

SUBSAMPLING MCMC - AN INTRODUCTION FOR THE SURVEY STATISTICIAN

MATIAS QUIROZ, MATTIAS VILLANI, ROBERT KOHN
MINH-NGOC TRAN, AND KHUE-DUNG DANG

ABSTRACT. The rapid development of computing power and efficient Markov Chain Monte Carlo (MCMC) simulation algorithms have revolutionized Bayesian statistics, making it a highly practical inference method in applied work. However, MCMC algorithms tend to be computationally demanding, and are particularly slow for large datasets. Data subsampling has recently been suggested as a way to make MCMC methods scalable on massively large data, utilizing efficient sampling schemes and estimators from the survey sampling literature. These developments tend to be unknown by many survey statisticians who traditionally work with non-Bayesian methods, and rarely use MCMC. Our article explains the idea of data subsampling in MCMC by reviewing one strand of work, Subsampling MCMC, a so called pseudo-marginal MCMC approach to speeding up MCMC through data subsampling. The review is written for a survey statistician without previous knowledge of MCMC methods since our aim is to motivate survey sampling experts to contribute to the growing Subsampling MCMC literature.

AMS (2000) subject classification. Primary 62-02; Secondary 62D05.

Keywords and phrases. Pseudo-marginal MCMC, Difference estimator, Hamiltonian Monte Carlo (HMC)

1. INTRODUCTION

The key drivers behind the widespread adoption of Bayesian inference in the last three decades have been the rapid improvements in computing power and the availability of powerful user-friendly simulation algorithms. The family of Markov Chain Monte Carlo (MCMC) sampling methods (Brooks et al., 2011), and in particular the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970), quickly became the method of choice for practitioners for simulating from complex posterior distributions. MCMC opened up the possibility of routine analysis of highly complex models with limited algorithmic tuning. MCMC sampling was also fast enough for most problems, and at first it seemed that the problem of computational intractability that had hindered early Bayesians had been solved once and for all.

Meanwhile, it became apparent that MCMC was too slow in certain specialized areas where particular problems still had practitioners waiting for days or even weeks for MCMC to deliver the results. For example, MCMC is too slow for many high-dimensional spatial problems where the INLA approximations (Rue et al., 2009) quickly gained popularity. Massive datasets in technology led to fast Variational Bayes (VB) approximations (Jordan et al.,

Quiroz, Kohn and Dang: Australian School of Business, University of New South Wales. Villani: Department of Statistics, Stockholm University and Department of Computer and Information Science, Linköping University. Tran: Discipline of Business Analytics, University of Sydney.

1999; Blei et al., 2017) and Expectation Propagation (EP) (Minka, 2001) in the machine learning field. The tension with MCMC for big data problems in the machine learning community is now present in many other scientific disciplines in the natural and social sciences, and, with increasing text digitalization, also in the humanities. In the current big data era, MCMC is often too slow and is, as a result, increasingly being replaced by other approximate methods. This is unfortunate since, unlike other methods, MCMC samples are guaranteed to converge to the posterior distribution if the MCMC sampler performs adequately. Although there is exciting new work with flexible simulation based VB methods (see Blei et al., 2017 for a recent review), it is fair to say that VB is still less accurate than MCMC and does not come with practical error bounds. Moreover, it is often very time consuming to obtain good VB approximations for new complex models.

To deal with the challenges of massive datasets, there has been a recent push to develop scalable MCMC samplers. This work has followed two main paths: i) Distributed MCMC and ii) Subsampling MCMC. Distributed MCMC is inspired by the MapReduce scheme (Dean and Ghemawat, 2008) where the data is partitioned and distributed to different machines. MCMC is then run separately on each machine to obtain a subposterior for each partition in a parallel and distributed manner. The key question is then how to combine these subposteriors into a single posterior for all the data; see Scott et al. (2016), Neiswanger et al. (2013), Minsker et al. (2014), Wang and Dunson (2014) and Nemeth and Sherlock (2018) for some attempts. Subsampling MCMC instead focuses on taking random subsamples of the data in each MCMC iteration. The FireFly Monte Carlo algorithm in Maclaurin and Adams (2014) introduces an auxiliary variable for each observation which determines if it should be included in the evaluation of the posterior; Gibbs sampling (Geman and Geman, 1984) is then used to switch between updates of the parameters and the auxiliary variables. Korattikara et al. (2014) and Bardenet et al. (2014, 2017) use increasingly larger subsets of the data until the accept-reject decision in MCMC can be taken with sufficiently high confidence. We refer to Bardenet et al. (2017) for an excellent review of these and other subsampling approaches. After the publication of Bardenet et al. (2017), there has been interesting new progress on non-reversible MCMC for subsampling applications using continuous time piecewise deterministic Markov processes, see Bierkens et al. (2018) and Bouchard-Côté et al. (2018). Moreover, a different approach using Noisy MCMC (Alquier et al., 2016) and data subsampling is explored in Maire et al. (2018).

We will here focus on so called pseudo-marginal MCMC (PMCMC) methods where the likelihood evaluation is replaced by an unbiased estimate from a data subsample in each MCMC iteration (Andrieu and Roberts, 2009). Using a small subset to estimate the otherwise computationally costly likelihood in a big data setting can give dramatic speed-ups. As explained here, PMCMC has been shown to give samples from the correct posterior distribution even if the likelihood estimator is very noisy. However, as we demonstrate in this review, controlling the variability of the log of the likelihood estimator is absolutely crucial for the performance of Subsampling MCMC based on pseudo-marginal methods. This makes it important to introduce subsampling MCMC to survey sampling experts. The specific approach presented here has been developed in a series of papers (Quiroz et al., 2018a,b,c; Dang et al.,

2017) and should be of particular interest to survey statisticians since the estimation problem in our approach focuses on estimating the log-likelihood. The log-likelihood is usually a sum, and is therefore akin to a population total, the fundamental quantity in survey sampling. We also present a subsampling approach that directly estimates the likelihood unbiasedly (Quiroz et al., 2018c), which is usually a product; this is a less standard problem in survey sampling that may open up new challenges for survey statisticians. Finally, we note that estimating the log-likelihood based on data subsampling has also been explored in subsampling Sequential Monte Carlo (SMC) for static Bayesian models (Gunawan et al., 2018). SMC (Doucet et al., 2001) is a powerful alternative to MCMC which produces an estimate of the marginal likelihood, useful for model selection, as a byproduct. However, for brevity, this review focuses on MCMC.

The paper is organized as follows. The next section introduces the Metropolis-Hastings algorithm, and its extension to pseudo-marginal Metropolis-Hastings which can be used when the likelihood is replaced by an unbiased estimator. Section 3 gives details on estimators for the likelihood and their properties, and discusses several recently proposed variance reduction strategies such as using control variates and dependent subsamples. Section 3 also presents a promising approach for subsampling for Hamiltonian Monte Carlo (HMC) sampling which has recently been at the forefront in high-dimensional problems. The final section concludes. Appendix A summarizes the main algorithms and Appendix B gives some implementation details for our running illustrative example in the text.

2. THE PSEUDO-MARGINAL METROPOLIS-HASTINGS (PMMH) ALGORITHM

2.1. The Metropolis-Hastings algorithm. Markov Chain Monte Carlo (MCMC) is a family of algorithms for random variate generation from potentially complicated multivariate distributions. MCMC simulates from a distribution $\pi(\boldsymbol{\theta})$, here taken as a Bayesian posterior distribution, by constructing a Markov Chain on the parameter space of $\boldsymbol{\theta}$ such that its invariant distribution is $\pi(\boldsymbol{\theta})$. Realizations from this Markov chain will therefore converge in distribution to $\pi(\boldsymbol{\theta})$ from any starting point of the Markov chain, such that after a burn-in period the path of the Markov chain is a dependent sample from $\pi(\boldsymbol{\theta})$. The celebrated *Metropolis-Hastings (MH) algorithm* (Metropolis et al., 1953; Hastings, 1970) in Algorithm 1 in Appendix A, is the most widely used MCMC algorithm.

While the MH algorithm is valid for any proposal density $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the current value of the parameter and $\boldsymbol{\theta}'$ is its proposed value, the specific proposal used is crucial for the efficiency of the algorithm. The two most commonly used proposals are the Random Walk Metropolis (RWM) and the Independence sampler (IMH); see Brooks et al. (2011) for an introduction. The most common implementation of RWM uses a random walk proposal $q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = N(\boldsymbol{\theta}, \kappa^2\Omega)$, where Ω captures the shape of posterior in an efficient implementation (often Ω is minus the inverse posterior Hessian or simply the identity matrix) and κ is a tuning parameter. A small κ is often needed to keep the acceptance probability reasonably large, and the algorithm therefore tends to traverse the parameter space very slowly. This is especially pronounced in high dimensions as the optimal $\kappa^2 = O(1/d)$, where d is the number of parameters (Roberts et al., 1997). The IMH sampler generates proposals independent of

the current position: $q(\theta'|\theta) = q(\theta')$. Here it is crucial that $q(\theta')$ is a fairly accurate approximation to the true posterior and that it has heavier tails, otherwise the sampler will generate long sequences of rejected draws, i.e. the sampler gets stuck for long spells. When the IMH proposal is a good approximation of the posterior, the sampler traverses the parameter space very swiftly.

2.2. Estimating a computationally costly likelihood. The MH algorithm in Algorithm 1 is extremely convenient for Bayesian computations since it does not require knowledge of the normalizing constant of the posterior, $p(\mathbf{y}) = \int p(\mathbf{y}|\theta)p(\theta)d\theta$, which is often intractable. Even so, there are many problems where the required evaluations of the likelihood $p(\mathbf{y}|\theta)$ are also very costly, for example with large datasets or when the underlying probability model is a complex dynamical system, causing MH to be very slow. Moreover, for some models the likelihood can be intractable, e.g. in random effects models. Such situations are increasingly common in many of important applications and the slow execution of MH has prompted users to develop faster posterior approximation methods, for example variational Bayes (Blei et al., 2017) and expectation propagation (Gelman et al., 2017). While such methods are computationally attractive and steadily improving, they usually provide substantially less accurate approximations than MCMC.

A natural way to circumvent the problem of evaluating a costly likelihood $p(\mathbf{y}|\theta)$ is to replace the likelihood by a computationally cheap estimate, $\hat{p}(\mathbf{y}|\theta)$. We will here illustrate this idea in two very different settings.

Big data. Consider first the big data case when we run the Metropolis-Hastings algorithm on a dataset with n independent observations, with n very large. Evaluating the likelihood is generally an $O(n)$ operation and can be very costly. A natural solution is to estimate the likelihood from a subsample of size m obtained by simple random sampling. We first focus on estimating the *log-likelihood* instead of the likelihood; the reason for estimating on the log-scale is that the log-likelihood is usually a sum and therefore equivalent to estimating a population total, a long studied problem in survey sampling (Särndal et al., 2003). The log-likelihood for independent observations is

$$(2.1) \quad \ell(\mathbf{y}|\theta) \equiv \log p(y_1, \dots, y_n|\theta) = \sum_{i=1}^n \ell_i(y_i|\theta),$$

where $\ell_i(y_i|\theta) = \log p(y_i|\theta)$ is the *log-likelihood contribution* of the i th observation. Let u_1, \dots, u_n be binary variables such that $u_i = 1$ if observation y_i is selected in the subsample, and zero otherwise. Assuming simple random sampling (SRS) without replacement, the usual unbiased estimator is of the simple form

$$(2.2) \quad \hat{\ell}(\mathbf{y}|\theta) \equiv \frac{n}{m} \sum_{i=1}^n \ell_i(y_i|\theta)u_i.$$

While it is convenient from a survey sampling point of view to estimate the log-likelihood, we will see in Section 2.3 that Subsampling MCMC actually requires an unbiased estimate of the *likelihood* on the original scale. This entails estimating a product, which is a much less studied problem in survey sampling. In order to remain in the realm of survey sampling we

can use the unbiased estimator for the log-likelihood in (2.2) with a bias-correction to obtain an estimator for the likelihood of the form (Ceperley and Dewing, 1999; Nicholls et al., 2012)

$$(2.3) \quad \hat{p}(\mathbf{y}|\boldsymbol{\theta}) \equiv \exp\left(\hat{\ell}(\mathbf{y}|\boldsymbol{\theta}) - \sigma_{\hat{\ell}}^2(\boldsymbol{\theta})/2\right),$$

where $\sigma_{\hat{\ell}}^2(\boldsymbol{\theta}) \equiv \text{Var}(\hat{\ell}(\mathbf{y}|\boldsymbol{\theta}))$. This bias-correction is exact if i) $\hat{\ell}(\mathbf{y}|\boldsymbol{\theta})$ is normally distributed and ii) $\sigma_{\hat{\ell}}^2$ is known. In practice, $\sigma_{\hat{\ell}}^2$ is replaced by the usual sample estimate. We return to this issue in more detail in Section 3.4.

Note that the log-likelihood can often be written as a sum even when the observations are not fully independent. The most straightforward example is longitudinal data where the time series of observations within a subject are typically dependent temporally, but the different subjects are independent. In this case the log-likelihood is a sum over subjects and we can estimate it from a subsample of subjects, rather than individual observations. Data with a direct Markovian structure can be handled similarly by subsampling an observation jointly with its relevant history, as is done in the block bootstrap for time series.

Random effects and importance sampling. Another common setting where the likelihood is intractable, but can be estimated unbiasedly, are random effects models. As an example, consider a logistic regression with both fixed and random effects

$$p(y_{it}|\mathbf{x}_{it}, \mathbf{w}_{it}, \beta, \alpha_i, \Sigma_{\alpha}) = \frac{\exp(\mathbf{x}_{it}^T \beta + \mathbf{w}_{it}^T \alpha_i)^{y_{it}}}{1 + \exp(\mathbf{x}_{it}^T \beta + \mathbf{w}_{it}^T \alpha_i)},$$

where $\mathbf{y}_i = (y_{i1}, \dots, y_{in_i})^T$ are n_i observations for the i th subject, $\alpha_i \stackrel{iid}{\sim} N(0, \Sigma_{\alpha})$ are random effects of the covariates in \mathbf{w} , and \mathbf{x} are covariates with fixed effects. The likelihood for a sample of n observations with the random effects integrated out is then

$$(2.4) \quad p(\mathbf{y}_1, \dots, \mathbf{y}_n | \mathbf{X}, \mathbf{W}, \beta, \Sigma_{\alpha}) = \prod_{i=1}^n \int_{\alpha_i} p(\mathbf{y}_i | \mathbf{X}_i, \mathbf{W}_i, \beta, \alpha_i) p(\alpha_i | \Sigma_{\alpha}) d\alpha_i,$$

where

$$p(\mathbf{y}_i | \mathbf{X}_i, \mathbf{W}_i, \beta, \alpha_i, \Sigma_{\alpha}) = \prod_{t=1}^{n_i} p(y_{it} | \mathbf{x}_{it}, \mathbf{w}_{it}, \beta, \alpha_i).$$

The integrals in (2.4) are often intractable, but can be estimated unbiasedly by Monte Carlo integration, or importance sampling. Let m_i denote the number of samples in the importance sampling estimate of each term, and $m = \sum_{i=1}^n m_i$ the total number of random numbers used to estimate the likelihood in (2.4). Here importance sampling can be used to construct an unbiased estimate of the likelihood in random effects models. Similarly, for state space models, the particle filter gives an unbiased estimator of the likelihood using random particles, see Del Moral (2004, Proposition 7.4.1) for the original result and Pitt et al. (2012) for an alternative proof.

It is important to highlight the randomness of the estimator so we write $\hat{p}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u})$, where $\mathbf{u} \sim p(\mathbf{u})$ are the random numbers used to form the estimate. In the large data setting, \mathbf{u} is the vector of sample selection indicators discussed above and $p(\mathbf{u})$ is given by the simple random sampling design. More specifically, $\hat{p}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u})$ is given by (2.3) with the log-likelihood estimate in (2.2) showing the explicit dependence of the estimator on the random numbers u_i .

In random effects models the \mathbf{u} would instead be the random numbers used to approximate the intractable random effects integrals by Monte Carlo integration.

2.3. The Pseudo-Marginal Metropolis-Hastings algorithm. Andrieu and Roberts (2009) prove the remarkable result that replacing the likelihood $p(\mathbf{y}|\boldsymbol{\theta})$ in the MH algorithm with a noisy estimate $\hat{p}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u})$ still gives a sample from the posterior $\pi(\boldsymbol{\theta}) \propto p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ if the likelihood estimator \hat{p} is positive and unbiased. This is done by defining an augmented target density that includes both $\boldsymbol{\theta}$ and \mathbf{u} such that its marginal for $\boldsymbol{\theta}$ with \mathbf{u} integrated out is the posterior of $\boldsymbol{\theta}$. The MH algorithm is run on this augmented target distribution and the \mathbf{u} draws are not used for inference. It turns out that this so called *pseudo-marginal* algorithm is exactly of the same form as the original MH algorithm, with the likelihood evaluation in the acceptance probability in each iteration replaced by its current estimate; see the Pseudo-Marginal Metropolis-Hastings (PMMH) in Algorithm 2 for details. The idea of substituting the likelihood in MH with a noisy estimate appeared initially in physics (Lin et al., 2000) and in genetics (Beaumont, 2003).

Even though samples from PMMH with any unbiased positive likelihood estimator will converge to the posterior distribution, it turns out that having a low estimator variance is absolutely crucial for the efficiency of the standard PMMH sampler, see for example Flury and Shephard (2011) and Section 3.3. An estimator with a large variance can easily lead to an accepted parameter draw with a large over-estimate of the likelihood; subsequent draws will be rejected until they also happen to be associated with another gross over-estimate. This causes the sampler to be stuck for long spells, making the MCMC algorithm very inefficient.

The variance of the likelihood estimate is controlled by m , the number of subsamples in the subsampling setting, or the number of draws in importance sampling estimators. An m that is too small inflates the variance of the likelihood estimator and gives an inefficient sampler. An m that is too large gives an unnecessarily precise estimator at an excessive computational cost. The optimal m finds the right balance between MCMC efficiency and computational cost, and is usually derived under the assumption that the cost of a single MCMC iteration is proportional to $1/\text{Var}(\log \hat{p}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u}))$, see e.g. Pitt et al. (2012) for details. This cost must be balanced against the efficiency of the MCMC (which can be shown to increase with m , as we will illustrate later). The usual measure of MCMC sampling inefficiency for a given parameter θ is given by the *Integrated AutoCorrelation Time* (IACT)

$$(2.5) \quad \text{IACT} = 1 + 2 \sum_{k=0}^{\infty} \rho_k,$$

where ρ_k is the k th autocorrelation of the MCMC chain for θ . In practice, the IACT is estimated using the spectral density evaluated at zero, see for example Plummer et al. (2006). We define the Computational Time (CT) for producing a sample equivalent to an iid draw from the posterior distribution as

$$(2.6) \quad \text{CT}(\sigma_{\log \hat{p}}^2) \equiv \text{IACT}(\sigma_{\log \hat{p}}^2) \times \text{Time for a single MH iteration} \propto \frac{\text{IACT}(\sigma_{\log \hat{p}}^2)}{\sigma_{\log \hat{p}}^2},$$

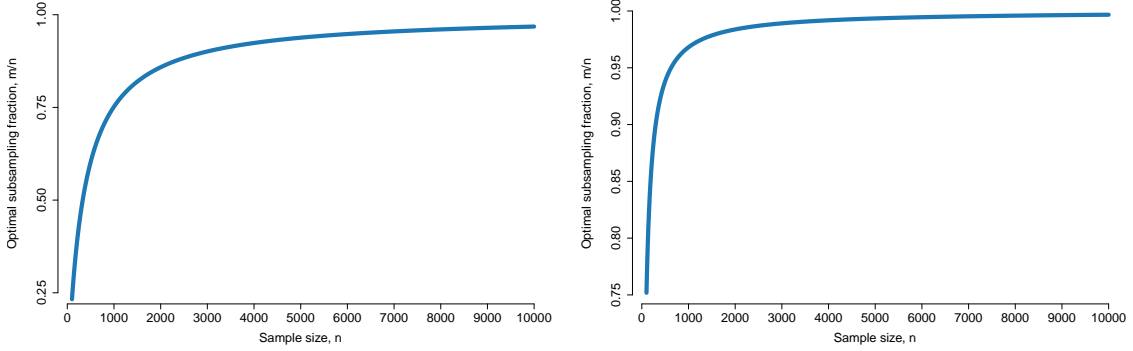


FIGURE 3.1. Optimal subsampling fractions (m/n) for SRS without replacement for $\sigma_{\ell_i}^2 = 1/100$ (left) and $\sigma_{\ell_i}^2 = 1/10$ (right), where $\sigma_{\ell_i}^2$ is the population variance in (3.1). The optimal subsample size (m) is set to target $\sigma_{\ell}^2 = 3.3$.

where $\sigma_{\log \hat{p}}^2 \equiv \text{Var}(\log \hat{p}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u}))$. We note that the IACT in (2.5) becomes a function of the variance of the log of the likelihood estimator when implementing pseudo-marginal MCMC. Here we follow Pitt et al. (2012) and Doucet et al. (2015) in assuming that the cost of a single iteration is proportional to m , which in turn is inversely proportional to $\sigma_{\log \hat{p}}^2$. Depending on the assumptions made, and the choice of proposal distribution for $\boldsymbol{\theta}$, the optimal subsample size m which minimizes CT is obtained by targeting a $\sigma_{\log \hat{p}}^2$ between 1 and 3.3 (Pitt et al., 2012; Doucet et al., 2015; Sherlock et al., 2015). It is also known that CT is relatively flat over the interval $\sigma_{\log \hat{p}}^2 \in [1, 3.3]$, but increases sharply outside this interval, in particular when $\sigma_{\log \hat{p}}^2$ is too large. We will illustrate some properties of the CT later in the text.

The definition of CT in (2.6) is the one traditionally used in pseudo-marginal MCMC. In some of the Subsampling MCMC methods the focus is on estimating the log-likelihood, which is subsequently converted into an estimator of the likelihood by bias-correction, see (2.3). The relevant Computational Time is then

$$(2.7) \quad \text{CT}(\sigma_{\ell}^2) \equiv \frac{\text{IACT}(\sigma_{\ell}^2)}{\sigma_{\ell}^2},$$

where $\sigma_{\ell}^2 \equiv \text{Var}(\hat{\ell}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u}))$. The two definitions of CT are identical if $\sigma_{\ell}^2(\boldsymbol{\theta})$ in (2.3) is known, and typically differ very little when $\sigma_{\ell}^2(\boldsymbol{\theta})$ in (2.3) is replaced by a sample estimate $\hat{\sigma}_{\ell}^2(\boldsymbol{\theta})$. To keep things simple, we will therefore use the same rule to set the subsample size to target $\sigma_{\ell}^2 \approx 1$ when using subsampling based on estimating the log-likelihood.

3. SUBSAMPLING FOR LIKELIHOOD ESTIMATION

The previous section described how an estimated likelihood can be used in a pseudo-marginal algorithm to sample from a posterior distribution. As long as the estimator is unbiased and nonnegative, and some non-onerous regularity conditions apply, the samples will converge in distribution to the target posterior based on the true likelihood function. This section discusses the importance of variance reduction and proposes alternative estimators from the survey literature and adapts them to the Subsampling MCMC context.

3.1. Simple Random Sampling is by itself not useful for Subsampling MCMC. We have already discussed that the optimal subsample size m should target a variance of the log-likelihood estimator in the interval $\sigma_{\hat{\ell}}^2 \in [1, 3.3]$. It turns out, however, that it is almost impossible in the subsampling setting to achieve a $\sigma_{\hat{\ell}}^2$ in that interval with Simple Random Sampling (SRS) without ending up with a sampling fraction m/n very close to unity. To see this, note that the variance of the estimator in (2.2) under the SRS design without replacement is (Särndal et al., 2003)

$$\sigma_{\hat{\ell}}^2 = \frac{n^2}{m} \left(1 - \frac{m}{n}\right) \sigma_{\ell_i}^2,$$

where $\sigma_{\ell_i}^2 \equiv \text{Var}(\ell_i) = n^{-1} \sum_{i=1}^n (\ell_i - \bar{\ell})^2$ is the population variance. Now, in order to target a given variance $\sigma_{\hat{\ell}}^2$, the subsample size must be

$$(3.1) \quad m = \frac{n^2 \sigma_{\ell_i}^2}{n \sigma_{\ell_i}^2 + \sigma_{\hat{\ell}}^2}.$$

Figure 3.1 illustrates the optimal sampling fraction as a function of n for two different values of σ_{ℓ_i} when the target is $\sigma_{\hat{\ell}}^2 = 3.3$. Note that this is the largest value $\sigma_{\hat{\ell}}^2$ among the recommended ones in the literature to keep the sampling fraction conservatively low here. The sampling fraction nevertheless quickly approaches unity, showing that SRS with the population total estimator in (2.2) is not useful for Subsampling MCMC. An even more dramatic way of illustrating this is to consider sampling with replacement. SRS with replacement gives $\sigma_{\hat{\ell}}^2 = n^2 \sigma_{\ell_i}^2 / m$ and the optimal m grows as $O(n^2)$, which is clearly unacceptable.

The variance of the estimator when sampling without replacement is lower by the factor $1 - m/n$ compared to the with-replacement case. This is a negligible improvement whenever $m \ll n$, which is the situation of interest here since otherwise subsampling would not be worthwhile. Since sampling with replacement is simpler to implement, and the implied independence makes the theory much easier to develop, this has been the preferred sampling method in the Subsampling MCMC literature. We will therefore use sampling with replacement throughout the paper. The sampling indicators $\mathbf{u} = (u_1, \dots, u_m)$ are now random observation indices such that $\Pr(u_k = i) = 1/n$ for $i = 1, \dots, n$ and the estimator in (2.2) becomes

$$(3.2) \quad \hat{\ell}(\mathbf{y}|\boldsymbol{\theta}) \equiv \frac{n}{m} \sum_{k=1}^m \ell_{u_k}(y_k|\boldsymbol{\theta}).$$

3.2. Efficient and scalable Subsampling MCMC using control variates.

The difference estimator. Part of the problem with SRS is that the log-likelihood contributions $\ell_i(y_i|\boldsymbol{\theta})$ can vary quite dramatically over the observations, hence inflating the variance of the estimator. There are at least three main ways to deal with the heterogeneity of population elements.

The first approach is stratified sampling with a higher sampling inclusion probability in the strata with largest units. This would ensure that most or all of the large $\ell_i(y_i|\boldsymbol{\theta})$ enter the sample. However, it turns out that stratified sampling tends to produce a variance that is too large for efficient Subsampling MCMC.

The second approach, proposed for Subsampling MCMC in the first version of Quiroz et al. (2018a) (see Quiroz et al., 2014 for the first version), is to use probability-proportional-to-size (PPS) sampling that assigns higher inclusion probabilities to larger units (Särndal et al., 2003). To implement PPS (or πPS in the case of sampling without replacement) we need to approximate the size of $\ell_i(\mathbf{y}_i|\boldsymbol{\theta})$ for all observations. In order to gain in computational speed from subsampling, those size measures must clearly be cheaper to compute than the $\ell_i(\mathbf{y}_i|\boldsymbol{\theta})$, and such size measures are proposed in (Quiroz et al., 2014). Nevertheless, the computational complexity of the subsampling algorithm remains $O(n)$.

The third approach is proposed in Quiroz et al. (2018a) and amounts to subtracting an approximation $q_i(\boldsymbol{\theta})$ from each $\ell_i(\mathbf{y}_i|\boldsymbol{\theta})$ such for each $\boldsymbol{\theta}$ that the new population $d_i(\boldsymbol{\theta}) = \ell_i(\mathbf{y}_i|\boldsymbol{\theta}) - q_i(\boldsymbol{\theta})$ is more homogeneous in size than $\ell_i(\mathbf{y}_i|\boldsymbol{\theta})$. Formally, we use simple random sampling and the *difference estimator* (Särndal et al., 2003)

$$(3.3) \quad \hat{\ell}_{DE}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u}) \equiv \sum_{i=1}^n q_i(\boldsymbol{\theta}) + \frac{n}{m} \sum_{k=1}^m d_{u_k}(\boldsymbol{\theta}),$$

with $q_i(\boldsymbol{\theta})$ is a potentially crude approximation to $\ell_i(\mathbf{y}_i|\boldsymbol{\theta})$ for $i = 1, \dots, n$. It is easy to show that $\hat{\ell}_{DE}(\mathbf{y}|\boldsymbol{\theta})$ is unbiased for any $q(\boldsymbol{\theta})$. The approximation $q(\boldsymbol{\theta})$ plays the same normalizing role as control variates in importance sampling (Hammersley and Handscomb, 1964) and we will use this term here.

Parameter-expanded control variates. A natural way of constructing control variates is by a Taylor expansion of $\ell(\mathbf{y}_i|\boldsymbol{\theta})$, $i = 1 \dots, n$, around some central value $\boldsymbol{\theta} = \boldsymbol{\theta}^*$ (Bardenet et al., 2017)

$$(3.4) \quad \ell(\mathbf{y}_i|\boldsymbol{\theta}) \approx \ell(\mathbf{y}_i|\boldsymbol{\theta}^*) + (\boldsymbol{\theta} - \boldsymbol{\theta}^*)^T \nabla_{\boldsymbol{\theta}} \ell(\mathbf{y}_i|\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} + \frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\theta}^*)^T \nabla_{\boldsymbol{\theta}\boldsymbol{\theta}^T}^2 \ell(\mathbf{y}_i|\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} (\boldsymbol{\theta} - \boldsymbol{\theta}^*),$$

where $\nabla_{\boldsymbol{\theta}} \ell(\mathbf{y}_i|\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}$ and $\nabla_{\boldsymbol{\theta}\boldsymbol{\theta}^T}^2 \ell(\mathbf{y}_i|\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}$ are the gradient and Hessian with respect to $\boldsymbol{\theta}$ evaluated at $\boldsymbol{\theta} = \boldsymbol{\theta}^*$, respectively. As argued in Bardenet et al. (2017), these *parameter-expanded* control variates work very well when the posterior is tightly concentrated; asymptotic posterior concentration is guaranteed by the Bernstein von Mises theorem (Van der Vaart, 1998) and will be practically relevant in big data problems with many observations, but not too many parameters, i.e. so called *tall data*. As discussed later, it also has good scaling properties with respect to n .

A crucial property of parameter-expanded covariates is that the sum $\sum_{i=1}^n q_i(\boldsymbol{\theta})$ in the difference estimator in (3.3) can be reduced from an $O(n)$ operation to an $O(1)$ operation since both $\sum_{i=1}^n \nabla_{\boldsymbol{\theta}} \ell(\mathbf{y}_i|\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}$ and $\sum_{i=1}^n \nabla_{\boldsymbol{\theta}\boldsymbol{\theta}^T}^2 \ell(\mathbf{y}_i|\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}$ are evaluated at $\boldsymbol{\theta}^*$, and can therefore be pre-computed before starting the MCMC iterations.

Data-expanded control variates. Let \mathbf{z}_i be a vector with all observed data for the i th item. For example, in a regression setting, $\mathbf{z}_i = (y_i, \mathbf{x}_i^T)^T$ would contain both the response variable y and the covariates \mathbf{x} . Further, let $\ell(\mathbf{z}_i|\boldsymbol{\theta})$ denote log-likelihood contribution for the i th observation. The idea with the data-expanded control variates proposed in Quiroz et al. (2018a) is that the $\ell(\mathbf{z}_i|\boldsymbol{\theta})$ tend to vary slowly across data space, and $\ell(\mathbf{z}_i|\boldsymbol{\theta})$ can therefore be approximated by $\ell(\mathbf{z}_{c_i}|\boldsymbol{\theta})$, where \mathbf{z}_{c_i} is the nearest centroid in a pre-clustering of the data.

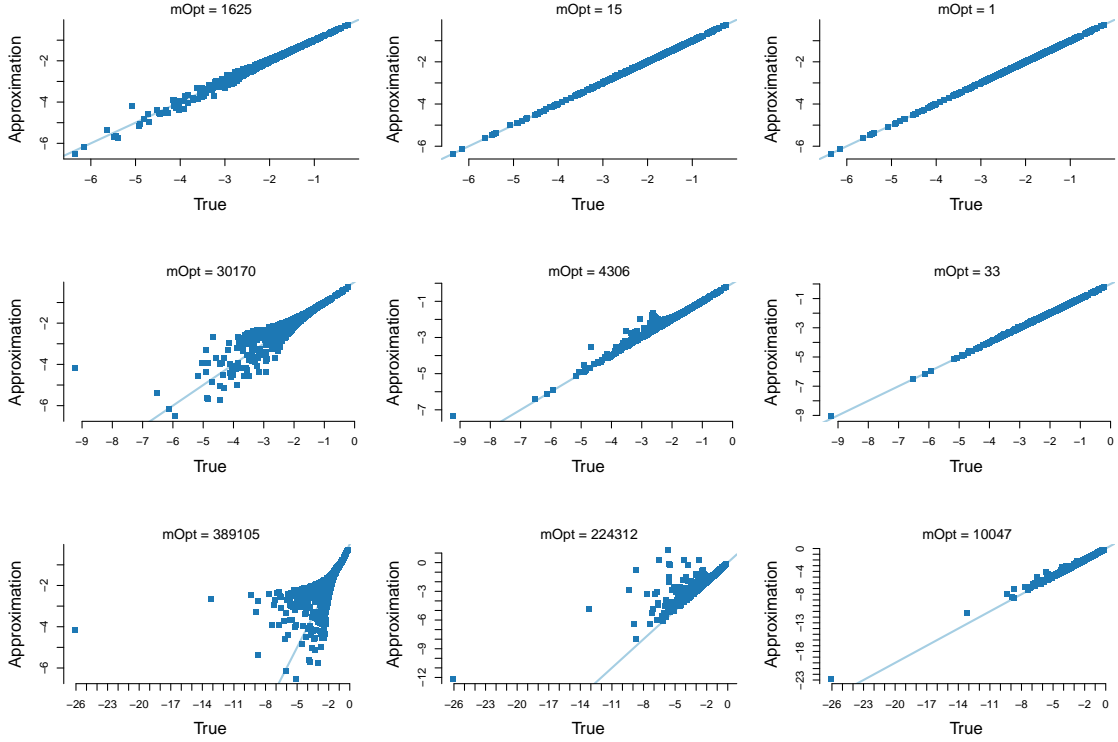


FIGURE 3.2. The accuracy of the parameter expanded control variates for the Poisson regression model in Eq. (3.6). Each subgraph plots the true ℓ_i against the control variate for that observation. The three columns correspond to 0, 1 and 2 terms in the Taylor expansion. The three rows correspond to different θ that are increasingly distant from the Taylor expansion point θ^* : i) $\|\theta - \theta^*\|_2 = 0.025$ (top row), ii) $\|\theta - \theta^*\|_2 = 0.1$ (middle row) and iii) $\|\theta - \theta^*\|_2 = 0.25$ (bottom row), where $\|\cdot\|_2$ is the Euclidean norm. As a point of comparison, θ 's on the 50% posterior ellipsoid have values for $\|\theta - \theta^*\|_2$ ranging between 0.013 and 0.028. The header of each subgraph displays the optimal subsample (m_{opt}) that gives the target variance $\text{Var}(\hat{\ell}_{DE}) = 3.3$. The quality of the parameter expanded control variates depends on $\theta - \theta^*$ being small.

Similarly to the parameter-expanded control variates, we can improve on this by using a Taylor expansion of $\ell(\mathbf{z}_i|\theta)$, but this time in data space around the centroid \mathbf{z}_{c_i} . The *data-expanded* control variates are of the form

$$(3.5) \quad \ell(\mathbf{z}_i|\theta) \approx \ell(\mathbf{z}_{c_i}|\theta) + (\mathbf{z}_i - \mathbf{z}_{c_i})^T \nabla_{\mathbf{z}} \ell(\mathbf{z}|\theta)|_{\mathbf{z}=\mathbf{z}_{c_i}} + \frac{1}{2} (\mathbf{z}_i - \mathbf{z}_{c_i})^T \nabla_{\mathbf{z}\mathbf{z}^T}^2 \ell(\mathbf{z}|\theta)|_{\mathbf{z}=\mathbf{z}_{c_i}} (\mathbf{z}_i - \mathbf{z}_{c_i}),$$

where $\nabla_{\mathbf{z}} \ell(\mathbf{z}|\theta)|_{\mathbf{z}=\mathbf{z}_{c_i}}$ and $\nabla_{\mathbf{z}\mathbf{z}^T}^2 \ell(\mathbf{z}|\theta)|_{\mathbf{z}=\mathbf{z}_{c_i}}$ are the gradient and Hessian with respect to \mathbf{z} , both evaluated at $\mathbf{z} = \mathbf{z}_{c_i}$.

Quiroz et al. (2018a) show that the complexity of $\sum_{i=1}^n q_i(\theta)$ is $O(K)$ for data-expanded control variates, where K is the number of clusters and typically $K \ll n$. Hence, data-expanded control variates also give scalable algorithms since the number of clusters tends to grow very slowly with n .

Comparing control variates from parameter-expansion and data-expansion. It is crucial to realize that our sampling problem is dynamic, in the sense that we will need estimates of $\hat{\ell}(\mathbf{y}|\boldsymbol{\theta})$ at every iteration of the pseudo-marginal MH algorithm, and $\boldsymbol{\theta}$ typically changes in every iteration. This means that we have sequence of survey sampling problems where the measurements on the population units, $\ell(\mathbf{y}_i|\boldsymbol{\theta}), i = 1, \dots, n$, change over time (MH iterations). Such situations also occur in real-world surveys (Steel and McLaren, 2009), but Subsampling MCMC has not as yet used any of the methods proposed in the repeated surveys literature. We return to this dynamic survey sampling perspective when we discuss dependent subsampling in Section 3.6. The fact that $\boldsymbol{\theta}$ changes over the iterations can cause problems for the parameter-expanded control variates, but does not significantly affect the data-expanded control variates. This is illustrated in Figures 3.2 and 3.3 where we plot the true $\ell(\mathbf{y}_i|\boldsymbol{\theta})$ against the two control variates for different number of terms in the Taylor expansions. For illustration purposes the underlying sample of $n = 1000$ observations comes from a simple Poisson regression

$$(3.6) \quad y_i|x_i \sim \text{Pois}(\exp(\theta_0 + \theta_1 x_i)),$$

where $\boldsymbol{\theta} = (\theta_0, \theta_1) = (1, 0.75)$, but the point we make holds generally. Figure 3.2 clearly shows that parameter-expansion around a static $\boldsymbol{\theta}^*$ is problematic when the current $\boldsymbol{\theta}$ is far from $\boldsymbol{\theta}^*$. Figure 3.3 shows that the data-expanded control variates remains relatively unaffected by movements in $\boldsymbol{\theta}$.

However, data-expanded control variates only give accurate approximations if enough centroids are used in the clustering; see Figures 3.4 and 3.5 for an illustration. The curse of dimensionality makes this a limitation in higher dimensional data spaces since many observations will be quite far from their nearest centroid even when using a larger number of centroids.

In summary, data-expanded control variates perform well for any $\boldsymbol{\theta}$, but do not scale well with the dimension of the data space. Parameter-expanded control variates scale well with dimension, but perform poorly when $\boldsymbol{\theta}$ is far from the expansion point $\boldsymbol{\theta}^*$. Quiroz et al. (2018a) therefore propose the strategy of starting the posterior sampling with data-expanded control variates and then switching over to parameter-expanded control variates when the sampler has reached a more central point in the posterior which can be used as $\boldsymbol{\theta}^*$.

Asymptotic behavior with control variates. We have shown that the optimal subsample size needs to grow as $O(n^2)$ when using simple random sampling with replacement in order to keep $\text{Var}(\hat{\ell}(\mathbf{y}|\boldsymbol{\theta}))$ around unity; control variates can improve on this asymptotic rate. With control variates, the variance of the difference estimator in (3.3) is given by

$$(3.7) \quad \text{Var}(\hat{\ell}_{DE}) = \frac{n^2 \sigma_d^2(n)}{m},$$

where $\sigma_d^2(n) \equiv (1/n) \sum_{i=1}^n (d_i - \bar{d})^2$ is the variance of the finite population of differences. Note that we have made explicit that the accuracy of the control variates depends on n . As explained above, to obtain the optimal m we need to ensure that $\text{Var}(\hat{\ell}_{DE})$ is $O(1)$, which requires understanding the behaviour of $\sigma_d^2(n)$ as $n \rightarrow \infty$. Lemma 2 in Quiroz et al. (2018a)

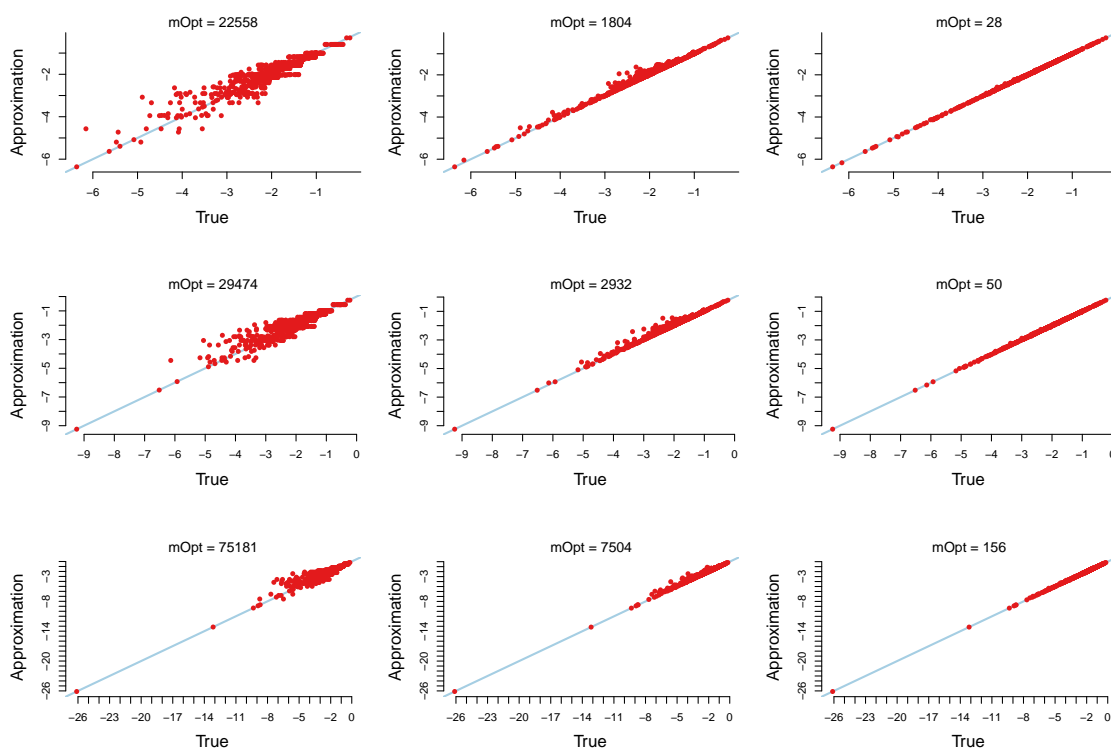


FIGURE 3.3. The accuracy of the data expanded control variates with 75 centroids for the Poisson regression model in Eq. (3.6). Each subgraph plots the true ℓ_i against the control variate for that observation. The three columns correspond to 0, 1 and 2 terms in the Taylor expansion. The three rows correspond to different θ that are increasingly distant from the Taylor expansion point θ^* : i) $\|\theta - \theta^*\|_2 = 0.025$ (top row), ii) $\|\theta - \theta^*\|_2 = 0.1$ (middle row) and iii) $\|\theta - \theta^*\|_2 = 0.25$ (bottom row), where $\|\cdot\|_2$ is the Euclidean norm. As a point of comparison, θ 's on the 50% posterior ellipsoid have values for $\|\theta - \theta^*\|$ ranging between 0.013 and 0.028. The header of each subgraph displays the optimal subsample (m_{opt}) that gives the target variance $\text{Var}(\hat{\ell}_{DE}) = 3.3$. The quality of the data expanded control variates is not sensitive to $\theta - \theta^*$.

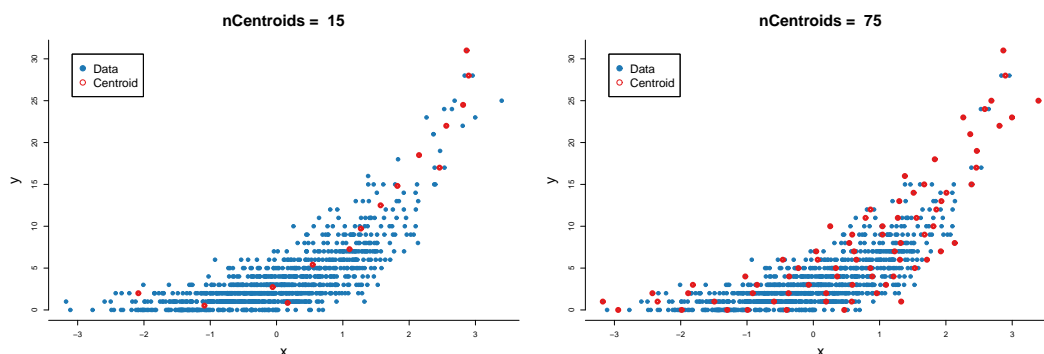


FIGURE 3.4. The data points in (x,y)-space and the centroids.

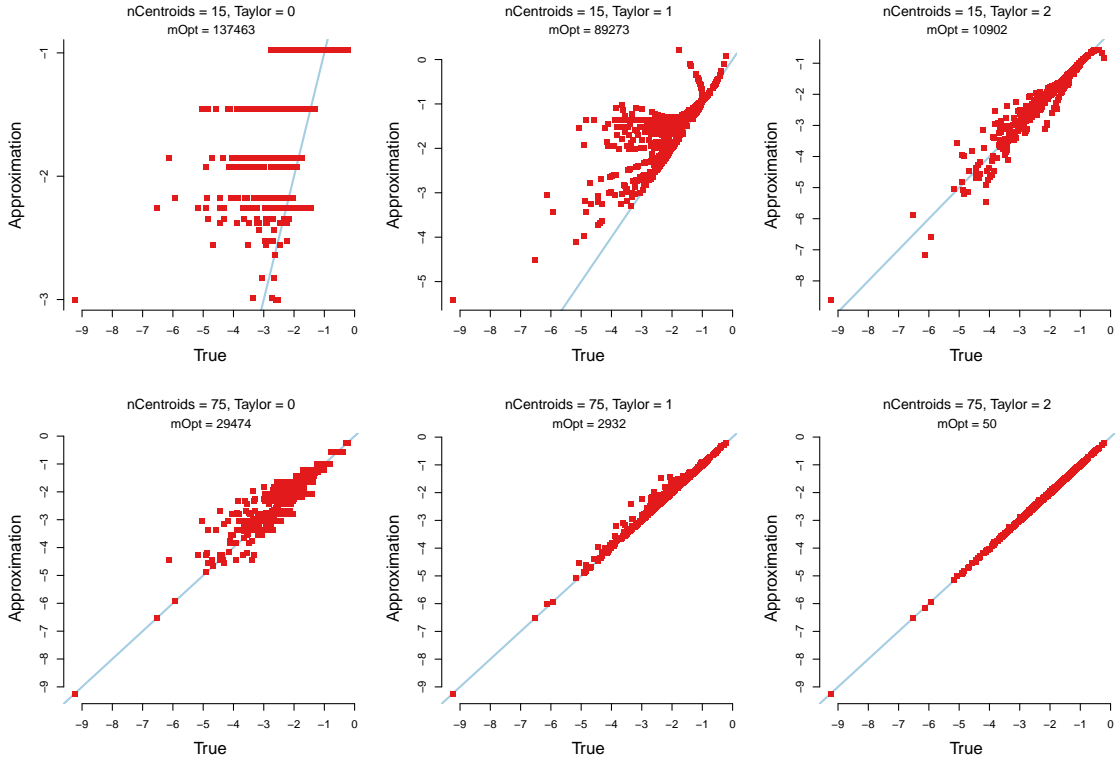


FIGURE 3.5. The accuracy of the data expanded control variates with different number of centroids for the Poisson regression model in Eq. (3.6) when θ is such that $\|\theta - \theta^*\|_2 = 0.1$, where $\|\cdot\|_2$ is the Euclidean distance. The three columns correspond to 0, 1 and 2 terms in the Taylor expansion. The header of each subgraph displays the optimal subsample (m_{opt}) that gives the target variance $\text{Var}(\hat{\ell}_{DE}) = 3.3$ and the order of the Taylor expansion (Taylor). The quality of the data expanded control variates deteriorates when the number of centroids is small, especially when using a lower order Taylor expansion.

shows that

$$(3.8) \quad \text{Var}(\hat{\ell}_{DE}) = \frac{n^2 O(a_n^2)}{m},$$

where

$$a_n(\theta) \equiv 2 \max_{i \in \{1, \dots, n\}} |d_i(\theta)|.$$

The asymptotic behaviour of $a_n(\theta)$ depends on the type of control variate, and also on choices within a given control variate such as how the number of centroids grows with n in the case of data-expanded control variates. We will focus here on the asymptotic properties of parameter-expanded control variates and refer to Quiroz et al. (2018a) for results on the data-expanded case.

Since the parameter-expanded control variate is based on a Taylor expansion around θ^* , the rate at which its accuracy improves with n is determined by the rate at which $\|\theta - \theta_n^*\|_2$ contracts, where we have made explicit that the expansion point θ_n^* typically depends on n . Quiroz et al. (2018a) prove the following lemma.

Lemma 3.1. *For the parameter-expanded control variates of second order we have*

$$a_n(\boldsymbol{\theta}) = \|\boldsymbol{\theta} - \boldsymbol{\theta}_n^*\|_2^3 \cdot O(1).$$

From the Bernstein-von Mises theorem (Chen, 1985), if $\boldsymbol{\theta}_n^*$ is the posterior mode based on all data, we have $\sqrt{n}(\boldsymbol{\theta} - \boldsymbol{\theta}_n^*) \xrightarrow{d} N(0, \tau^2)$ as $n \rightarrow \infty$. This implies that $\Pr(\|\boldsymbol{\theta} - \boldsymbol{\theta}_n^*\| \leq K\tau/\sqrt{n})$ will be close to unity for large enough K . We therefore have that $a_n(\boldsymbol{\theta}) = O(n^{-3/2})$ for all $\boldsymbol{\theta} \in \{\boldsymbol{\theta} : \|\boldsymbol{\theta} - \boldsymbol{\theta}_n^*\| \leq K\tau/\sqrt{n}\}$. Hence, for such $\boldsymbol{\theta}$, the optimal subsample size that targets $\text{Var}(\hat{\ell}_{DE}) = O(1)$ is, by (3.8), $m = O(n^{-1})$, suggesting that Subsampling MCMC with parameter-expanded control variates scales extremely well to large datasets. There are at least three objections to this analysis, however. First, the conditions under which this optimality is derived requires that m is large enough for $\hat{\ell}$ to be approximately normally distributed, so the optimal $m = O(n^{-1})$ is not attainable. Second, having control variates that expand around the posterior mode of $\boldsymbol{\theta}$ based on all data is not practical in large data settings. Third, as discussed in Quiroz et al. (2018a), setting $m = O(n^{-1})$ gives a PMMH algorithm that samples from a target distribution that deviates from the true posterior by an $O(n)$ factor, which is clearly not acceptable. A more practical approach with control variates based on the posterior mode from a small subset of the data is analyzed in Quiroz et al. (2018a) and presented in Section 3.4 below.

Other control variates. We have emphasized parameter- and data-expanded control variates as general and scalable solutions for variance reduction in Subsampling MCMC. However, many other control variates can be used in particular applications. For example, in many models the evaluation of the log-likelihood contributions $\ell(y_i|\boldsymbol{\theta})$ is very time-consuming because some aspect of the model needs to be solved numerically. The likelihood can then be costly also for smaller n . For example, an intractable integral may be approximated by Gaussian quadrature, a differential equation can be solved by the Runge-Kutta method, an optimum found by Newton’s method. Any numerical method depends on tuning parameters which control the accuracy of the solution. A natural control variate can then be obtained from tuning parameters that give cruder, but much faster, evaluations of $\ell(y_i|\boldsymbol{\theta})$ (a coarse grid in numerical integration and in solving differential equations, a small number of Newton steps for optimization). The log-likelihood contributions for the sampled subset of observations are computed based on tuning parameters that give very accurate evaluations. Note however that for such control variates we need in general to evaluate the control variate for all n observations (but n may be small), so the algorithm will still run in $O(n)$ time, but with a much smaller cost for each MCMC iteration.

3.3. Control variates are crucial for the Integrated AutoCorrelation Time (IACT). We have argued that control variates provide significant variance reduction for the log-likelihood estimator and that the MCMC sampling efficiency (as measured by the IACT) is a function of the variance. Figure 3.6 illustrates that when targeting a variance of one for $\hat{\ell}(\boldsymbol{\theta})$ (second row) our subsampling MCMC essentially behaves as the full data MCMC (first row). However, Subsampling MCMC with a large estimator variance ($\text{Var}(\hat{\ell}(\boldsymbol{\theta})) = 10$ in the third row and $\text{Var}(\hat{\ell}(\boldsymbol{\theta})) = 50$ in the fourth row) does not efficiently explore the posterior distribution

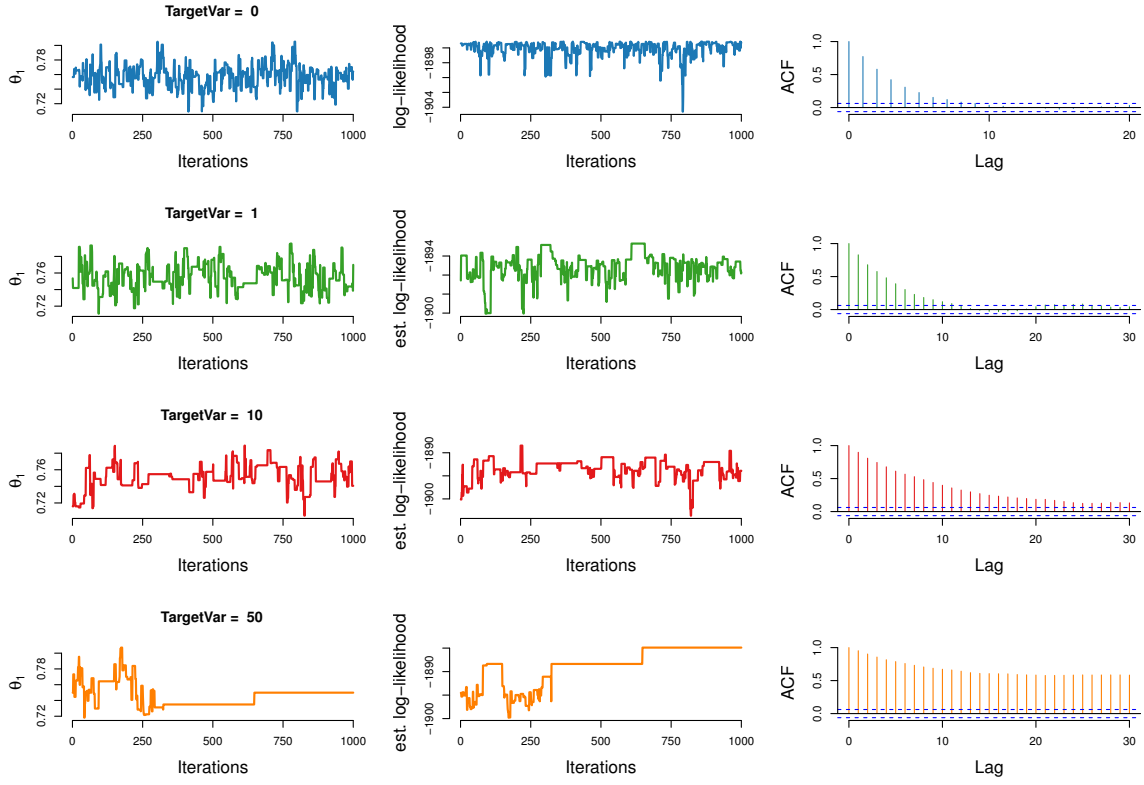


FIGURE 3.6. Sampling the posterior distribution of θ_1 in the Poisson regression model in (3.6). The figure shows, for different values of $\sigma_\ell^2 \in \{0, 1, 10, 50\}$ on the four rows, the sampling chains (left), the estimates of the log-likelihood (middle) and estimates of the chain's autocorrelation ρ_k (right).

of our Poisson regression example, and has a much greater tendency of getting stuck. This stickiness is also clearly borne out in the autocorrelation function of the MCMC draws in the right panel of Figure 3.6.

3.4. An approximate approach using bias-corrected log-likelihood estimators. We have so far shown the importance of variance reduction of the log-likelihood estimator using control variates. The reason for focusing on estimators of the log-likelihood, rather than the likelihood, is that the log-likelihood is a sum, which is the usual aim in survey sampling, allowing us to exploit century-old experience in that area.

However, pseudo-marginal MCMC will only generate a sample from the correct posterior if the *likelihood* is estimated by a positive unbiased estimator (Andrieu and Roberts, 2009). The difference estimator in (3.3) is unbiased for the log-likelihood, but biased for the likelihood. As discussed in Section 2.2, we can bias-correct the biased estimator $\exp(\hat{\ell}(\mathbf{y}|\boldsymbol{\theta}))$, where $\hat{\ell}(\mathbf{y}|\boldsymbol{\theta})$ is any unbiased estimator of the log-likelihood. In particular, using the difference estimator the bias-corrected estimator is of the form

$$(3.9) \quad \exp\left(\hat{\ell}_{DE}(\mathbf{y}|\boldsymbol{\theta}) - \sigma_{\hat{\ell}_{DE}}^2(\boldsymbol{\theta})/2\right),$$

where $\hat{\ell}_{DE}(\mathbf{y}|\boldsymbol{\theta})$ is the difference estimator in (3.3) and $\sigma_{\hat{\ell}_{DE}}^2 = \text{Var}(\hat{\ell}_{DE}(\mathbf{y}|\boldsymbol{\theta}))$. The estimator in (3.9) is unbiased if i) $\hat{\ell}_{DE}(\mathbf{y}|\boldsymbol{\theta})$ is normally distributed and ii) $\sigma_{\hat{\ell}_{DE}}^2$ is known. The assumption

of normality can often be defended by a central limit theorem in the large m setting (assuming also that n grows). Even when m is very small we have observed that $\hat{\ell}_{DE}(\mathbf{y}|\boldsymbol{\theta})$ is very close to normal since the control variates homogenize the population so that $d_i(\boldsymbol{\theta}) = \ell(y_i|\boldsymbol{\theta}) - q_i(\boldsymbol{\theta}), i \in \{1, \dots, n\}$, are usually distributed much more symmetrically and have lighter tails than the population of $\{\ell(y_i|\boldsymbol{\theta})\}_{i=1}^n$. Assuming that $\sigma_{\hat{\ell}_{DE}}^2$ is known is harder to defend since knowing $\sigma_{\hat{\ell}_{DE}}^2$ requires computing $d_i(\boldsymbol{\theta})$ for all n observations. The approach in Quiroz et al. (2018a) replaces $\sigma_{\hat{\ell}_{DE}}^2$ in (3.9) by

$$(3.10) \quad \hat{\sigma}_{\hat{\ell}_{DE}}^2(\boldsymbol{\theta}) \equiv \frac{1}{m} \sum_{k=1}^m (d_{u_k}(\boldsymbol{\theta}) - \bar{d}^{(m)}(\boldsymbol{\theta}))^2,$$

where $\bar{d}^{(m)}(\boldsymbol{\theta}) = m^{-1} \sum_{k=1}^m d_{u_k}(\boldsymbol{\theta})$, giving the estimator

$$(3.11) \quad \hat{p}_{DE}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u}) \equiv \exp\left(\hat{\ell}_{DE}(\mathbf{y}|\boldsymbol{\theta}) - \hat{\sigma}_{\hat{\ell}_{DE}}^2(\boldsymbol{\theta})/2\right).$$

Substituting an estimate $\hat{\sigma}_{\hat{\ell}_{DE}}^2(\boldsymbol{\theta})$ makes the estimator in (3.11) only approximately unbiased, and raises the question: what do samples from a PMMH algorithm using the estimator $\hat{p}_{DE}(\mathbf{y}|\boldsymbol{\theta})$ converge to, if anything? Quiroz et al. (2018a) note that this PMMH is still a valid MCMC on the joint $(\boldsymbol{\theta}, \mathbf{u})$ space, but targets the density

$$(3.12) \quad \bar{\pi}(\boldsymbol{\theta}, \mathbf{u}) = \frac{\hat{p}_{DE}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u})p(\mathbf{u})p(\boldsymbol{\theta})}{\bar{p}(\mathbf{y})}, \text{ where } \bar{p}(\mathbf{y}) = \iint_{\mathbf{u}, \boldsymbol{\theta}} \hat{p}_{DE}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u})p(\mathbf{u})p(\boldsymbol{\theta})d\mathbf{u}d\boldsymbol{\theta}.$$

The marginal density of $\boldsymbol{\theta}$ is

$$(3.13) \quad \bar{\pi}(\boldsymbol{\theta}) = \frac{\bar{p}(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\bar{p}(\mathbf{y})}, \text{ where } \bar{p}(\mathbf{y}|\boldsymbol{\theta}) \equiv \int_{\mathbf{u}} \hat{p}_{DE}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u})p(\mathbf{u})d\mathbf{u}.$$

Note that $\bar{p}(\mathbf{y}|\boldsymbol{\theta}) \neq p(\mathbf{y}|\boldsymbol{\theta})$ in general because of the (slight) bias in the likelihood estimator $\hat{p}_{DE}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u})$. This shows that PMMH based on $\hat{p}_{DE}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u})$ is still a valid MCMC scheme, but the draws $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(N)}$ target the perturbed posterior $\bar{\pi}(\boldsymbol{\theta})$ instead of the actual posterior $\pi(\boldsymbol{\theta})$.

Our next result from Quiroz et al. (2018a) gives the rate at which the perturbed target $\bar{\pi}(\boldsymbol{\theta})$ approaches the true target posterior $\pi(\boldsymbol{\theta})$. Note that $\pi(\boldsymbol{\theta})$ depends on n and $\bar{\pi}(\boldsymbol{\theta})$ depends on both n and m . We make this dependence explicit by using the relevant subscripts in our asymptotic results.

Theorem 3.1. *Suppose that a PMMH algorithm is implemented with the estimator $\hat{p}_{DE}(\mathbf{y}|\boldsymbol{\theta})$ in (3.11) using the second order parameter expanded control variates where the expansion point $\boldsymbol{\theta}^*$ is the posterior mode, and assume that the regularity conditions in Assumption 2 in Quiroz et al. (2018a) are satisfied. Then,*

i)

$$\int_{\Theta} |\bar{\pi}_{m,n}(\boldsymbol{\theta}) - \pi_n(\boldsymbol{\theta})| d\boldsymbol{\theta} = O\left(\frac{1}{nm^2}\right).$$

ii) *Suppose that $h(\boldsymbol{\theta})$ is a function such that $E_{\pi_n}[h^2(\boldsymbol{\theta})] < \infty$. Then*

$$|E_{\bar{\pi}_{m,n}}[h(\boldsymbol{\theta})] - E_{\pi_n}[h(\boldsymbol{\theta})]| = O\left(\frac{1}{nm^2}\right).$$

Theorem 3.1 shows that the perturbation error vanishes rapidly with the subsample size at rate $O(m^{-2})$ for fixed n . The theorem also shows that when for example $m = O(n^{1/2})$, the perturbation error is $O(n^{-2})$.

To analyze the scalability of the algorithm for practical work, Quiroz et al. (2018a) make the more realistic assumption that control variates are expanded around $\theta_{\tilde{n}}^*$, the posterior mode based on a small subset of \tilde{n} observations, rather the costly posterior mode θ_n^* based on all n observations. The following corollary is proved in Quiroz et al. (2018a).

Corollary 3.1. *Suppose that a PMMH algorithm is implemented with the estimator $\hat{p}_{DE}(\mathbf{y}|\theta)$ in (3.11) using the second order parameter expanded control variates with expansion point $\theta_{\tilde{n}}^*$ based on a subset $\tilde{n} \ll n$ of observations. Assume that $\theta_{\tilde{n}}^* - \theta_n^* = O(\tilde{n}^{-1/2})$, and that the regularity conditions in Assumption 2 in Quiroz et al. (2018a) are satisfied. Then,*

i)

$$\int_{\Theta} |\bar{\pi}_{m,n}(\theta) - \pi_n(\theta)| d\theta = O\left(\frac{n}{m^2\tilde{n}^3}\right).$$

ii) *Suppose that $h(\theta)$ is a function such that $E_{\pi_n}[h^2(\theta)] < \infty$. Then*

$$|E_{\bar{\pi}_{m,n}}[h(\theta)] - E_{\pi_n}[h(\theta)]| = O\left(\frac{n}{m^2\tilde{n}^3}\right).$$

If $\tilde{n} = n^\kappa$ for some κ , then $m = O(n^{2-3\kappa})$ achieves the optimal variance of $O(1)$, and the perturbation errors in Corollary 3.1 decreases with n if and only if $\kappa < 2/3$. For example, if we take $\kappa = 1/2$, then $m = O(n^{1/2})$ and the posterior perturbation error is $O(n^{-1/2})$.

The asymptotics in Theorem 3.1 and Corollary 3.1 are reassuring for the method, but does not provide a practically useful way to quantify the discrepancy between $\bar{\pi}_{m,n}(\theta)$ and $\pi_n(\theta)$. Quiroz et al. (2018a) derive an accurate approximation to the point-wise fractional error in the perturbed posterior distribution

$$(3.14) \quad \text{error}(\theta) = \frac{\bar{\pi}_{m,n}(\theta) - \pi_n(\theta)}{\pi_n(\theta)}.$$

and show that the error(θ) increases with $\sigma_{\hat{\ell}_{DE}}^2(\theta)$ for large $\sigma_{\hat{\ell}_{DE}}^2(\theta)$. It is important to note however that it is only the part of $\sigma_{\hat{\ell}_{DE}}^2(\theta)$ that depends on θ that affects the perturbation error; an additive constant to $\sigma_{\hat{\ell}_{DE}}^2(\theta)$ will give rise to a multiplicative constant to $\hat{p}_{DE}(\mathbf{y}|\theta)$ in (3.11) that also appears in $\bar{p}(\mathbf{y})$ and will therefore cancel in (3.13). Hence, a large $\sigma_{\hat{\ell}_{DE}}^2(\theta)$ only implies a large perturbation error if $\sigma_{\hat{\ell}_{DE}}^2(\theta)$ varies with θ . This can be an advantage for data-expanded control variates since their errors are by construction relatively insensitive to θ , as demonstrated in Figure 3.3.

The next subsection presents an alternative approach which produces an unbiased estimator of the likelihood. Although exact, this method has two drawbacks compared to the approximate method presented in this subsection. First, the relative computational time of the algorithm is higher than the approximate method above, see Figure S8 in the supplementary material of Quiroz et al. (2018c). Second, the exact approach can only estimate expectations of functions of the parameters, rather than the whole posterior distribution.

3.5. Signed PMMH with the Block-Poisson estimator. The approach in the previous subsection used an unbiased estimator of the log-likelihood, which was subsequently approximately bias-corrected to estimate the likelihood

$$(3.15) \quad p(\mathbf{y}|\boldsymbol{\theta}) = \prod_{i=1}^n p(y_i|\boldsymbol{\theta}).$$

We now review how to estimate this product unbiasedly using the Block-Poisson estimator proposed in Quiroz et al. (2018c).

The *Block-Poisson* estimator is defined as

$$(3.16) \quad \hat{p}_B(\mathbf{y}|\boldsymbol{\theta}) = Q(\boldsymbol{\theta}) \prod_{l=1}^{\lambda} \zeta_l, \quad \zeta_l = \exp\left(\frac{a + \lambda}{\lambda}\right) \prod_{h=1}^{\mathcal{X}_l} \left(\frac{\hat{d}_{m_b}^{(h,l)} - a}{\lambda}\right),$$

where $Q(\boldsymbol{\theta}) = \exp(\sum_{i=1}^n q_i(\boldsymbol{\theta}))$, with $q_i(\boldsymbol{\theta})$ being the control variates in (3.3). The Block-Poisson estimator is essentially a product of $\lambda \in \mathbb{N}^+$ Poisson estimators, $\zeta_l, l = 1, \dots, \lambda$ (Wagner, 1988; Papaspiliopoulos, 2009). Each Poisson estimator in the product is based on a random number $\mathcal{X}_l \stackrel{\text{indep.}}{\sim} \text{Pois}(1)$ of unbiased estimates $\hat{d}_{m_b}^{(h,l)}$ of $d = \sum_{i=1}^n d_i(\boldsymbol{\theta})$, i.e. the second term in the difference estimator (3.3), but from a mini-batch of $m_b < m$ observations. The scalar $a \in \mathbb{R}$ is a lower bound of the $\hat{d}_{m_b}^{(h,l)}$ to ensure that $\hat{p}_B(\mathbf{y}|\boldsymbol{\theta}) > 0$ for all $\boldsymbol{\theta}$.

Quiroz et al. (2018c) show that the Block-Poisson estimator is unbiased for the likelihood $p(\mathbf{y}|\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$. The product construction in the estimator is not used for variance reduction, but to induce dependency in the subsamples over the MCMC iterations, see Section 3.6 below; in fact, Quiroz et al. (2018c) prove that the variance of $\hat{p}_B(\mathbf{y}|\boldsymbol{\theta})$ is finite and exactly the same as the variance of the usual Poisson estimator in Papaspiliopoulos (2009).

To ensure that $\hat{p}_B(\mathbf{y}|\boldsymbol{\theta})$ in (3.16) is positive with probability 1, which is necessary for PMMH, a needs to be a lower bound of \hat{d}_{m_b} . Obtaining a lower bound is problematic for two reasons. First, a lower bound requires evaluating $d_i(\boldsymbol{\theta})$ for all data points. Second, $-a$ can be prohibitively large as the most extreme outcome of \mathbf{u} needs to be covered. This is problematic because Quiroz et al. (2018c) show that $\text{Var}(\hat{p}_B(\mathbf{y}|\boldsymbol{\theta}))$ is minimized for $a = d - \lambda$ for any given λ . Hence, λ must typically be very large in order for a to be a lower bound, and a large λ means many mini-batches and a high computational cost.

Quiroz et al. (2018c) instead advocate the use of a *soft lower bound*, which is a lower bound resulting in $\tau \equiv \Pr(\hat{p}_B(\mathbf{y}|\boldsymbol{\theta}) \geq 0)$ less than one, but close to it. Since the estimator might not be positive, the target cannot be defined as in (3.12). However, further augmenting the density $\tilde{\pi}(\boldsymbol{\theta}, \mathbf{u}, s)$ with the variable $s = \text{sign}(\hat{p}_B(\mathbf{y}|\boldsymbol{\theta})) \in \{-1, 1\}$, we obtain (cf. Section 3.4)

$$(3.17) \quad \tilde{\pi}(\boldsymbol{\theta}, \mathbf{u}, s) \equiv \frac{|\hat{p}_B(\mathbf{y}|\boldsymbol{\theta})|p(\boldsymbol{\theta})p(\mathbf{u})}{\tilde{p}(\mathbf{y})} = s\tilde{\pi}(\boldsymbol{\theta}, \mathbf{u}), \quad \text{with } \tilde{\pi}(\boldsymbol{\theta}, \mathbf{u}) \equiv \frac{\hat{p}_B(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})p(\mathbf{u})}{\tilde{p}(\mathbf{y})},$$

where $\tilde{p}(\mathbf{y}) = \iiint_{\boldsymbol{\theta}, s, \mathbf{u}} s \hat{p}_B(\mathbf{y}|\boldsymbol{\theta}) p(\mathbf{u}) p(\boldsymbol{\theta}) d\mathbf{u} ds d\boldsymbol{\theta}$ is a normalization constant. Note that if $\tau = \Pr(s = 1) = 1$, $\iint_{s, \mathbf{u}} \tilde{\pi}(\boldsymbol{\theta}, \mathbf{u}, s) d\mathbf{u} ds = \pi(\boldsymbol{\theta})$ and hence samples from the true posterior are obtained, instead of an approximation as in Section 3.4. We argued above that $\tau = 1$ is too expensive and therefore Quiroz et al. (2018c) follow Lyne et al. (2015), who cleverly note that

$$(3.18) \quad \mathbb{E}_{\pi}(\psi) = \frac{\int_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}) \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}}{\int_{\boldsymbol{\theta}} \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}} = \frac{\iint_{\boldsymbol{\theta}, \mathbf{u}} \psi(\boldsymbol{\theta}) s |\hat{p}_B(\mathbf{y}|\boldsymbol{\theta})| p(\mathbf{u}) p(\boldsymbol{\theta}) d\mathbf{u} d\boldsymbol{\theta}}{\iint_{\boldsymbol{\theta}, \mathbf{u}} s |\hat{p}_B(\mathbf{y}|\boldsymbol{\theta})| p(\mathbf{u}) p(\boldsymbol{\theta}) d\mathbf{u} d\boldsymbol{\theta}} = \frac{\mathbb{E}_{\tilde{\pi}}(\psi s)}{\mathbb{E}_{\tilde{\pi}}(s)}.$$

We can therefore obtain N samples from $\bar{\pi}(\boldsymbol{\theta}, u, s)$ in (3.17) and estimate (3.18) by

$$(3.19) \quad \widehat{\mathbb{E}}_{\pi}(\psi) = \frac{\sum_{i=1}^N \psi(\boldsymbol{\theta}^{(i)}) s^{(i)}}{\sum_{i=1}^N s^{(i)}},$$

that satisfies $\widehat{\mathbb{E}}_{\pi}(\psi) \xrightarrow{a.s.} \mathbb{E}_{\pi}(\psi)$ as $N \rightarrow \infty$. The approach in Lyne et al. (2015) of running PMMH on the absolute posterior followed by a sign-correction by importance sampling to consistently estimate expectations of functionals is termed *Signed PMMH* by Quiroz et al. (2018c).

Under the optimal variance condition $a = d - \lambda$ it remains to choose values for the tuning parameters λ and m_b . The natural approach is to choose λ and m_b to minimize a computational time similar to (2.5). Quiroz et al. (2018c) show that the computational time of Signed PMMH with the Block-Poisson estimator is

$$(3.20) \quad \text{CT}(\lambda, m_b) = m_b \lambda \frac{\text{IACT}(\sigma_{\log|\hat{p}_B|}^2(\lambda, m_b))}{(2\tau(\lambda, m_b) - 1)^2},$$

where $\sigma_{\log|\hat{p}_B|}^2(\lambda, m_b)$ is the variance of the log of the absolute value of the Block-Poisson estimator. To minimize $\text{CT}(\lambda, m_b)$ we need to compute i) IACT, ii) $\sigma_{\log|\hat{p}_B|}^2(\lambda, m_b)$ and iii) $\tau(\lambda, m_b)$. All three quantities are derived in closed form in Quiroz et al. (2018c) where practical strategies for optimally tuning of λ and m_b to minimize CT are also proposed. The derivations are made under idealized assumptions, but the tuning is demonstrated to be near optimal. Furthermore, the guidelines for selecting λ and m_b are shown to be conservative in the sense of not giving too low values for λ and m_b , which is known to be crucial in pseudo-marginal methods.

We end this subsection with a discussion of the possibility of using the Block-Poisson estimator in survey sampling, outside of a Subsampling MCMC context. We are not aware of survey sampling applications where the interest is in estimating a population product. However, the Poisson estimator is a special case of so called *debiasing* estimators (Rhee and Glynn, 2015). Such estimators are useful for unbiased estimation of a quantity (e.g. the likelihood) which is a non-linear function of a quantity that can easily be estimated unbiasedly (the log-likelihood). The debiasing approach resolves this issue for general functions. It is for example possible to apply this idea to debias calibration estimators (Deville and Särndal, 1992) such as the ratio estimator in survey sampling.

3.6. Dependent subsampling. We have argued that controlling the variance of the log of the likelihood estimator is crucial for the efficiency of PMMH. A closer inspection of Algorithm 2 shows that it is more correct to say it is the variance of the *difference* in the log likelihood estimates at $\boldsymbol{\theta}'$ and $\boldsymbol{\theta}^{(i-1)}$ that matters for PMMH. Using independent proposals for \mathbf{u} makes it easy to get a gross over-estimate of the likelihood at some iteration and get stuck, as illustrated in Figure 3.6. Refreshing only parts of the subsample in each iteration reduces the variance of the difference in the log of the estimates of the likelihood between the proposed and current point. This is achieved by making $\mathbf{u}^{(i-1)}$ (last accepted draw) and \mathbf{u}' (proposed draw) dependent. We now present two approaches from the Subsampling MCMC literature

for generating dependence in \mathbf{u} over the MCMC iterations, which were developed independently of the literature on repeated survey sampling for estimating changing populations over time (Steel and McLaren, 2009) in the survey sampling field. Much of this literature is focused on problems unrelated to Subsampling MCMC, for example how to avoid responders fatigue in repeated surveys, but this is certainly an area where the knowledge of survey statisticians can advance Subsampling MCMC.

The correlated pseudo-marginal. Deligiannidis et al. (2018) present a general Correlated Pseudo-Marginal (CPM) approach to dependent particles in PMMH. Their focus is on random effects models and particle filters in state-space models where the \mathbf{u} are usually Gaussian random numbers used to generate the importance samples or the particles. The correlation of the \mathbf{u} over iterations is achieved by an autoregressive proposal

$$(3.21) \quad \mathbf{u}' = \phi \mathbf{u}^{(i-1)} + \sqrt{1 - \phi^2} \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} \sim N(0, \mathbf{I})$. The tuning parameter ϕ is set close to one to generate high persistence in the iterates of the estimated likelihood, which makes it possible to run PMMH with a variance of the log of the estimated likelihood which is roughly two orders of magnitude larger than the variance around unity in the case with independently proposed \mathbf{u} (Deligiannidis et al., 2018).

Quiroz et al. (2018a) apply the approach in Deligiannidis et al. (2018) to a subsampling context. In subsampling without replacement, \mathbf{u} are binary variables with $u_i = 1$ if the i th observation is in the subsample; see Section 2.2. Quiroz et al. (2018a) propose using a two-state Markov Chain to generate binary dependent proposals where the transition probabilities are set to obtain the desired degree of persistence and a pre-determined expected subsample size m^* . They show that this can be formulated using the same autoregressive proposal with Gaussian random variables as in Deligiannidis et al. (2018) using a Gaussian Copula (Joe, 2014).

Block pseudo-marginal. Quiroz et al. (2018c) propose an alternative way to generate dependence in PMMH. For the specific problem of Subsampling MCMC without replacement, their Block Pseudo-Marginal (BPM) algorithms starts by partitioning the subsample indices $\mathbf{u} = (u_1, \dots, u_n)$ into G blocks: $\mathbf{u} = (\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(G)})$. For the Block-Poisson estimator, each block consist of \mathbf{u} 's in one or several of the λ products. Rather than updating all of \mathbf{u} as in regular PMMH, BPM updates only one of the blocks $\mathbf{u}^{(g)}$, $g \in \{1, \dots, G\}$ in each iteration, jointly with the model parameters $\boldsymbol{\theta}$. Updating only a single block in each iteration and leaving the other $G - 1$ blocks unchanged makes the log-likelihood estimates highly correlated over the iterations, again making it possible to use estimators with much larger variances and still not get stuck in the PMMH. BPM is a less general approach than CPM, but has a number of advantages over CPM when it is applicable. For example, the correlation ρ between log-likelihood estimates over the iterations is, under simplifying assumptions, $1 - 1/G$ (Quiroz et al., 2018c) and is therefore directly controlled by the number of blocks; in CPM the correlation between the logs of the estimated likelihoods is only indirectly and

nonlinearly controlled by ϕ . For subsampling, BPM offers some advantages, for example that only the u 's in the current block need to be generated.

3.7. Subsampling in Hamiltonian Monte Carlo. In Section 2.1 we presented the Random Walk Metropolis (RWM) algorithm which proposes θ using a random walk over the parameter space. RWM is a robust algorithm, but the local nature of RWM makes it very slow to traverse the posterior, especially in high-dimensional parameter spaces. This section presents Hamiltonian Monte Carlo, which can make much more distant proposals, and its recent extension to subsampling.

Hamiltonian Monte Carlo. Hamiltonian Monte Carlo (HMC), introduced in Duane et al. (1987), is a very popular algorithm for sampling from high-dimensional posteriors; see Neal (2011) and Betancourt (2017) for very accessible introductions to HMC. HMC augments the posterior $\pi(\theta)$ with fictitious momentum variables $\mathbf{m} \in \mathbb{R}^d$, of the same dimension as θ , and carries out the sampling on an extended target distribution $\tilde{\pi}(\theta, \mathbf{m})$. This is similar to the augmentation with the subsample indicators \mathbf{u} in PMMH, but the \mathbf{m} are not introduced to reduce computational cost, but to increase sampling efficiency. The momentum variables allow the algorithm to produce distant proposals while maintaining a high acceptance probability. HMC targets

$$(3.22) \quad \tilde{\pi}(\theta, \mathbf{m}) \propto \exp(-\mathcal{H}(\theta, \mathbf{m})),$$

where \mathcal{H} is the so called Hamiltonian, or total energy, which is here assumed to be separable in the potential (\mathcal{U}) and kinetic energies (\mathcal{K}):

$$(3.23) \quad \mathcal{H}(\theta, \mathbf{m}) = \mathcal{U}(\theta) + \mathcal{K}(\mathbf{m}),$$

where

$$(3.24) \quad \mathcal{U}(\theta) = -\log[p(\mathbf{y}|\theta)p(\theta)] \text{ and } \mathcal{K}(\mathbf{m}) = \frac{1}{2}\mathbf{m}^T\mathbf{M}^{-1}\mathbf{m},$$

and \mathbf{M} is a $d \times d$ positive definite matrix.

The HMC algorithm uses an initial momentum from $\mathbf{m} \sim N(0, \mathbf{M})$ to propagate both θ and \mathbf{m} over time t along a trajectory mapped out by the Hamiltonian dynamics

$$(3.25) \quad \nabla_t \theta = \nabla_{\mathbf{m}} \mathcal{H}(\theta, \mathbf{m}) = \mathbf{M}^{-1} \mathbf{m}$$

$$(3.26) \quad \nabla_t \mathbf{m} = -\nabla_{\theta} \mathcal{H}(\theta, \mathbf{m}) = -\nabla_{\theta} \mathcal{U}(\theta),$$

where ∇_t denotes the time derivative, and $\nabla_{\mathbf{m}}$ and ∇_{θ} are the gradients with respect to \mathbf{m} and θ , respectively. Hamiltonian dynamics has several very attractive properties (Neal, 2011), one of them being that it keeps the Hamiltonian conserved: $\nabla_t \mathcal{H} = 0$. Hamiltonian dynamics can therefore be used to generate proposals for θ over long distances that are accepted with probability one. In practical computer implementations, however, one needs to discretize the Hamiltonian dynamics, so the total energy is not preserved and we need a MH

accept/reject step with acceptance probability less than one. The most common way to discretize the Hamiltonian dynamics in HMC is the leapfrog method (Neal, 2011). Algorithm 3 outlines the complete HMC algorithm using the leapfrog method.

The performance of HMC is very sensitive to its two tuning parameters, the leapfrog step size ϵ and number of leapfrog steps L . The No-U-Turn algorithm proposed by Hoffman and Gelman (2014) is an effective method to tune ϵ and L .

Hamiltonian Monte Carlo with Energy Conserving Subsampling. HMC is a very efficient algorithm that scales well to high-dimensional posteriors, but it needs to repeatedly evaluate the gradient $\nabla_{\theta}\mathcal{U}(\theta)$ at each of the L leapfrog iterations in every MH iteration. Note that L typically needs to be rather large if we want to make distant moves without too much energy loss (ϵ small). Usually $\nabla_{\theta}\mathcal{U}(\theta)$ is costly whenever $\mathcal{U}(\theta)$ is costly, so the same computational hurdles discussed for standard MH apply also to HMC, but now to a much larger extent because of the L gradient evaluations in the leapfrog steps, see Algorithm 3 in Appendix A. Several authors have proposed running the leapfrog iterations on a subsample of the data to speed up computations (Neal, 2011; Chen et al., 2014; Betancourt, 2015), thereby replacing $\nabla_{\theta}\mathcal{U}(\theta)$ by an unbiased subsample estimate. However, such an approach strips HMC of its energy conserving property and distant proposals tend to be rejected with high probability (Betancourt, 2015). The energy loss comes from using a subsample estimate of the Hamiltonian dynamics that no longer operates on the true Hamiltonian used in the accept/reject step.

Dang et al. (2017) observe that this disconnect between the dynamics and the Hamiltonian can be easily avoided by extending the Subsampling MCMC algorithm in Quiroz et al. (2018a) to HMC proposals. The Energy Conserving Subsampling (HMC-ECS) algorithm in Dang et al. (2017) samples from the extended target

$$(3.27) \quad \bar{\pi}(\theta, \mathbf{m}, \mathbf{u}) \propto \exp(-\hat{\mathcal{H}}(\theta, \mathbf{m}, \mathbf{u}))p(\mathbf{u}),$$

where $p(\mathbf{u})$ is the distribution for the subsample selection indicators. The Hamiltonian in HMC-ECS is based on a subsample estimate of the Hamiltonian

$$(3.28) \quad \hat{\mathcal{H}}(\theta, \mathbf{m}, \mathbf{u}) = \hat{\mathcal{U}}(\theta, \mathbf{u}) + \mathcal{K}(\mathbf{m}),$$

where

$$(3.29) \quad \hat{\mathcal{U}}(\theta, \mathbf{u}) = -\left(\hat{\ell}_{DE}(\mathbf{y}|\theta, \mathbf{u}) - \frac{1}{2}\hat{\sigma}_{\ell_{DE}}^2 + \log p(\theta)\right) \text{ and } \mathcal{K}(\mathbf{m}) = \frac{1}{2}\mathbf{m}^T\mathbf{M}^{-1}\mathbf{m}.$$

The potential energy estimator in (3.29) makes HMC-ECS target a (slightly) perturbed posterior distribution, whose error can be controlled by the theory in Quiroz et al. (2018a).

Algorithm 4 gives the HMC-ECS algorithm. This sampler is a so called two-block Metropolis-within-Gibbs sampler which iteratively samples from the two full conditional posterior distributions

- $\mathbf{u} \sim \bar{\pi}(\mathbf{u}|\theta, \mathbf{m})$ using Metropolis-Hastings
- $(\theta, \mathbf{m}) \sim \bar{\pi}(\theta, \mathbf{m}|\mathbf{u})$ using HMC.

Following the usual HMC algorithm, the gradient used for generating proposal trajectories in the update for $(\boldsymbol{\theta}, \mathbf{m})$ in HMC-ECS is with respect to the target $\hat{\mathcal{U}}(\boldsymbol{\theta}, \mathbf{u})$.

The key feature of HMC-ECS is using the *same* subsample \mathbf{u} to estimate the Hamiltonian and to generate the trajectories in the Hamiltonian dynamics. Thus, HMC-ECS conserves the energy exactly as in the original HMC. As an example, Dang et al. (2017) compares HMC and HMC-ECS in a big data application on firm bankruptcy in a logistic additive spline model with a $d = 89$ -dimensional posterior. Dang et al. (2017) report that HMC-ECS gives an effective sample size that is up to three orders of magnitude larger than HMC for a given time budget. Moreover, the average acceptance probability of HMC-ECS is 79.3%, which is only marginally lower than the 81.8% for HMC.

4. CONCLUDING REMARKS

We have presented the pseudo-marginal approach to subsampling in Markov Chain Monte Carlo from the perspective of a survey statistician. We have reviewed several effective control variates for variance reduction of the likelihood estimator which make Subsampling MCMC scalable to large datasets. We have also presented methods for correlating the subsamples over the MCMC iterations, ultimately leading to algorithms that allow much more variable likelihood estimators. Much of the focus was given to unbiased estimators of the log-likelihood and methods for bias-correction of the resulting likelihood estimators. Focusing on the log-likelihood gives a direct analogy to estimating the population total, a long studied problem in survey sampling. This comes at the cost of giving an algorithm that samples from a slightly perturbed posterior, and we also review an alternative approach with unbiased likelihood estimators that can be used to obtain exact posterior expectations of functions of the parameters. We hope that this review makes it easier for survey statisticians to enter the field of Subsampling MCMC, and that it inspires them to make contributions to further enhance the efficiency of the algorithms.

5. ACKNOWLEDGEMENTS

Matias Quiroz and Robert Kohn were partially supported by Australian Research Council Center of Excellence grant CE140100049.

REFERENCES

- Alquier, P., Friel, N., Everitt, R., and Boland, A. (2016). Noisy Monte Carlo: Convergence of Markov chains with approximate transition kernels. *Statistics and Computing*, 26(1-2):29–47.
- Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725.
- Bardenet, R., Doucet, A., and Holmes, C. (2014). Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach. In *International Conference on Machine Learning (ICML)*, pages 405–413.
- Bardenet, R., Doucet, A., and Holmes, C. (2017). On Markov chain Monte Carlo methods for tall data. *Journal of Machine Learning Research*, 18(47):1–43.

- Beaumont, M. A. (2003). Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–1160.
- Betancourt, M. (2015). The fundamental incompatibility of scalable Hamiltonian Monte Carlo and naive data subsampling. In *International Conference on Machine Learning*, pages 533–540.
- Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*.
- Bierkens, J., Fearnhead, P., and Roberts, G. (2018). The zig-zag process and super-efficient sampling for Bayesian analysis of big data. *Annals of Statistics*, forthcoming.
- Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, forthcoming.
- Bouchard-Côté, A., Vollmer, S. J., and Doucet, A. (2018). The bouncy particle sampler: A nonreversible rejection-free markov chain monte carlo method. *Journal of the American Statistical Association*, pages 1–13.
- Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of Markov chain Monte Carlo*. CRC press.
- Ceperley, D. and Dewing, M. (1999). The penalty method for random walks with uncertain energies. *The Journal of Chemical Physics*, 110(20):9812–9820.
- Chen, C.-F. (1985). On asymptotic normality of limiting density functions with Bayesian implications. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(3):540–546.
- Chen, T., Fox, E., and Guestrin, C. (2014). Stochastic gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning*, pages 1683–1691.
- Dang, K.-D., Quiroz, M., Kohn, R., Tran, M.-N., and Villani, M. (2017). Hamiltonian Monte Carlo with energy conserving subsampling. *arXiv preprint arXiv:1708.00955*.
- Dean, J. and Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.
- Del Moral, P. (2004). *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Springer.
- Deligiannidis, G., Doucet, A., and Pitt, M. K. (2018). The correlated pseudo-marginal method. *Journal of the Royal Statistical Society B*, (forthcoming).
- Deville, J.-C. and Särndal, C.-E. (1992). Calibration estimators in survey sampling. *Journal of the American Statistical Association*, 87(418):376–382.
- Doucet, A., De Freitas, N., and Gordon, N. (2001). An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer.
- Doucet, A., Pitt, M., Deligiannidis, G., and Kohn, R. (2015). Efficient Implementation of Markov Chain Monte Carlo when using an Unbiased Likelihood Estimator. *Biometrika*, 102(2):295–313.
- Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222.
- Flury, T. and Shephard, N. (2011). Bayesian inference based only on simulated likelihood: particle filter analysis of dynamic economic models. *Econometric Theory*, 27(5):933–956.

- Gelman, A., Vehtari, A., Jylänki, P., Sivula, T., Tran, D., Sahai, S., Blomstedt, P., Cunningham, J. P., Schiminovich, D., and Robert, C. (2017). Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data. *arXiv preprint arXiv:1412.4869*.
- Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.
- Gunawan, D., Kohn, R., Quiroz, M., Dang, K.-D., and Tran, M.-N. (2018). Subsampling sequential Monte Carlo for static Bayesian models. *arXiv preprint arXiv:1805.03317*.
- Hammersley, J. M. and Handscomb, D. C. (1964). *Monte Carlo methods*. Chapman and Hall.
- Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.
- Hoffman, M. D. and Gelman, A. (2014). The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623.
- Joe, H. (2014). *Dependence modeling with copulas*. CRC Press.
- Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.
- Korattikara, A., Chen, Y., and Welling, M. (2014). Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 181–189.
- Lin, L., Liu, K., and Sloan, J. (2000). A noisy Monte Carlo algorithm. *Physical Review D*, 61(7):074505.
- Lyne, A.-M., Girolami, M., Atchade, Y., Strathmann, H., and Simpson, D. (2015). On Russian roulette estimates for Bayesian inference with doubly-intractable likelihoods. *Statistical Science*, 30(4):443–467.
- Maclaurin, D. and Adams, R. P. (2014). Firefly Monte Carlo: Exact MCMC with subsets of data. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI 2014)*, pages 543–552.
- Maire, F., Friel, N., and Alquier, P. (2018). Informed sub-sampling MCMC: approximate Bayesian inference for large datasets. *Statistics and Computing*, forthcoming.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.
- Minka, T. P. (2001). Expectation propagation for approximate Bayesian inference. In *Proceedings of the 17th conference on Uncertainty in Artificial Intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc.
- Minsker, S., Srivastava, S., Lin, L., and Dunson, D. (2014). Scalable and robust Bayesian inference via the median posterior. In *International Conference on Machine Learning*, pages 1656–1664.
- Neal, R. M. (2011). MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11).
- Neiswanger, W., Wang, C., and Xing, E. (2013). Asymptotically exact, embarrassingly parallel MCMC. *arXiv preprint arXiv:1311.4780*.

- Nemeth, C. and Sherlock (2018). Merging MCMC subposteriors through Gaussian-process approximations. *Bayesian Analysis*, 13(2):507–530.
- Nicholls, G. K., Fox, C., and Watt, A. M. (2012). Coupled MCMC with a randomized acceptance probability. *arXiv preprint arXiv:1205.6857*.
- Papaspiliopoulos, O. (2009). A methodological framework for Monte Carlo probabilistic inference for diffusion processes. *Manuscript*. Available at http://wrap.warwick.ac.uk/35220/1/WRAP_Papaspiliopoulos_09-31w.pdf.
- Pitt, M. K., dos Santos Silva, R., Giordani, P., and Kohn, R. (2012). On some properties of Markov Chain Monte Carlo simulation methods based on the particle filter. *Journal of Econometrics*, 171(2):134–151.
- Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). Coda: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11.
- Quiroz, M., Kohn, R., Villani, M., and Tran, M.-N. (2018a). Speeding up MCMC by efficient data subsampling. *Journal of the American Statistical Association*, (forthcoming).
- Quiroz, M., Tran, M.-N., Villani, M., and Kohn, R. (2018b). Speeding up MCMC by delayed acceptance and data subsampling. *Journal of Computational and Graphical Statistics*, 27:12–22.
- Quiroz, M., Tran, M.-N., Villani, M., Kohn, R., and Dang, K.-D. (2018c). The block-Poisson estimator for optimally tuned exact subsampling MCMC. *arXiv preprint arXiv:1603.08232*.
- Quiroz, M., Villani, M., and Kohn, R. (2014). Speeding up MCMC by efficient data subsampling. *arXiv preprint arXiv:1603.08232v1*.
- Rhee, C. and Glynn, P. W. (2015). Unbiased estimation with square root convergence for SDE models. *Operations Research*, 63(5):1026–1043.
- Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120.
- Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392.
- Särndal, C.-E., Swensson, B., and Wretman, J. (2003). *Model Assisted Survey Sampling*. Springer Science & Business Media.
- Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and McCulloch, R. E. (2016). Bayes and big data: The consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management*, 11(2):78–88.
- Sherlock, C., Thiery, A. H., Roberts, G. O., and Rosenthal, J. S. (2015). On the efficiency of pseudo-marginal random walk Metropolis algorithms. *The Annals of Statistics*, 43(1):238–275.
- Steel, D. and McLaren, C. (2009). Design and analysis of surveys repeated over time. *Handbook of Statistics*, 29:289–313.
- Van der Vaart, A. W. (1998). *Asymptotic Statistics*, volume 3. Cambridge university press.
- Wagner, W. (1988). Unbiased multi-step estimators for the Monte Carlo evaluation of certain functional integrals. *Journal of Computational Physics*, 79(2):336–352.

Wang, X. and Dunson, D. B. (2014). Parallel MCMC via Weierstrass sampler. *arXiv preprint arXiv:1312.4605v2*.

APPENDIX A. ALGORITHMS

This appendix contains the main sampling algorithms discussed in the paper.

Algorithm 1: The Metropolis-Hastings algorithm

Input: data \mathbf{y} , likelihood function $p(\mathbf{y}|\boldsymbol{\theta})$, prior density $p(\boldsymbol{\theta})$, proposal density $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$, random number generator for $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$, initial value $\boldsymbol{\theta}^{(0)}$, number of iterations N .

for $i = 1$ **to** N **do**
 | draw $\boldsymbol{\theta}' \sim q(\cdot|\boldsymbol{\theta}^{(i-1)})$
 | set $\boldsymbol{\theta}^{(i)} \leftarrow \boldsymbol{\theta}'$ with probability
 | $\alpha = \min\left(1, \frac{p(\mathbf{y}|\boldsymbol{\theta}')p(\boldsymbol{\theta}')}{p(\mathbf{y}|\boldsymbol{\theta}^{(i-1)})p(\boldsymbol{\theta}^{(i-1)})} \frac{q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}')}{q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(i-1)})}\right)$
 | else set $\boldsymbol{\theta}^{(i)} \leftarrow \boldsymbol{\theta}^{(i-1)}$
end

Output: autocorrelated random draws $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(N)}$ from $\pi(\boldsymbol{\theta}) \propto p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$.

Algorithm 2: The pseudo-marginal Metropolis-Hastings algorithm

Input: data \mathbf{y} , unbiased likelihood estimator $\hat{p}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u})$, prior density $p(\boldsymbol{\theta})$, proposal density $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$, random number generator for $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$, initial value $\boldsymbol{\theta}^{(0)}, \mathbf{u}^{(0)}$, random number generator for the augmentation variables \mathbf{u} , number of augmentation variables m , number of iterations N .

for $i = 1$ **to** N **do**
 | generate $\mathbf{u}' \sim p(\mathbf{u})$
 | generate $\boldsymbol{\theta}' \sim q(\cdot|\boldsymbol{\theta}^{(i-1)})$
 | set $(\boldsymbol{\theta}^{(i)}, \mathbf{u}^{(i)}) \leftarrow (\boldsymbol{\theta}', \mathbf{u}')$ with probability
 | $\alpha = \min\left(1, \frac{\hat{p}(\mathbf{y}|\boldsymbol{\theta}', \mathbf{u}')p(\boldsymbol{\theta}')}{\hat{p}(\mathbf{y}|\boldsymbol{\theta}^{(i-1)}, \mathbf{u}^{(i-1)})p(\boldsymbol{\theta}^{(i-1)})} \frac{q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}')}{q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(i-1)})}\right)$
 | else set $(\boldsymbol{\theta}^{(i)}, \mathbf{u}^{(i)}) \leftarrow (\boldsymbol{\theta}^{(i-1)}, \mathbf{u}^{(i-1)})$
end

Output: autocorrelated random draws $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(N)}$ from $\pi(\boldsymbol{\theta}) \propto p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$.

APPENDIX B. DETAILS FOR THE POISSON REGRESSION EXAMPLE

This appendix gives the details for the control variates in our illustrative Poisson regression example. Quiroz et al. (2018a) gives general expressions for the gradients and Hessians in the GLM class, and provides general compact expression that reduces the computational complexity of the control variates.

The Poisson regression model. The Poisson regression is of the form

$$y_i | \mathbf{x}_i, \boldsymbol{\theta} \stackrel{\text{indep.}}{\sim} \text{Pois}(\lambda_i), \quad \lambda_i = \exp(\alpha + \mathbf{x}_i^T \boldsymbol{\beta}),$$

where $\boldsymbol{\theta} = (\alpha, \boldsymbol{\beta})^T$.

Algorithm 3: Hamiltonian Monte Carlo (HMC)

Input: data \mathbf{y} , $p(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u})$, prior density $p(\boldsymbol{\theta})$, initial value $\boldsymbol{\theta}^{(0)}$, step size ϵ , number of leapfrog steps L , number of iterations N .

define $\mathcal{U}(\boldsymbol{\theta}) = -\log[p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})]$

define $\mathcal{K}(\mathbf{m}) = \frac{1}{2}\mathbf{m}^T\mathbf{M}^{-1}\mathbf{m}$

for $i = 1$ **to** N **do**

$\backslash\backslash$ generate trajectory by L leapfrog steps

 draw initial momentum $\mathbf{m} \sim N(0, \mathbf{M})$

 set $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}^{(i-1)}$

 set $\mathbf{m}' \leftarrow \mathbf{m} - \frac{\epsilon}{2}\nabla_{\boldsymbol{\theta}}\mathcal{U}(\boldsymbol{\theta}')$

for $l = 1$ **to** L **do**

 set $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}' + \epsilon\mathbf{M}^{-1}\mathbf{m}'$

if $l \neq L$ **then**

$\mathbf{m}' \leftarrow \mathbf{m}' - \epsilon\nabla_{\boldsymbol{\theta}}\mathcal{U}(\boldsymbol{\theta}')$

else

$\mathbf{m}' \leftarrow \mathbf{m}' - \frac{\epsilon}{2}\nabla_{\boldsymbol{\theta}}\mathcal{U}(\boldsymbol{\theta}')$

end

end

$\backslash\backslash$ accept or reject $(\boldsymbol{\theta}', \mathbf{m}')$

 set $\boldsymbol{\theta}^{(i)} \leftarrow \boldsymbol{\theta}'$ with probability

$$\alpha = \min \left[1, \exp \left(-\mathcal{U}(\boldsymbol{\theta}') + \mathcal{U}(\boldsymbol{\theta}^{(i-1)}) - \mathcal{K}(\mathbf{m}') + \mathcal{K}(\mathbf{m}) \right) \right]$$

 else set $\boldsymbol{\theta}^{(i)} \leftarrow \boldsymbol{\theta}^{(i-1)}$

end

Output: autocorrelated random draws $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(N)}$ from $\pi(\boldsymbol{\theta}) \propto p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$.

Parameter-expanded control variates. Let $\mathbf{w}_i = (1, \mathbf{x}_i^T)^T$. The log-likelihood contribution from the i th observation is

$$\ell_i(\boldsymbol{\theta}) = y_i \mathbf{w}_i^T \boldsymbol{\theta} - \exp(\mathbf{w}_i^T \boldsymbol{\theta}) - \log(y_i!)$$

with gradient and Hessian

$$\nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}) = (y_i - \exp(\mathbf{w}_i^T \boldsymbol{\theta})) \mathbf{w}_i$$

$$\nabla_{\boldsymbol{\theta}\boldsymbol{\theta}^T}^2 \ell_i(\boldsymbol{\theta}) = -\exp(\mathbf{w}_i^T \boldsymbol{\theta}) \mathbf{w}_i \mathbf{w}_i^T$$

Let $\mu(\boldsymbol{\theta}, \mathbf{x}) = \alpha + \mathbf{x}^T \boldsymbol{\beta} = \mathbf{w}^T \boldsymbol{\theta}$. The parameter-expanded control variate in (3.4) is then

$$\begin{aligned} \ell_i(\boldsymbol{\theta}) &\approx y_i \mu(\hat{\boldsymbol{\theta}}, \mathbf{x}_i) - \exp(\mu(\hat{\boldsymbol{\theta}}, \mathbf{x}_i)) - \log(y_i!) \\ &\quad + [y_i - \exp(\mu(\hat{\boldsymbol{\theta}}, \mathbf{x}_i))] (\mu_i(\boldsymbol{\theta}) - \mu_i(\hat{\boldsymbol{\theta}})) \\ &\quad - \frac{1}{2} \exp(\mu(\hat{\boldsymbol{\theta}}, \mathbf{x}_i)) (\mu(\boldsymbol{\theta}, \mathbf{x}_i) - \mu(\hat{\boldsymbol{\theta}}, \mathbf{x}_i))^2, \end{aligned}$$

Data-expanded control variates. The log-likelihood contribution from the i th observation is

$$\ell_i(\boldsymbol{\theta}) = y_i(\alpha + \mathbf{x}_i^T \boldsymbol{\beta}) - \exp(\alpha + \mathbf{x}_i^T \boldsymbol{\beta}) - \log(y_i!)$$

Algorithm 4: HMC with Energy Conserving Subsampling (HMC-ECS)

Input: data \mathbf{y} , unbiased likelihood estimator $\hat{p}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u})$, prior density $p(\boldsymbol{\theta})$, initial value $\boldsymbol{\theta}^{(0)}$, initial subsample $\mathbf{u}^{(0)}$, random number generator for \mathbf{u} , step size ϵ , number of leapfrog steps L , number of iterations N .

define $\mathcal{U}(\boldsymbol{\theta}, \mathbf{u}) = -\log[p(\mathbf{y}|\boldsymbol{\theta}, \mathbf{u})p(\boldsymbol{\theta})]$

define $\mathcal{K}(\mathbf{m}) = \frac{1}{2}\mathbf{m}^T\mathbf{M}^{-1}\mathbf{m}$

for $i = 1$ **to** N **do**

$\backslash\backslash$ update the subsample \mathbf{u}

 generate $\mathbf{u}' \sim p(\mathbf{u})$

 set $\mathbf{u}^{(i)} \leftarrow \mathbf{u}'$ with probability

$$\alpha_{\mathbf{u}} = \min\left(1, \frac{\hat{p}(\mathbf{y}|\boldsymbol{\theta}^{(i-1)}, \mathbf{u}')}{\hat{p}(\mathbf{y}|\boldsymbol{\theta}^{(i-1)}, \mathbf{u}^{(i-1)})}\right)$$

else set $\mathbf{u}^{(i)} \leftarrow \mathbf{u}^{(i-1)}$

$\backslash\backslash$ generate trajectory by L leap frog steps

 draw initial momentum $\mathbf{m} \sim N(0, \mathbf{M})$

 set $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}^{(i-1)}$

 set $\mathbf{m}' \leftarrow \mathbf{m} - \frac{\epsilon}{2}\nabla_{\boldsymbol{\theta}}\mathcal{U}(\boldsymbol{\theta}', \mathbf{u}^{(i)})$

for $l = 1$ **to** L **do**

 set $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}' + \epsilon\mathbf{M}^{-1}\mathbf{m}'$

if $l \neq L$ **then**

$\mathbf{m}' \leftarrow \mathbf{m}' - \epsilon\nabla_{\boldsymbol{\theta}}\mathcal{U}(\boldsymbol{\theta}', \mathbf{u}^{(i)})$

else

$\mathbf{m}' \leftarrow \mathbf{m}' - \frac{\epsilon}{2}\nabla_{\boldsymbol{\theta}}\mathcal{U}(\boldsymbol{\theta}', \mathbf{u}^{(i)})$

end

end

$\backslash\backslash$ accept or reject $(\boldsymbol{\theta}', \mathbf{m}')$

 set $\boldsymbol{\theta}^{(i)} \leftarrow \boldsymbol{\theta}'$ with probability

$$\alpha = \min\left[1, \exp\left(-\mathcal{U}(\boldsymbol{\theta}', \mathbf{u}^{(i)}) + \mathcal{U}(\boldsymbol{\theta}^{(i-1)}, \mathbf{u}^{(i)}) - \mathcal{K}(\mathbf{m}') + \mathcal{K}(\mathbf{m})\right)\right]$$

else set $\boldsymbol{\theta}^{(i)} \leftarrow \boldsymbol{\theta}^{(i-1)}$

end

Output: autocorrelated random draws $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(N)}$ from $\pi(\boldsymbol{\theta}) \propto p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$.

with gradient and Hessian

$$\nabla_{y_i}\ell_i(\boldsymbol{\theta}) = \alpha + \mathbf{x}_i^T\boldsymbol{\beta} - \psi_0(y_i + 1),$$

where $\psi_k(z) = \nabla_z^k \log \Gamma(z)$ is the polygamma function of order k ,

$$\nabla_{\mathbf{x}_i}\ell_i(\boldsymbol{\theta}) = (y_i - \exp(\alpha + \mathbf{x}_i^T\boldsymbol{\beta}))\boldsymbol{\beta}, \quad \nabla_{y_i}^2\ell_i(\boldsymbol{\theta}) = -\psi_1(y_i + 1),$$

$$\nabla_{\mathbf{x}_i\mathbf{x}_i^T}^2\ell_i(\boldsymbol{\theta}) = -\exp(\alpha + \mathbf{x}_i^T\boldsymbol{\beta})\boldsymbol{\beta}\boldsymbol{\beta}^T, \quad \text{and} \quad \nabla_{y_i\mathbf{x}_i^T}^2\ell_i(\boldsymbol{\theta}) = \boldsymbol{\beta}.$$

We can write the gradients and Hessian compactly by defining $\mathbf{z}_i = (y_i, \mathbf{x}_i^T)^T$,

$$\nabla_{\mathbf{z}_i}\ell_i(\boldsymbol{\theta}) = \begin{bmatrix} \alpha + \mathbf{x}_i^T\boldsymbol{\beta} - \psi_0(y_i + 1) \\ (y_i - \exp(\alpha + \mathbf{x}_i^T\boldsymbol{\beta}))\boldsymbol{\beta} \end{bmatrix}$$

$$\nabla_{z_i z_i^T}^2 \ell_i(\boldsymbol{\theta}) = \begin{bmatrix} -\psi_1(y_i + 1) & \beta^T \\ \beta & -\exp(\alpha + \mathbf{x}_i^T \beta) \beta \beta^T \end{bmatrix}.$$

Let $\mu(\boldsymbol{\theta}, \mathbf{x}) = \alpha + \mathbf{x}^T \beta$. The data-expanded control variate in (3.5) can after some simplifications be expressed as

$$\begin{aligned} \ell_i(\boldsymbol{\theta}) &\approx y_{c_i} \mu(\boldsymbol{\theta}, \mathbf{x}_{c_i}) - \exp(\mu(\boldsymbol{\theta}, \mathbf{x}_{c_i})) - \log(y_{c_i}!) \\ &+ (y_i - y_{c_i})(\mu(\boldsymbol{\theta}, \mathbf{x}_{c_i}) - \psi_0(y_{c_i} + 1)) - \frac{1}{2}(y_i - y_{c_i})^2 \psi_1(y_{c_i} + 1) \\ &+ [y_i - \exp(\mu(\boldsymbol{\theta}, \mathbf{x}_{c_i}))](\mu(\boldsymbol{\theta}, \mathbf{x}_i) - \mu(\boldsymbol{\theta}, \mathbf{x}_{c_i})) \\ &- \frac{1}{2} \exp(\mu(\boldsymbol{\theta}, \mathbf{x}_{c_i})) (\mu(\boldsymbol{\theta}, \mathbf{x}_i) - \mu(\boldsymbol{\theta}, \mathbf{x}_{c_i}))^2. \end{aligned}$$