithm in this augmented space. In fact, we can go further, calculating gradients to run a Hamiltonian Monte Carlo algorithm (Neal, 2010) that makes significantly more efficient proposals than a random-walk sampling algorithm. In particular, let $N = n + \sum_{i=1}^{n} |Y_i|$, and $S = \sum_{i=1}^{n} (X_i + \sum_{j=1}^{n} |Y_i| Y_{ij})$. The log joint probability $L \equiv \log \left\{ p(\{X_i, Y_i\}) \right\}$ is

$$L = \text{trace}(G^T S) + n \sum_{i=1}^{n} \sum_{j=1}^{|Y_i|} \log \left\{ D(\kappa) - D(Y_{ij}, \kappa) \right\} - n \log \left\{ D(\kappa) \right\} .$$

Writing $D(Y, \kappa) = \left\{ C \prod_{r=1}^{p} \frac{I_{(d-r-1)/2}(\|\kappa r G_r\|)}{\|\kappa r G_r\|^{|d-r-1|/2}} \right\}$ as $C \tilde{D}(Y, \kappa)$, Appendix B shows that

$$\frac{dL}{d\kappa_k} = \frac{G_k^T S_k}{\|\kappa_k G_k\|^2} + \sum_{i=1}^{n} \sum_{j=1}^{|Y_i|} \left[ \frac{I_{(d-k-1)/2}(\kappa_k)}{I_{(d-1)/2}(\kappa_k N_k G_k)} \frac{I_{(d-k-1)/2}(\kappa_k N_k G_k)}{I_{(d-k)/2}(\kappa_k)} \right] - N \frac{I_{(d-k+1)/2}(\kappa_k)}{I_{(d-k)/2}(\kappa_k)} .$$

We use this gradient information to construct a Hamiltonian Markov chain Monte Carlo algorithm (Neal, 2010) that explores $\kappa$-space. For our purposes, it suffices to note that a proposal of this sampler involves taking $L$ ‘leapfrog’ steps of size $\epsilon$ along the gradient, and then accepting the resulting state with probability proportional to the product of equation (4), and a simple Gaussian ‘momentum’ term. The acceptance probability depends on how accurately the $\epsilon$-discretization approximates the continuous dynamics of the system, and choosing a small $\epsilon$ and a large $L$ can give global moves with high acceptance probability. On the other hand, a large $L$ comes at the cost of a large number of gradient evaluations. We study this trade-off in Section 5.5.

5.4 Vectorcardiogram dataset

The vectorcardiogram is a loop traced by the cardiac vector during a cycle of the heart beat. The two directions of orientation of this loop in three-dimensions form a point on the Stiefel
manifold. The dataset of Downs et al. (1971) includes 98 such recordings, and is displayed in the left subplot of Figure 1. We represent each observation with a pair of orthonormal vectors, with the set of cyan lines to the right forming the first component. This empirical distribution possesses a single mode, so that the matrix Langevin distribution appears a suitable model.

We place independent uninformative exponential priors with mean 10 and variance 100 on the scale parameter $\kappa$, and a uniform prior on the location parameter $G$. We restrict $H$ to be the identity matrix. Inferences were carried out using the Hamiltonian sampler to produce 10000 samples, with a burn-in period of 1000. For the leapfrog dynamics, we set a step size of 0.3, with the number of steps equal to 5. We fix the ‘mass parameter’ to the identity matrix as is typical. We implemented all algorithms in R, building on code from the rstiefel package of Peter Hoff. All simulations were run on an Intel Core 2 Duo 3 Ghz CPU. For comparison, we include the maximum likelihood estimates of $\kappa$ and $G$. For $\kappa_1$ and $\kappa_2$, these were 11.9 and 5.9, and we plot these in the right half of Figure 1 as the red circles.

The bold straight lines in Figure 1(left) show the maximum likelihood estimates of the components of $G$, with the small circles corresponding to 90% Bayesian credible regions estimated from the Monte Carlo output. The dashed circles correspond to 90% predictive probability regions for the Bayesian model. For these, we generated 50 points on $V_{3,2}$ for each sample, with parameters specified by that sample. The dashed circles contain 90% of these points across all samples. Figure 1(right) show the posterior over $\kappa_1$ and $\kappa_2$. 

Figure 1: (Left) Vector cardiogram dataset with inferences. Bold lines are maximum likelihood estimates of $G$, and solid circles contain 90% posterior mass. Dashed circles are 90% predictive probability regions. (Right) Posterior over $\kappa_1$ and $\kappa_2$, circles are maximum likelihood estimates.
Figure 2: Effective samples per second for (left) random walk and (right) Hamiltonian samplers. From bottom to top at abscissa 0.5: (left) Metropolis-Hastings data-augmentation sampler and exchange sampler, and (right) 1/10/5/3 leapfrog steps of Hamiltonian sampler.

5.5 Comparison of exact samplers

To quantify sampler efficiency, we estimate the effective sample sizes produced per unit time. This corrects for correlation between successive Markov chain samples by estimating the number of independent samples produced; for this we used the \texttt{r\,c\,o\,d\,a} package of \cite{Plummer2006}.

The left plot in Figure 2 considers two Metropolis-Hastings samplers, the exchange sampler and our latent variable sampler on the vectorcardiogram dataset. Both samplers perform a random walk in the $\kappa$-space, with the steps drawn for a normal distribution whose variance increases along the horizontal axis. The vertical axis shows the median effective sample size per second for the components of $\kappa$. The figure shows that both samplers’ performance peaks when the proposals have a variance between 1 and 1.5, with the exchange sampler performing slightly better. However, the real advantage of our sampler is that introducing the latent variables results in a joint distribution without any intractable terms, allowing the use of more sophisticated sampling algorithms. The plot to the right studies the Hamiltonian Monte Carlo sampler described at the end of Section 3.1. Here we vary the size of the leapfrog steps along the horizontal axis, with the different curves corresponding to different numbers of leapfrog steps. We see that this performs an order of magnitude better than either of the previous algorithms, with performance peaking with 3 to 5 steps of size 0.3 to 0.5, fairly typical values for this algorithm. This shows the advantage of exploiting gradient information in exploring the parameter space.

5.6 Comparison with an approximate sampler

In this section, we consider an approximate sampler based on an asymptotic approximation to $Z(\kappa) = \Phi_{1}(\frac{1}{2}d, \frac{1}{4}\kappa^T\kappa)$ for large values of $(\kappa_1, \ldots, \kappa_n)$ \cite{Khatri1977}:

$$Z(\kappa) \simeq \left\{ \frac{2^{-\frac{d}{2}p(p+5)+\frac{1}{2}pd}}{\pi^{\frac{d}{2}p}} \right\} \text{etr}(\kappa) \prod_{j=1}^{p} \Gamma \left( \frac{d - j + 1}{2} \right) \left\{ \prod_{j=2}^{p} \prod_{i=1}^{j-1} (\kappa_i + \kappa_j)^{\frac{1}{2}} \right\} \prod_{i=1}^{p} \kappa_i^{\frac{1}{2}(d-p)} \right\}^{-1}. $$
We use this approximation in the acceptance probability of a Metropolis-Hastings algorithm; it can similarly be used to construct a Hamiltonian sampler. Such approximate schemes involve the ratio of two approximations, and can have very unpredictable performance.

On the vectorcardiogram dataset, the approximate sampler based on the asymptotic expansion is about forty times faster than the exact samplers. For larger datasets, this difference will be even greater, and the real question is how accurate the approximation is. Our exact sampler allows us to study this: we consider the Stiefel manifold $V_{d,3}$, with the three diagonal elements of $\kappa$ set to 1, 5 and 10. With this setting of $\kappa$, and a random $G$, we generate datasets with 50 observations with $d$ taking values 3, 4, 5, 8, and 10. In each case, we estimate the posterior mean of $\kappa$ by running the exchange sampler, and treat this as the truth. We compare this with posterior means returned by our Hamiltonian sampler, as well as the approximate sampler. Figure 3 shows these results, with the three subplots corresponding to the three components of $\kappa$, and ambient dimensionality $d$ increasing along the horizontal axis. As expected, the two exact samplers agree, and the Hamiltonian sampler has almost no ‘error’. The approximate sampler is more complicated. For values of $d$ around 5, its estimated posterior mean is close to that of the exact samplers. Smaller values lead to an approximate posterior mean that underestimates the actual posterior mean, while in higher dimensions, the opposite occurs. Recalling that $\kappa$ controls the concentration of the matrix Langevin distribution about its mode, this implies that in high dimensions, the approximate sampler will underestimate uncertainty in the distribution of future observations.

6. The Gaussian process density sampler

Our next application is the Gaussian process density sampler of Adams et al. (2009), a non-parametric prior for a probability density induced by a logistic transformation of a random function from a Gaussian process. Letting $\sigma(\cdot)$ denote the logistic function, the random
density is
\[ g(x) \propto g_0(x) \sigma\{f(x)\}, \quad f \sim \text{GP}, \]
with \( g_0(\cdot) \) a parametric base density and \( \text{GP} \) denoting a Gaussian process. The inequality \( g_0(x) \sigma\{f(x)\} \leq g_0(x) \) allows a rejection sampling algorithm by making proposals from \( g_0(\cdot) \). At a proposed location \( x^* \), we sample the function value \( f(x^*) \) conditioning on all previous evaluations, and accept the proposal with probability \( \sigma\{f(x^*)\} \). Such a scheme involves no approximation error, and only requires evaluating the random function on a finite set of points. Algorithm 4 describes the steps involved in generating \( n \) observations.

Algorithm 4 Generate \( n \) new samples from the Gaussian process density sampler

\begin{algorithm}
\begin{enumerate}
\item \textbf{repeat}
\item Sample a proposal \( y \) from \( g_0(\cdot) \).
\item Sample \( f_y \), the Gaussian process evaluated at \( y \), conditioning on \( f_X \), \( f_Y \), \( f_{\tilde{X}} \) and \( f_{\tilde{Y}} \).
\item \textbf{with probability} \( \sigma(f_y) \)
\item Accept \( y \) and add it to \( X \). Add \( f_y \) to \( f_X \).
\item \textbf{else}
\item Reject \( y \) and add it to \( Y \). Add \( f_y \) to \( f_Y \).
\item \textbf{end}
\item \textbf{until} \( n \) samples are accepted.
\end{enumerate}
\end{algorithm}

6.1 Posterior inference

Given observations \( X = \{x_1, \ldots, x_n\} \), we are interested in \( p(g \mid X) \), the posterior over the underlying density. Since \( g \) is determined by the modulating function \( f \), we focus on \( p(f \mid X) \). While this quantity is doubly intractable, after augmenting the state space to include the proposals \( Y \) from the rejection sampling algorithm, \( p(f \mid X, Y) \) has density with respect to the Gaussian process prior given by \( \prod_{i=1}^n \sigma\{f(x_i)\} \prod_{|Y|} [1 - \sigma\{f(y_i)\}] \), see also [Adams et al.] (2009). In words, the posterior over \( f \) evaluated at \( X \cup Y \) is just the posterior from a Gaussian process classification problem with a logistic link-function, and with the accepted and rejected proposals corresponding to the two classes. There are a number of Markov chain Monte Carlo methods such as Hamiltonian Monte Carlo or elliptical slice sampling ([Murray et al.] 2010) that are applicable in such a situation. Given \( f \) on \( X \cup Y \), it can be evaluated anywhere else by conditionally sampling from a multivariate normal.
Sampling the rejected proposals $\mathcal{Y}$ given $X$ and $f$ is straightforward by Algorithm 1: run the rejection sampler until $n$ accepts, and treat the rejected proposals generated along the way as $\mathcal{Y}$. In practice, we do not have access to the entire function $f$, only its values evaluated on $X$ and $\mathcal{Y}_{\text{old}}$, the location of the previous thinned variables. However, just as under the generative mechanism, we can retrospectively evaluate the function $f$ where needed. After proposing from $g_0(\cdot)$, we sample the value of the function at this location conditioned on all previous evaluations, and use this value to decide whether to accept or reject. We outline the inference algorithm in Algorithm 5, noting that it is much simpler than that proposed in Adams et al. (2009). We also refer to that paper for limitations of the exchange sampler in this problem.

**Algorithm 5** A Markov chain iteration for inference in the Gaussian process density sampler

**Input:** Observations $X$ with corresponding function evaluations $\hat{f}_X$, Current rejected proposals $\hat{Y}$ with corresponding function evaluations $\hat{f}_Y$.

**Output:** New rejected proposals $Y$, New Gaussian process evaluations $f_X$ and $f_Y$ at $X$ and $Y$, New hyperparameters.

1: Run Algorithm 4 to produce $|X|$ accepted samples, with $X, \hat{Y}, \hat{f}_X$ and $\hat{f}_Y$ as inputs.
2: Replace $\hat{Y}$ and $\hat{f}_Y$ with values returned by the previous step; call these $Y$ and $f_Y$.
3: Update $\hat{f}_X$ and $f_Y$ using for example, hybrid Monte Carlo, to get $f_X$ and $f_Y$.
4: Update parameters of the base-distribution, as well as the Gaussian process hyperparameters.
Figure 5: (Left) Kernel density estimate of the locations of the rejected proposals and (Right) Histogram of the number of rejected proposals in each Markov chain iteration.

6.2 Experiments

We consider the galaxy dataset from [Postman et al. (1986)], consisting of recorded velocities of 82 galaxies in the Corona Borealis. We normalized these to vary from 0 to 10. We used the model of [Adams et al. (2009)] as a prior on the underlying probability density, setting $g_0(\cdot)$ to a normal $\mathcal{N}(\mu, \sigma^2)$, with a normal-inverse-Gamma prior on $(\mu, \sigma)$. The latter had parameters $(0, 1, 1, 10)$. The Gaussian process had a squared-exponential kernel, with variance and length-scale of 1.

We ran a Matlab implementation of our data augmentation algorithm to produce 2000 posterior samples after a burn-in of 500 samples. The Figure 4(left) shows the resulting posterior over the underlying density. This is clearly multimodal and non-Gaussian, with the modulating function allowing low-probability areas around either side of the main cluster of observations. The right bump away from the origin is slightly thinner than the left one because we have fewer observations there, but also since we centered our model around the origin. The right plot in the same figure focuses on the deviation from normality by plotting the posterior over the latent function. Inside the interval $[0, 10]$, there are two dips in the Gaussian process intensity at $[1, 3]$ and $[7, 9]$, while over $[3, 7]$, it is larger than its prior mean of 0.

Figure 5 studies the distribution of the rejected proposals $Y$. The left plot shows the distribution of their locations: most of these occured in the low probability intervals $[1, 3]$ and $[7, 9]$. The right plot is a histogram of the number of rejected proposals: this is typically around 100 to 200, though the largest value we observed was 611. Since inference over the latent function involves evaluating it at the locations of the accepted as well as rejected proposals, the largest covariance matrix we had to deal with was about $600 \times 600$; typical values were around $200 \times 200$. Using the same setup as Section 5.5, it took a naïve Matlab implementation 11 minutes to run 2500 iterations. One can imagine computations becoming unwieldy for a large number of observations, or when there is large mismatch between the true density and the base-measure $g_0(\cdot)$. In such situations, one might have to choose the Gaussian process covariance kernel more carefully, use one of many sparse approximation techniques, or use other nonparametric priors like splines instead. In all
these cases, we can use our algorithm to recover the rejected proposals \( \mathcal{Y} \), and given these, posterior inference over \( f \) can be carried out using standard techniques.

7. Discussion

We described a simple approach to carry out Markov chain Monte Carlo inference when data generation involves a rejection sampling algorithm. Our algorithm is simple and efficient, and allows us to exploit ideas like Hamiltonian Monte Carlo to carry out efficient inference. While our algorithm is exact, it also provides a framework for faster, approximate algorithms. For instance, the number of rejected proposals preceeding any observation is a random number that apriori is unbounded. One can bound the computational cost of an iteration by limiting the maximum number of rejected proposals. Similarly, one might try sharing rejected proposals across observations. We leave the study of the resulting Markov chain algorithms for future research. Also left open is a more careful analysis of Markov mixing rates for the applications we considered.

8. Acknowledgement

This work was supported by the National Institute of Environmental Health Sciences of the National Institute of Health.

Appendix A. Proofs

Proof [of Proposition 1]

The probability density of an accepted sample \( x \) is \( p(x \mid \theta) = \frac{f(x, \theta)}{Z(\theta)} \). Independently introduce the set \( (\mathcal{Y}, \hat{x}) \) by running the rejection sampler until acceptance, so that

\[
p(x, \mathcal{Y}, \hat{x} \mid \theta) = \frac{f(x, \theta)}{Z(\theta)} \left[ \frac{f(\hat{x}, \theta)}{M} \prod_{i=1}^{|\mathcal{Y}|} \left\{ q(y_i \mid \theta) - \frac{f(y_i, \theta)}{M} \right\} \right].
\]

Marginalizing out \( \hat{x} \), we have \( p(\mathcal{Y}, x) = \frac{f(x, \theta)}{M} \prod_{i=1}^{|\mathcal{Y}|} \left\{ q(y_i \mid \theta) - \frac{f(y_i, \theta)}{M} \right\} \). From equation (1), we see this is the desired joint distribution, proving our scheme is correct.

\[\square\]
Proof [of Theorem 2] Let the number of observations $|X|$ be $n$. Then,

$$k(\hat{\theta} \mid \theta) = \int_{\mathcal{Y}^n} p(\hat{\theta} \mid \mathcal{Y}, X) p(\mathcal{Y} \mid \theta, X) d\mathcal{Y}$$

$$\geq p(\hat{\theta} \mid X) \prod_{i=1}^{n} \int_{\mathcal{Y}_i} \beta^{q_\theta(y_i \mid \theta, X)} \prod_{j=1}^{n} \left\{ q(y_{ji} \mid \theta) - \frac{f(y_{ji}, \theta)}{M} \right\} \lambda(dy_{ji})$$

$$= \frac{p(\hat{\theta} \mid X)}{M^n} \prod_{i=1}^{n} \sum_{|\mathcal{Y}_i|=0}^{\infty} \beta^{q_\theta} \prod_{j=1}^{n} \left( 1 - \frac{1}{M} \right)$$

$$= \frac{p(\hat{\theta} \mid X)}{M^n} \prod_{i=1}^{n} \frac{1}{\delta^n} = \delta^n = M - \beta(M - 1) = M(1 - \beta) - \beta$$

Thus $k(\hat{\theta} \mid \theta)$ satisfies equation (2), with $\delta^n = 1/\left[ M \{1 - \beta\} \right]$, and $h(\hat{\theta})$, the posterior $p(\hat{\theta} \mid X)$.

Appendix B. Gradient information

For $n$ pairs $\{X_i, \mathcal{Y}_i\}$, with $N = n + \sum_{i=1}^{n} |\mathcal{Y}_i|$, and $S = \sum_{i=1}^{n} (X_i + \sum_{j=1}^{n} Y_{ij})$, we have

$$\log \{P(\{X_i, \mathcal{Y}_i\})\} = \text{trace}(G^T \kappa S) + \sum_{i=1}^{n} \sum_{j=1}^{n} |\mathcal{Y}_i| \log \{D(\kappa) - D(Y_{ij}, \kappa)\} - \log \{D(Y_{ij}, \kappa)\} - N \log \{D(\kappa)\}$$

Write $D(Y, \kappa) = \left\{ C \prod_{r=1}^{p} \frac{I_{(d-r+1)/2}(\|\mathcal{Y}_r N_r^T G_r\|)}{\|\mathcal{Y}_r N_r^T G_r\|^{(d-r+1)/2}} \right\}$ as $C \tilde{D}(Y, \kappa)$. Since $\frac{d}{dx} \left\{ I_m(x) \right\} = x^{-m} I_{m+1}(x)$,

$$\frac{d\tilde{D}(Y, \kappa)}{d\kappa_j} = N^T_j G_j \tilde{D}(Y, \kappa) \frac{I_{(d-j+1)/2}(\kappa_j N^T_j G_j)}{I_{(d-j-1)/2}(\kappa_j)} \text{ and } \frac{d\tilde{D}(\kappa)}{d\kappa_j} = \tilde{D}(\kappa) \frac{I_{(d-j+1)/2}(\kappa_j)}{I_{(d-j-1)/2}(\kappa_j)}.$$

Then, writing $L = \log \{P(\{X_i, \mathcal{Y}_i\})\}$, we have

$$\frac{dL}{d\kappa_k} = G^T_{[k]} S_{[k]} + \sum_{i=1}^{n} \sum_{j=1}^{n} |\mathcal{Y}_i| \left\{ \frac{\tilde{D}'(\kappa) - \tilde{D}'(Y_{ij}, \kappa)}{\tilde{D}'(Y_{ij}, \kappa)} - \frac{\tilde{D}'(Y_{ij}, \kappa)}{\tilde{D}(Y_{ij}, \kappa)} \right\} - N \frac{\tilde{D}'(\kappa)}{\tilde{D}(\kappa)}$$

$$= G^T_{[k]} S_{[k]} + \sum_{i=1}^{n} \sum_{j=1}^{n} \left\{ \frac{I_{(d-k+1)/2}(\kappa_k)}{I_{(d-k-1)/2}(\kappa_k)} - \frac{\tilde{D}'(Y_{ij}, \kappa)}{\tilde{D}'(Y_{ij}, \kappa)} \right\} - N \frac{I_{(d-k+1)/2}(\kappa_k)}{I_{(d-k-1)/2}(\kappa_k)}.$$
**References**

Ryan Prescott Adams, Iain Murray, and David J. C. MacKay. The Gaussian process density sampler. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Adv. Neural Inf. Process. Syst. 21*, pages 9–16. MIT Press, 2009.

Christophe Andrieu and Gareth O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *Ann. Stat.*, 37(2):697–725, April 2009. ISSN 0090-5364. doi: 10.1214/07-AOS574. URL [http://dx.doi.org/10.1214/07-AOS574](http://dx.doi.org/10.1214/07-AOS574).

Y. Chikuse. *Statistics on Special Manifolds*. Springer, New York, 2003.

T. Downs, J. Liebman, and W. Mackay. Statistical methods for vectorcardiogram orientations. In *Vectorcardiography 2: Proc. Xth Intn. Symp. Vectorcardiography (I. Hoffman, R.I. Hamby and E. Glassman, Eds.),* pages 216–222, 1971. North-Holland, Amsterdam.

A. Edelman, T.A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl*, 20(2):303–353, 1998.

P. D. Hoff. Simulation of the Matrix Bingham-von Mises-Fisher Distribution, with Applications to Multivariate and Relational Data. *J. Comp. Graph. Stat.*, 18(2):438–456, 2009a.

P. D. Hoff. A hierarchical eigenmodel for pooled covariance estimation. *J. R. Statist. Soc. B*, 71(5):971–992, 2009b.

Galin L. Jones and James P. Hobert. Honest Exploration of Intractable Probability Distributions via Markov Chain Monte Carlo. *Statist. Sci.*, 16(4):312–334, 11 2001. doi: 10.1214/ss/1015346317. URL [http://dx.doi.org/10.1214/ss/1015346317](http://dx.doi.org/10.1214/ss/1015346317).

C. G. Khatri and K. V. Mardia. The Von Mises-Fisher Matrix Distribution in Orientation Statistics. *J. R. Statist. Soc. B*, 39(1), 1977. ISSN 00359246. doi: 10.2307/2984884. URL [http://dx.doi.org/10.2307/2984884](http://dx.doi.org/10.2307/2984884).

J. Møller, A. N. Pettitt, R. Reeves, and K. K. Berthelsen. An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika*, 93(2):451–458, 2006. doi: 10.1093/biomet/93.2.451. URL [http://biomet.oxfordjournals.org/content/93/2/451.abstract](http://biomet.oxfordjournals.org/content/93/2/451.abstract).

Iain Murray, Zoubin Ghahramani, and David J. C. MacKay. MCMC for doubly-intractable distributions. In *Proc. 22nd Conf. Uncert. Artif. Intell.*, pages 359–366. AUAI Press, 2006.

Iain Murray, Ryan Prescott Adams, and David J.C. MacKay. Elliptical slice sampling. *J. Mach. Learn. Res. W&CP*, 9, 2010.
Radford M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 54:113–162, 2010.

Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, March 2006. URL http://CRAN.R-project.org/doc/Rnews/.

M. Postman, J. Huchra, and M. Geller. Probes of large-scale structure in the Corona Borealis region. *Astron. J.*, 92:1238–1247, 1986.

Christian P. Robert and George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 0387212396.

Stephen G. Walker. Posterior sampling when the normalizing constant is unknown. *Commun. Stat. Simulat.*, 40(5):784–792, 2011. doi: 10.1080/03610918.2011.555042. URL http://www.tandfonline.com/doi/abs/10.1080/03610918.2011.555042.

Andrew T.A. Wood. Simulation of the von Mises Fisher distribution. *Commun. Stat. Simulat.*, 23(1):157–164, 1994. ISSN 0361-0918.