Journal of Computational and Graphical Statistics

ISSN: 1061-8600 (Print) 1537-2715 (Online) Journal homepage: http://www.tandfonline.com/loi/ucgs20

Speeding up MCMC by Delayed Acceptance and Data Subsampling

Matias Quiroz, Minh-Ngoc Tran, Mattias Villani & Robert Kohn

To cite this article: Matias Quiroz, Minh-Ngoc Tran, Mattias Villani & Robert Kohn (2018) Speeding up MCMC by Delayed Acceptance and Data Subsampling, Journal of Computational and Graphical Statistics, 27:1, 12-22, DOI: 10.1080/10618600.2017.1307117

To link to this article: https://doi.org/10.1080/10618600.2017.1307117

Accepted author version posted online: 24 Mar 2017.
Published online: 28 Jul 2017.

Submit your article to this journal

Article views: 351

View related articles

View Crossmark data

Citing articles: 1 View citing articles

Full Terms & Conditions of access and use can be found at
http://www.tandfonline.com/action/journalInformation?journalCode=ucgs20

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2018, VOL. 27, NO. 1, 12–22
https://doi.org/10.1080/10618600.2017.1307117

Taylor & Francis
Taylor & Francis Group

Check for updates

Speeding up MCMC by Delayed Acceptance and Data Subsampling

Matias Quiroz[a,b], Minh-Ngoc Tran[c], Mattias Villani[a], and Robert Kohn[d]

[a]Division of Statistics and Machine Learning, Linköping University, Linköping, Sweden; [b]Research Division, Sveriges Riksbank, Stockholm, Sweden; [c]Discipline of Business Analytics, University of Sydney, Camperdown NSW, Australia; [d]Australian School of Business, University of New South Wales, Sydney NSW, Australia

ABSTRACT

The complexity of the Metropolis–Hastings (MH) algorithm arises from the requirement of a likelihood evaluation for the full dataset in each iteration. One solution has been proposed to speed up the algorithm by a delayed acceptance approach where the acceptance decision proceeds in two stages. In the first stage, an estimate of the likelihood based on a random subsample determines if it is likely that the draw will be accepted and, if so, the second stage uses the full data likelihood to decide upon final acceptance. Evaluating the full data likelihood is thus avoided for draws that are unlikely to be accepted. We propose a more precise likelihood estimator that incorporates auxiliary information about the full data likelihood while only operating on a sparse set of the data. We prove that the resulting delayed acceptance MH is more efficient. The caveat of this approach is that the full dataset needs to be evaluated in the second stage. We therefore propose to substitute this evaluation by an estimate and construct a state-dependent approximation thereof to use in the first stage. This results in an algorithm that (i) can use a smaller subsample m by leveraging on recent advances in Pseudo-Marginal MH (PMMH) and (ii) is provably within $O(m^{-2})$ of the true posterior.

ARTICLE HISTORY
Received September 2015
Revised December 2016

KEYWORDS
Bayesian inference; Delayed acceptance MCMC; Large data; Markov chain Monte Carlo; Survey sampling

1. Introduction

Markov chain Monte Carlo (MCMC) methods have been the workhorse for sampling from nonstandard posterior distributions in Bayesian statistics for nearly three decades. Recently, with increasingly more complex models and/or larger datasets, there has been a surge of interest in improving the $O(n)$ complexity emerging from the necessity of a complete data scan in each iteration of the algorithm.

There are a number of approaches proposed in the literature to speed up MCMC. Some authors divide the data into different partitions and carry out MCMC for the partitions in a parallel and distributed manner. The draws from each partition's subposterior are subsequently combined to obtain an approximation of the full posterior distribution. This line of work includes Scott et al. (2013); Neiswanger, Wang, and Xing (2014); Wang and Dunson (2013); Minsker et al. (2014); Nemeth and Sherlock (2016), among others. Other authors use a subsample of the data in each MCMC iteration to speed up the algorithm, see, for example, Korattikara, Chen, and Welling (2014), Bardenet, Doucet, and Holmes (2014), Maclaurin and Adams (2014), Maire, Friel, and Alquier (2015), Bardenet, Doucet, and Holmes (2015), and Quiroz et al. (2016, 2017). Finally, delayed acceptance MCMC has been used to speed up computations (Banterle, Grazian, and Robert 2014; Payne and Mallick 2015). The main idea in delayed acceptance is to avoid computations if there is an indication that the proposed draw will ultimately be rejected. Payne and Mallick (2015) considered a two-stage delayed acceptance MCMC that uses a random subsample of the data in the first stage to estimate the likelihood function at the proposed draw. If this estimate suggests that the proposed draw is likely to be rejected, the algorithm does not proceed to the second stage that evaluates the true likelihood (using all data). Banterle, Grazian, and Robert (2014) divided the acceptance ratio in the standard Metropolis–Hastings (MH; Metropolis et al. 1953; Hastings 1970) into several stages, which are sequentially performed until a first rejection is detected implying a final rejection of the proposed draw. Clearly, both delayed acceptance approaches save computations for proposals that are unlikely to be accepted, but on the other hand require all stages to be computed for those likely to be accepted. The latter corresponds to at least the same computational cost for one iteration as the standard MH.

This article extends the delayed acceptance algorithms of Payne and Mallick (2015) in the following directions. First, we replace the likelihood estimator of the first stage by an efficient estimator that employs control variates to significantly reduce the variance. We show that this modification results in an algorithm that is more effective in promoting good proposals to the second stage. Since this algorithm computes the true likelihood (as does MH) in the second stage whenever a draw is up for a final accept/reject decision, we refer to it as delayed acceptance (standard) MH (DA-MH). Second, we propose a delayed acceptance algorithm that overcomes the caveat of a full data likelihood evaluation in the second stage by replacing it with an estimate. This second-stage estimate is initially developed in the approximate subsampling pseudo-marginal MH

CONTACT Matias Quiroz quiroz.matias@gmail.com Division of Statistics and Machine Learning, Linköping University, 581 83 Linköping, Sweden.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/r/JCGS.
© 2018 American Statistical Association, Institute of Mathematical Statistics, and Interface Foundation of North America

(PMMH) framework in Quiroz et al. (2016). Thus, our second contribution is to speed up their algorithm by combining with delayed acceptance and we document large speedups for our so-called delayed acceptance PMMH (DA-PMMH) algorithm compared to MH. Payne and Mallick (2015) instead proposed to circumvent the full data evaluation by combining their delayed acceptance sampler with the consensus Monte Carlo in Scott et al. (2013), which currently lacks any control of the error produced in the approximate posterior. In contrast, we can make use of results in Quiroz et al. (2016) to control the approximation error and also ensure that it is $O(m^{-2})$, where m is the subsample size used for estimating the likelihood.

While the delayed acceptance MH has the disadvantage of using all data whenever a proposed draw proceeds to the second stage, we believe that the successful implementation provided here is essential if exact simulation is of importance. By exact simulation we mean that the Markov chain produced by the subsampling algorithm has the same invariant distribution as that of a Markov chain produced by an algorithm that uses all the data. Exact simulation using subsets of the data has proven to be extremely challenging. Pseudo-marginal MCMC (Beaumont 2003; Andrieu and Roberts 2009) provides a framework for conducting Markov chain simulation by only using an estimate (in this setting estimated from a subsample of the data) of the likelihood. Remarkably, although the true likelihood is never evaluated, the simulation is exact provided that the likelihood estimate is unbiased and almost surely positive. One route to exact simulation by data subsampling suggested by Bardenet, Doucet, and Holmes (2015) is to compute a sequence of unbiased estimators of the log-likelihood, apply the technique in Rhee and Glynn (2015) to debias the resulting likelihood estimator, and subsequently use it within a pseudo-marginal MCMC. Bardenet, Doucet, and Holmes (2015) noted that, as proved by Jacob and Thiery (2015), a lower bound on the log-likelihood estimators is needed to ensure positiveness. This typically requires computations using the full dataset, naturally defeating the purpose of subsampling. Quiroz et al. (2017) instead suggested to compute a soft lower bound that is defined to be a lower bound with a high probability. Since the estimate can occasionally be negative, the pseudo-marginal in Quiroz et al. (2017) instead targets an absolute measure following Lyne et al. (2015), who showed that the draws can be corrected with an importance sampling type of step to estimate expectations of posterior functions exactly. We note that although this certainly is exact inference of the expectation (it is not biased), the algorithm does not provide exact simulation as defined above. Maclaurin and Adams (2014) proposed Firefly Monte Carlo, which introduces an auxiliary variable for each observation, which determines if it is included when evaluating the likelihood. The distribution of these binary variables is such that when they are integrated out, the marginal posterior is the same as the one targeted by MH, thus providing exact simulation. Typically, a small fraction of observations is included, hence speeding up the execution time significantly compared to MH. However, the augmentation scheme severely affects the mixing of the Firefly algorithm and it has been demonstrated to perform poorly compared to MH and other subsampling approaches (Bardenet, Doucet, and Holmes 2015; Quiroz et al. 2016, 2017). We conclude that delayed acceptance, out of the discussed methods, seems to be the only feasible route to obtain exact simulation via data subsampling. We demonstrate that our implementation is crucial to obtain an algorithm that can improve on MH.

This article is organized as follows. Section 2 presents the efficient likelihood estimator. Sections 3 and 4 outline the delayed acceptance MH and PMMH methodologies in the context of subsampling. Section 5 applies the method to a micro-economic dataset containing nearly 5 million observations. Section 6 concludes and the Appendix proves Theorem 1.

2. Likelihood Estimators With Subsets of Data

2.1. Structure of the Likelihood

Let θ be the parameter in a model with density $p(y_k|\theta, x_k)$, where y_k is a potentially multivariate response vector and x_k is a vector of covariates for the kth observation. Let $l_k(\theta) = \log p(y_k|\theta, x_k)$ denote the kth observation's log-density, $k = 1, \ldots, n$. Given conditionally independent observations, the likelihood function can be written as

$$p(y|\theta) = \exp[l(\theta)], \tag{2.1}$$

where $l(\theta) = \sum_{k=1}^{n} l_k(\theta)$ is the log-likelihood function. To estimate $p(y|\theta)$, we first estimate $l(\theta)$ based on a sample of size m from the population $\{l_1(\theta), \ldots, l_n(\theta)\}$ and subsequently use (2.1). The first step corresponds to the classical survey sampling problem of estimating a population total: see Särndal, Swensson, and Wretman (2003) for an introduction to survey sampling. Note that (2.1) is more general than identically independently distributed (iid) observations, although we require that the log-likelihood can be written as a sum of terms, where each term depends on a unique piece of data information. One example is models with a random effect for each subject, with possibly multiple individual observations per subject (longitudinal data). In this case, a single term in the log-likelihood sum corresponds to the log joint density for a subject, and we therefore sample subjects (rather than individual observations) when estimating (2.1).

2.2. Data Subsampling

A main distinction of sampling schemes is whether the sample is obtained with or without replacement. It is clear that sampling with replacement gives a higher variance for any function of the sample: including the same element more than once does not provide any further information about finite population characteristics. However, it results in sample elements that are independent, which facilitates the derivation of Theorem 1 for the delayed acceptance MH. It also allows us to apply the theory and methodology in Quiroz et al. (2016) for the delayed acceptance PMMH in Section 4. It should be noted that the two sampling schemes are approximately the same when $m \ll n$.

Let $u = (u_1, \ldots u_m)$ be a vector of indices obtained by sampling m indices with replacement from $\{1, \ldots, n\}$ and let $\{l_{u_1}(\theta), \ldots, l_{u_m}(\theta)\}$ be the sample. We will consider simple random sampling (SRS), which means that

$$\Pr(u_i = k) = \frac{1}{n} \quad \text{for } k = 1, \ldots, n, \quad \text{and for all } i = 1, \ldots, m.$$

Payne and Mallick (2015) used SRS (but without replacement) to form an unbiased estimate of the log-likelihood. However, the elements $l_k(\theta)$ (for a fixed θ) vary substantially across the population and estimating the total $\sum_{k=1}^{n} l_k(\theta)$ with SRS is well known to be very inefficient under such a scenario. Instead $\Pr(u_i = k)$ should be (approximately) proportional to a size measure for $l_k(\theta)$. Quiroz et al. (2016) argued that this so-called proportional-to-size sampling is in many cases unlikely to be successful as knowledge of a size measure (which also depends on θ) for all k can often defeat the purpose of subsampling. They instead proposed to use SRS but incorporate control variates in the estimator for variance reduction which we now turn to.

2.3. Efficient Log-Likelihood Estimators

The idea in Quiroz et al. (2016) is to homogenize the population $\{l_1(\theta), \ldots, l_n(\theta)\}$: if the resulting elements are roughly of the same size then SRS is expected to be efficient. Let $q_k(\theta)$ denote an approximation of $l_k(\theta)$ and decompose

$$\begin{aligned} l(\theta) &= \sum_{k=1}^{n} q_k(\theta) + \sum_{k=1}^{n} \left[l_k(\theta) - q_k(\theta) \right] \\ &= q(\theta) + d(\theta), \end{aligned} \tag{2.2}$$

where

$$q(\theta) = \sum_{k=1}^{n} q_k(\theta), \quad d(\theta) = \sum_{k=1}^{n} d_k(\theta),$$

and

$$d_k(\theta) = l_k(\theta) - q_k(\theta).$$

We emphasize that all quantities depend on θ, which, from now on, is sometimes suppressed for a compact notation.

Define the random variables $\eta_i = n d_{u_i}$ and $d_{u_i} = l_{u_i} - q_{u_i}$ with $\mathrm{E}[\eta_i] = d$ and

$$\sigma_\eta^2 = \mathrm{V}[\eta_i] = n^2 \mathrm{V}[d_{u_i}] = n \sum_{k=1}^{n} (d_k - \bar{d})^2, \quad \text{with } \bar{d} = \sum_{k=1}^{n} d_k.$$

The difference estimator estimates d in (2.2) with the Hansen–Hurwitz estimator (Hansen and Hurwitz 1943),

$$\hat{d}_m = \frac{1}{m} \sum_{i=1}^{m} \eta_i, \quad \text{with } \mathrm{E}[\hat{d}_m] = d \quad \text{and} \quad \mathrm{V}[\hat{d}_m] = \frac{\sigma_\eta^2}{m}.$$

We can obtain an unbiased estimate $\hat{\sigma}_\eta^2 = n^2 \hat{\mathrm{V}}[d_{u_i}]$, where $\hat{\mathrm{V}}[d_{u_i}]$ is the usual unbiased sample variance estimator (of the population $\{d_1, \ldots d_n\}$). The estimator of the log-likelihood is thus

$$\hat{l}_m = \sum_{k=1}^{n} q_k(\theta) + \hat{d}_m, \quad \text{with} \quad \mathrm{E}[\hat{l}_m] = l$$

and

$$\sigma^2 = \mathrm{V}[\hat{l}_m] = \frac{\sigma_\eta^2}{m}. \tag{2.3}$$

It also follows that $\hat{\sigma}_m^2 = \hat{\sigma}_\eta^2 / m$ is an unbiased estimator of σ^2.

Quiroz et al. (2016) reasoned that observations close in data space (x_k, y_k) should, for a fixed θ, have similar $l_k(\theta)$ values. They clustered the data space into K clusters and approximated, within each cluster, $l_k(\theta)$ by a second-order Taylor series expansion $q_k(\theta)$ around the centroid of the cluster. This allows computing q using K evaluations (instead of n), see Quiroz et al. (2016) for details. Bardenet, Doucet, and Holmes (2015) proposed similar control variates, but instead expand with respect to θ around a reference value $\theta^\star$. While it can be shown that q can be computed using a single evaluation, the approximation can be inaccurate when θ is far from $\theta^\star$ giving a large σ^2.

We note that the log-likelihood estimate in Payne and Mallick (2015) (if with replacement sampling is used) is a special case of (2.3), namely, when $q_k = 0$ for all k. We will see that this results in a poor performance of the delayed acceptance algorithm and that control variates are crucial for a successful implementation.

2.4. Approximately Bias-Corrected Likelihood Estimator

It is clear that an unbiased log-likelihood estimator becomes biased for the likelihood when transformed to the ordinary scale by the exponential function. Since by the central limit theorem (CLT)

$$\sqrt{m} \left(\hat{l}_m(\theta) - l(\theta) \right) \to \mathcal{N}\left(0, \sigma_\eta^2\right) \quad \text{as} \quad m \to \infty,$$

Quiroz et al. (2016) (see also Ceperley and Dewing 1999; Nicholls, Fox, and Watt 2012) approximately bias corrected the likelihood estimate

$$\hat{p}_m(y|\theta, u) = \exp\left(\hat{l}_m(\theta) - \hat{\sigma}_\eta^2(\theta)/2m \right). \tag{2.4}$$

Equation (2.4) is unbiased if σ_η^2 is used in place of $\hat{\sigma}_\eta^2$ (and normality holds for $\hat{l}_m$). In practice, we need to use $\hat{\sigma}_\eta^2$ (to not use all data) and Quiroz et al. (2016) showed that (2.4) is asymptotically unbiased, with the bias decreasing as $O(m^{-2})$. For the rest of the article, we refer to (2.4) as a "bias-corrected" estimator, where the quotation marks highlight that the correction is not exact.

3. Delayed Acceptance MH With Subsets of Data

3.1. The Algorithm

Payne and Mallick (2015) proposed a subsampling delayed acceptance MH following Christen and Fox (2005). The aim in delayed acceptance is to simulate a Markov chain $\{\theta^{(j)}\}_{j=1}^{N}$, which admits the posterior

$$\pi(\theta) = \frac{p(y|\theta)p(\theta)}{p(y)},$$

where $p(y) = \int p(y|\theta)p(\theta)d\theta$ and $p(\theta)$ denotes the prior,

as invariant distribution. Moreover, the likelihood $p(y|\theta)$ should only be evaluated if there is a good chance of accepting the proposed θ.

The algorithm in Payne and Mallick (2015) proceeds as follows. Let $\theta_c = \theta^{(j)}$ denote the current state of the Markov chain. In the first stage, propose $\theta' \sim q_1(\theta|\theta_c)$ and compute

$$\alpha_1(\theta_c \to \theta') = \min\left\{ 1, \frac{\hat{p}_m(y|\theta', u)p(\theta')/q_1(\theta'|\theta_c)}{\hat{p}_m(y|\theta_c, u)p(\theta_c)/q_1(\theta_c|\theta')} \right\}, \tag{3.1}$$

where $\hat{p}_m(y|\theta, u)$ is the estimator in Section 2.4 without control variates for $\hat{l}_m$ (but not "bias-corrected," see below). Now, propose

$$\theta_p = \begin{cases} \theta' & \text{w.p.} \quad \alpha_1(\theta_c, \theta') \\ \theta_c & \text{w.p.} \quad 1 - \alpha_1(\theta_c, \theta'), \end{cases}$$

and move the chain to the next state $\theta^{(j+1)} = \theta_p$ with probability

$$\alpha_2(\theta_c \to \theta_p) = \min\left\{1, \frac{p(y|\theta_p)p(\theta_p)/q_2(\theta_p|\theta_c)}{p(y|\theta_c)p(\theta_c)/q_2(\theta_c|\theta_p)}\right\}, \quad (3.2)$$

where

$$q_2(\theta_p|\theta_c) = \alpha_1(\theta_c \to \theta_p)q_1(\theta_p|\theta_c) + r(\theta_c)\delta_{\theta_c}(\theta_p),$$
$$r(\theta_c) = 1 - \int \alpha_1(\theta_c \to \theta_p)q_1(\theta_p|\theta_c)d\theta_p,$$

and δ is the Dirac delta function. If rejected we set $\theta^{(j+1)} = \theta_c$.

Note that α_2 in (3.2) is equivalent to the acceptance probability of a standard MH with a proposal density $q_2(\theta_p|\theta_c)$ that is a mixture of two proposal densities: the first proposes to move from θ_c to θ_p (from the "slab" q_1) and the second proposes to stay at θ_c (from the "spike"). The mixture weight for the "slab" is α_1 in (3.1) (with $\theta' = \theta_p$): if the likelihood estimate is higher at θ' compared to that of θ_c (after correcting with q_1), we propose $\theta' = \theta_p$ with probability 1. Conversely, if it is lower, we propose to move but with a probability smaller than 1, which decreases the less likely we think that θ' will be accepted as indicated by the estimated likelihood. The estimator $\hat{p}_m(y|\theta', u)$ used by Payne and Mallick (2015) does not depend on the current state of the Markov chain and hence the mixture weights in q_2 are state-independent. Convergence to the invariant distribution therefore follows from standard MH theory. The same applies when the estimator uses the control variates in Quiroz et al. (2016), or in Bardenet, Doucet, and Holmes (2015) but only for a fixed $\theta^\star$. Bardenet, Doucet, and Holmes (2015) suggested setting $\theta^\star = \theta_c$ every now and then to prevent that the control variates can be poor if the chain is far from $\theta^\star$: the resulting approximation is clearly state-dependent and standard MH theory does not apply. Instead, convergence to the invariant $\pi(\theta)$ is proved in Christen and Fox (2005) and the delayed algorithm is exactly as above but with

$$\hat{p}_m(y|\cdot, u) = \hat{p}_m^{(\theta_c)}(y|\cdot, u),$$

emphasizing that it depends on the current state θ_c.

The state-dependent algorithm is a key ingredient when developing the delayed acceptance block PMMH in Section 4.

As noted above, Payne and Mallick (2015) did not "bias correct" the likelihood estimate as they pointed out that the algorithm will have the correct invariant distribution anyway. We remark that without control variates it is not a good idea to apply the correction as the variance of $\hat{\sigma}_m^2$ is huge, which adversely affects $\hat{p}_m(y|\cdot, u)$. An efficient estimate of $\hat{\sigma}_m^2$, however, would certainly improve $\hat{p}_m(y|\cdot, u)$. While the control variates allow us to estimate $\hat{\sigma}_m^2$ accurately, we will not implement the "bias-correction" when comparing to Payne and Mallick (2015) to make the comparison fair.

It is beneficial to also update u to avoid the risk of having a subset of observations for which the approximation is poor. This is especially important for the estimator in Payne and Mallick (2015), as not using control variates can result in the particular subset having highly heterogenous elements, which is detrimental for SRS. We note that updating u is still a valid MH because (i) u is not a state of the Markov chain and (ii) the distribution $p(u) = 1/n^m$ (SRS) does not depend on θ. Thus, the transition kernel of an algorithm that updates u is a state-independent mixture of transition kernels, where each of the kernels satisfies detailed balance either by standard MH (or Christen and Fox (2005) if the approximation depends on θ_c). Since the weights $1/n^m$ do not depend on θ, it follows that the mixture also satisfies detailed balance, and thus has $\pi(\theta)$ as invariant distribution. It is unnecessary to update u in every iteration, instead one can update u randomly to save the overhead cost of indexing the data matrix when obtaining the subset of observations.

We remark that delayed acceptance (sometimes with names as early rejection or surrogate transition), although without data subsampling, has been considered earlier in the literature. References include Fox and Nicholls (1997); Liu (2008); Cui, Fox, and O'Sullivan (2011); Smith (2011); Solonen et al. (2012); Golightly, Henderson, and Sherlock (2015); Sherlock, Golightly, and Henderson (2017). Each stage in Banterle, Grazian, and Robert (2014) (see Section 1) uses a partition of the data and can thus be considered as delayed acceptance with data subsampling. The advantage of our algorithm is that, because it only has two stages and the second stage evaluates the full data likelihood, we can instead estimate this likelihood to never do the evaluation for the full dataset, see Section 4.

3.2. Efficiency of Delayed Acceptance MH When Subsampling the Data

When considering efficiency of MCMC algorithms with additional computational costs (e.g., estimating the target), there are two types of fundamentally different efficiencies that interplay. The first is the statistical efficiency, which we will measure by the asymptotic (as the number of MCMC iterates go to infinity) variance of an estimate based on output from the Markov chain. Consider two MCMC algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ with the same invariant distribution. Then $\mathcal{A}_1$ is statistically more (less) efficient than $\mathcal{A}_2$ if it has a lower (higher) asymptotic variance. The second is the computational efficiency, which solely concerns "execution time" to produce a given number of iterates. The measures of statistical and computational efficiency (and a combination of them) used in this article are presented later in this section.

Sherlock, Thiery, and Golightly (2015a) (in a nonsubsampling context) studied the statistical efficiency for delayed acceptance random walk Metropolis and, moreover, an efficiency that also takes into account the computational efficiency for the case where the target is estimated (DA-PMMH in Section 4).

Christen and Fox (2005) noted that, because the transition kernels of both the MH and delayed acceptance MH are derived from the same proposal q_1, and in addition $\alpha_2 \leq 1$, the delayed acceptance MH will be less statistically efficient than MH. The intuition is that under these conditions the chain clearly exhibits a more "sticky" behavior and an estimate based on these samples will have a larger asymptotic variance under DA-MH than MH. Notice that the closer α_2 is to 1, the more statistically efficient the

delayed acceptance algorithm is, and when $\alpha_2 = 1$ it is equivalent to the standard MH, which gives the upper bound of the possible statistical efficiency achieved by a DA-MH.

Result 1 in Payne and Mallick (2017) gives the alternative formulation (for state-independent approximations)

$$\alpha_2(\theta_c \to \theta_p) = \min\left\{1, \frac{\hat{p}_m(y|\theta_c, u)/p(y|\theta_c)}{\hat{p}_m(y|\theta_p, u)/p(y|\theta_p)}\right\}. \quad (3.3)$$

Let $l_k(\theta_c, \theta_p) = l_k(\theta_c) - l_k(\theta_p)$ and denote by $\hat{l}_m(\theta_c, \theta_p)$ the estimate of $l(\theta_c, \theta_p) = \sum_{k=1}^{n} l_k(\theta_c, \theta_p)$. Similarly to (2.3),

$$\hat{l}_m(\theta_c, \theta_p) = q(\theta_c, \theta_p) + \frac{1}{m}\sum_{i=1}^{m}\zeta_i,$$

with

$$q(\theta_c, \theta_p) = \sum_{k=1}^{n} q_k(\theta_c, \theta_p), \quad (3.4)$$

where $q_k(\theta_c, \theta_p) = q_k(\theta_c) - q_k(\theta_p)$ and the ζ_i's are iid with

$$\Pr\left(\zeta_i = nD_k(\theta_c, \theta_p)\right) = \frac{1}{n}, \quad \text{with } D_k = \left(l_k(\theta_c, \theta_p) - q_k(\theta_c, \theta_p)\right) \text{ for } i = 1, \ldots m.$$

We can also show that

$$\mathrm{E}[\hat{l}_m(\theta_c, \theta_p)] = l(\theta_c, \theta_p) \quad \text{and} \quad V[\hat{l}_m(\theta_c, \theta_p)] = \frac{\sigma_\zeta^2}{m}$$

with

$$\sigma_\zeta^2 = n\sum_{k=1}^{n}\left(D_k(\theta_c, \theta_p) - \bar{D}_F(\theta_c, \theta_p)\right)^2, \quad (3.5)$$

where $\bar{D}_F$ is the mean over the full population. When not "bias-corrected," the ratio appearing in (3.3) becomes

$$R_m = \exp\left(\hat{l}_m(\theta_c, \theta_p) - l(\theta_c, \theta_p)\right). \quad (3.6)$$

We now propose a theorem that relates $\mathrm{E}[\alpha_2(\theta_c \to \theta_p)]$ to the variance $\sigma_R^2 = \mathrm{V}[\hat{l}_m(\theta_c, \theta_p)]$ under the assumption that $\hat{l}_m(\theta_c, \theta_p) \sim \mathcal{N}(l(\theta_c, \theta_p), \sigma_R^2)$ (equivalently $R_m \sim \log\mathcal{N}(0, \sigma_R^2)$ in (3.6)). In turn, α_2 relates to the statistical efficiency as discussed above. The assumption of normality is justified by a standard CLT for $\hat{l}_m(\theta_c, \theta_p)$ since the ζ_i's are iid.

Theorem 1. Suppose that we run a DA-MH with a state-independent approximation

$$\hat{l}_m(\theta_c, \theta_p) \sim \mathcal{N}\left(l(\theta_c, \theta_p), \sigma_R^2\right),$$

where

$$\sigma_R^2(\theta_c, \theta_p) = \mathrm{V}[\log(R_m)],$$

which has a second-stage acceptance probability

$$\alpha_2(\theta_c \to \theta_p) = \min\left(1, R_m\right),$$
$$R_m = \exp\left(\hat{l}_m(\theta_c, \theta_p) - l(\theta_c, \theta_p)\right).$$

Then

$$\mathrm{E}[\alpha_2(\theta_c \to \theta_p)] = \exp\left(\sigma_R^2(\theta_c, \theta_p)/2\right)\left(1 - \Phi\left(\sigma_R(\theta_c, \theta_p)\right)\right) + 0.5.$$

In particular, $\mathrm{E}[\alpha_2(\theta_c \to \theta_p)]$ is a decreasing function of σ_R.

Proof. See the Appendix. □

Remark. It is possible to state and prove a similar theorem using a "bias-corrected" likelihood estimator, which in log-scale is $\hat{l}_m(\theta_c, \theta_p) - \hat{\sigma}_R^2(\theta_c, \theta_p)/2$. This is omitted as our application in Example 1 in Section 5 does not use a "bias-corrected" likelihood estimator to conduct a fair comparison to Payne and Mallick (2015).

Theorem 1 says that if the variance of the log of the ratio R_m in (3.6) is lower, then the algorithm has a higher α_2 on average. Moreover, it shows that α_2 deteriorates quickly with σ_R, which illustrates the importance of achieving a small σ_R.

Although the delayed acceptance MH is always less statistically efficient than MH it can of course still be more generally efficient in terms of balancing computational and statistical efficiency. Clearly, a necessary condition to achieve a more general efficient DA-MH algorithm than the corresponding MH requires that its computing time is faster, that is, it must be computationally more efficient. When comparing DA-MH algorithms with the same computing time (by, e.g., having the same subsample size) then Theorem 1 shows that a smaller variance of the log-ratio results in a more general efficient algorithm.

To also compare algorithms of different computing times, we now define a measure for the general efficiency discussed above. In the rest of the article, whenever we claim that algorithms are more or less efficient to each other it is based on this measure. The statistical (in)efficiency part is measured by the inefficiency factor (IF), which quantifies the amount by which the variance of $(1/N)\sum_{j=1}^{N}\theta^{(j)}$ is inflated when $\theta^{(j)}$ is obtained by Markov chain simulation compared to that of iid simulation. It is given by

$$\mathrm{IF} = 1 + 2\sum_{l=1}^{\infty}\rho_l, \quad (3.7)$$

where ρ_l is the autocorrelation at the lth lag of the chain, and can be computed with the coda package in R (Plummer et al. 2006). To include computational efficiency in the measure, we use effective draws (ED) per computing unit

$$\mathrm{ED} = \frac{N}{\mathrm{IF} \times t}, \quad (3.8)$$

where N is the number of MCMC iterations and t is the computing time. The measure of interest is the effective draws per computing time of a delayed acceptance algorithm $\mathcal{A}$ relative to that of MH, that is,

$$\mathrm{RED} = \frac{\mathrm{ED}^{\mathcal{A}}}{\mathrm{ED}^{\mathrm{MH}}}. \quad (3.9)$$

4. Delayed Acceptance PMMH With Subsets of Data

4.1. PMMH for Data Subsampling

Quiroz et al. (2016) proposed a pseudo-marginal approach to subsampling based on the "bias-corrected" likelihood estimator in (2.4). In the next subsection, this estimate replaces the true likelihood, thereby avoiding the evaluation of the full dataset. To allow for a smaller subsample size without adversely affecting the mixing of the chain, Quiroz et al. (2016) developed a correlated pseudo-marginal approach based on Deligiannidis, Doucet, and Pitt (2016). Tran et al. (2016) (see

also Quiroz et al. 2016) used an alternative approach to correlated pseudo-marginal, which we use for our delayed acceptance block PMMH (DA-BPMMH) in Section 4.3 Although not considered here, it is straightforward to instead correlate the subsamples as in Deligiannidis, Doucet, and Pitt (2016).

Pseudo-marginal by data subsampling targets the following posterior on an augmented space (θ, u),

$$\tilde{\pi}_m(\theta, u) = \hat{p}_m(y|\theta, u)p(u)p(\theta)/p_m(y),$$

with

$$p_m(y) = \int p_m(y|\theta)p(\theta)d\theta. \tag{4.1}$$

The algorithm is similar to MH except that θ and u are proposed and accepted (or rejected) jointly, with probability

$$\alpha_{\text{PMMH}} = \min\left\{1, \frac{\hat{p}_m(y|\theta_p, u_p)p(\theta_p)/q(\theta_p|\theta_c)}{\hat{p}_m(y|\theta_c, u_c)p(\theta_c)/q(\theta_c|\theta_p)}\right\}, \tag{4.2}$$

where the subscripts denote the current and proposed values of θ and u. Quiroz et al. (2016) showed that the algorithm converges to a slightly perturbed target (only approximately unbiased likelihood estimator) $\pi_m(\theta)$ and proved that

$$\frac{|\pi_m(\theta) - \pi(\theta)|}{\pi(\theta)} \leq O(m^{-2}).$$

Moreover, if $h(\theta)$ is a function that is absolutely integrable with respect to $\pi(\theta)$, then $\mathrm{E}_{\pi_m}[h(\theta)]$ is also within $O(m^{-2})$ of its true value.

The variance of $\hat{l}_m$ is crucial for the general efficiency of the algorithm: σ^2 in the approximate interval $[1, 3.3]$ is optimal (Pitt et al. 2012; Doucet et al. 2015; Sherlock et al. 2015b). The correlated pseudo-marginal approach induces a high positive correlation between the estimates in (4.2) by correlating u_c and u_p. This allows the use of a less precise estimator without getting stuck: the errors in the numerator and denominator tend to cancel. As a consequence, σ^2 can be larger than above, hence speeding up the algorithm by taking a smaller subsample. We follow Tran et al. (2016) and divide u (defined in Section 2.2) into G blocks,

$$u = (u_{(1)}, \ldots, u_{(G)}) \quad \text{with } \frac{m}{G} \text{ observations within each block,}$$

and update a single block (randomly) in each iteration. Setting G large induces a high positive correlation ρ between $\hat{l}_m(\theta_c)$ and $\hat{l}_m(\theta_p)$: Tran et al. (2016) showed that $\rho = 1 - 1/G$ under certain assumptions. We set $G = 100$ for the state-dependent algorithm in our application.

4.2. State-Independent Algorithm: DA-PMMH

Delayed acceptance PMMH uses an estimate of the target in the second stage of the algorithm. Such algorithms have recently been considered by Sherlock, Thiery, and Golightly (2015a); Sherlock, Golightly, and Henderson (2017); and Golightly, Henderson, and Sherlock (2015). We propose a state-independent data subsampling DA-PMMH (uncorrelated PMMH) and a state-dependent (block PMMH, where the correlation between the subsamples follows Tran et al. 2016) extension DA-Block (correlated) PMMH (DA-BPMMH) in the next subsection.

The estimated "bias-corrected" likelihood in (2.4) is approximated in a first-stage screening by $\hat{s}$, which is discussed in Section 4.4. Similarly to the algorithm in Section 3.1, we desire to only evaluate the (estimated) target (on the augmented space) if the proposed state is likely to be accepted. Let $(\theta_c, u_c) = (\theta^{(j)}, u^{(j)})$ denote the current state of the augmented Markov chain. In the first stage, propose $\theta' \sim q_1(\theta|\theta_c)$ and $u' \sim p(u)$ and evaluate

$$\alpha_1^{\text{PMMH}}\left\{(\theta_c, u_c) \to (\theta', u')\right\} = \min\left\{1, \frac{\hat{s}(\theta', u')p(\theta')/q_1(\theta'|\theta_c)}{\hat{s}(\theta_c, u_c)p(\theta_c)/q_1(\theta_c|\theta')}\right\}, \tag{4.3}$$

where $\hat{s}(\theta, u)$ is the approximation of $\hat{p}_m(y|\theta, u)$ (2.4). Propose

$$(\theta_p, u_p) = \begin{cases} (\theta', u') & \text{w.p. } \alpha_1^{\text{PMMH}}\left\{(\theta_c, u_c) \to (\theta', u')\right\} \\ (\theta_c, u_c) & \text{w.p. } 1 - \alpha_1^{\text{PMMH}}\left\{(\theta_c, u_c) \to (\theta', u')\right\}, \end{cases}$$

and move to $(\theta^{(j+1)}, u^{(j+1)}) = (\theta_p, u_p)$ with probability

$$\alpha_2^{\text{PMMH}}\left\{(\theta_c, u_c) \to (\theta_p, u_p)\right\} = \min\left\{1, \frac{\hat{p}_m(y|\theta_p, u_p)p(\theta_p)/q_2(\theta_p|\theta_c)}{\hat{p}_m(y|\theta_c, u_c)p(\theta_c)/q_2(\theta_c|\theta_p)}\right\}, \tag{4.4}$$

where

$$q_2(\theta_p|\theta_c) = \alpha_1^{\text{PMMH}}q_1(\theta_p|\theta_c) + r(\theta_c)\delta_{\theta_c}(\theta_p),$$
$$r(\theta_c) = 1 - \int \alpha_1^{\text{PMMH}}q_1(\theta_p|\theta_c)d\theta_p,$$

and δ is the Dirac delta function. If rejected we set $(\theta^{(j+1)}, u^{(j+1)}) = (\theta_c, u_c)$. Similarly to the argument for the DA-MH, we recognize α_2^{PMMH} as the acceptance probability for a pseudo-marginal algorithm. Since the approximation is state-independent ($\hat{s}(\theta, u)$ independent of the current state implies state-independent mixture weights for q_2), convergence follows from being a pseudo-marginal algorithm (Andrieu and Roberts 2009). However, as the estimate is biased, the target is perturbed (Quiroz et al. 2016) but within $O(m^{-2})$ as discussed in Section 4.1

4.3. State-Dependent Algorithm: DA-BPMMH

As u is part of the state in a pseudo-marginal algorithm, obtaining a state-dependent approximation is easily achieved by correlating u: we sample $u_p \sim p(u|u_c)$ thus $\hat{s}(\theta, u) = \hat{s}_{\theta_c, u_c}(\theta, u)$. This is the state-dependent (on the augmented space) setup in Christen and Fox (2005) and it follows that the invariant distribution is $\tilde{\pi}_m(\theta, u)$ in (4.1). This is the invariant distribution in the algorithm in Quiroz et al. (2016), which we already mentioned has a marginal $\pi_m(\theta)$ within $O(m^{-2})$ of $\pi(\theta)$. We note that this state-dependent approximation is very convenient because it automatically allows us to have a higher variance on $\hat{l}_m$ because of the block PMMH mechanism (see Section 4.1).

4.4. An Approximation of the Estimated Likelihood

Our approximation is inspired by adaptive delayed acceptance ideas in Cui, Fox, and O'sullivan (2011) and Sherlock, Golightly, and Henderson (2017). However, our approach is not strictly adaptive as we only learn about the proposal for a fixed training period of N_{train} iterations, which are discarded from the final draws.

The idea is to use a sparser set of the data to construct control variates $q_k^{(1)}(\theta)$ in the first stage. During the training period, we learn about the discrepancy between $q^{(1)}(\theta) = \sum_{k=1}^{n} q_k^{(1)}(\theta)$ and $q(\theta)$ of the second stage (obtained with a denser set), that is,

$$\underbrace{q(\theta) - q^{(1)}(\theta)}_{=e(\theta)} = f(\theta) + \epsilon,$$
$$\epsilon \text{ is the noise (assumed independent of } \theta). \quad (4.5)$$

Learning about f is a standard regression problem: the collection of proposed θ's during the fixed training period are the inputs and the discrepancies are the training data.

The trivial decomposition

$$\begin{aligned}\hat{l}_m(\theta) &= \sum_{k=1}^{n} q_k(\theta) + \hat{d}_m(\theta) - \hat{\sigma}_m^2/2 \\ &= \sum_{k=1}^{n} q_k^{(1)}(\theta) + \left(\sum_{k=1}^{n} q_k(\theta) - \sum_{k=1}^{n} q_k^{(1)}(\theta)\right) \\ &\quad + \hat{d}_m(\theta) - \hat{\sigma}_m^2/2\end{aligned}$$

suggests the first-stage approximation

$$\hat{s}(\theta, u) = \sum_{k=1}^{n} q_k^{(1)}(\theta) + \hat{e}(\theta) + \hat{d}_m(\theta) - \hat{\sigma}_m^2/2,$$

where $\hat{e}(\theta)$ is the prediction of the discrepancy at θ. Note that computing the true discrepancy requires the denser set to be evaluated, whereas the prediction is very fast. For example, in linear (or nonlinear by basis functions) regression the parameters of $f(\theta)$ are estimated once after the training data have been collected. Prediction of $e(\theta)$ for a new "data-observation" θ is then a simple dot product computation, which is typically much faster than evaluating $q(\theta)$. Note that if the u's are correlated then $\hat{s}(\theta, u)$ also depends on the current state, that is, $\hat{s}_{\theta_c, u_c}(\theta, u)$.

In Section 5, we implement both a linear regression and (noise-free) Gaussian process to learn f in (4.5), but any regression technique can be used.

5. Application

5.1. Data and Model

We model the probability of bankruptcy conditional on a set of covariates using a dataset of 534,717 Swedish firms for the time period 1991–2008. We have in total $n = 4{,}748{,}089$ firm-year observations. The variables included are: earnings before interest and taxes, total liabilities, cash and liquid assets, tangible assets, logarithm of deflated total sales, and logarithm of firm age in years. We also include the macroeconomic variables GDP-growth rate (yearly) and the interest rate set by the Swedish central bank. See Giordani et al. (2014) for a detailed description of the dataset.

We consider the logistic regression model

$$p(y_k|x_k, \beta) = \left(\frac{1}{1 + \exp\left(x_k^T\beta\right)}\right)^{y_k} \left(\frac{1}{1 + \exp\left(-x_k^T\beta\right)}\right)^{1-y_k},$$

where x_k includes the variables above plus an intercept term. We set $p(\beta) \sim N(0, 10I)$ for simplicity. Since the bankruptcy observations ($y_k = 1$) are sparse in the data, we follow Payne and Mallick (2015) and estimate the likelihood only for the $y_k = 0$ observations. That is, we decompose

$$l(\beta) = \sum_{\{k; y_k = 1\}} l_k(\beta) + \sum_{\{k; y_k = 0\}} l_k(\beta),$$

and evaluate the first term whereas a random sample is only taken to estimate the second term. The second term clearly follows the structure presented in Section 2.1.

5.2. Performance Evaluation Measures

The quantity of interest is the effective draws as defined in (3.8). We now outline how to compute t as CPU time and present an alternative measure independent of the implementation that is based on number of evaluations. We first outline a robust measure of the CPU time.

The delayed acceptance algorithms we implement have two stages, where the first stage is filtering out draws unlikely to be accepted. One can view MH as an algorithm that does not filter any proposed draws at all: any draw is subject to an accept/reject decision based on the second-stage likelihood. To make total CPU time comparisons fair between a delayed acceptance (DA) MH and its corresponding MH, we compute the total time for the latter (MH) by the median time the former (DA) spends in the second stage and multiply by the number of MCMC iterations N:

$$\text{CPU}_{\text{MH}} = N \times \text{median time DA stage 2.}$$

The median is used to avoid extreme values that can arise due to external disturbances (CPU time should be independent of θ here). For the delayed acceptance algorithm, the CPU time is

$$\begin{aligned}\text{CPU}_{\text{DA}} = N &\times \text{median time DA stage 1} \\ &+ \text{FullEval} \times \text{median time DA stage 2,}\end{aligned}$$

where FullEval is the number of second-stage evaluations the delayed acceptance performs.

The (total) number of evaluations measure is straightforward for MH ($N \times n$), DA-MH without control variates ($N \times m +$ FullEval $\times\, n$) and with ($N \times (K + m) +$ FullEval $\times\, n$), and PMMH/BPMMH ($N \times (K + m)$). For the delayed versions of PMMH/BPMMH, we similarly add the different evaluations, but note that it will be different for the pre and post-training period. Moreover, there is a one time cost of learning $f(\theta)$ and also a (post-training) cost of predicting $e(\theta)$. We translate these to number of evaluations as follows. First, for learning $f(\theta)$, we measure the CPU time it takes to fit it with the training data. We compare this time to the average CPU time for the iterations during the training period. We do so because we know the measure of the latter in terms of number of evaluations: $K^{(1)} + K + m$, where $K^{(1)}$ and K are, respectively, the number of centroids in the sparser and denser set of data. Therefore, if learning $f(\theta)$ is say T times slower in CPU time, then this translates to $T \cdot (K^{(1)} + K + m)$ number of evaluations. Finally, the number of evaluations for predicting a single $e(\theta)$ depends on the model for $f(\theta)$. For linear regression,

Table 1. Delayed acceptance MH with control variates.

	Static					Dynamic				
	RED_1	RED_2	$\bar{\sigma}_R$	α_1	α_2	RED_1	RED_2	$\bar{\sigma}_R$	α_1	α_2
$K = 0.03$										
0.1	0.93	0.95	5.90	22	14	2.35	2.56	1.48	23	57
1	2.43	2.65	1.40	22	59	2.23	3.47	0.45	22	85
5	2.09	2.78	0.57	24	82	0.73	3.27	0.19	24	93
$K = 0.21$										
0.1	1.51	1.52	3.87	21	24	2.87	2.84	0.83	25	74
1	2.81	3.02	0.99	22	69	3.03	3.91	0.27	22	91
5	2.05	2.71	0.39	26	88	0.78	3.17	0.11	25	96
$K = 0.71$										
0.1	2.24	2.20	2.10	21	46	3.03	3.49	0.41	23	86
1	3.53	3.30	0.60	22	81	1.97	3.70	0.13	23	96
5	2.22	3.09	0.26	22	92	0.70	3.28	0.06	24	98
$K = 3.68$										
0.1	2.64	2.75	0.82	21	74	1.28	3.63	0.13	21	96
1	3.71	3.19	0.25	23	92	1.15	3.57	0.04	21	99
5	3.60	2.93	0.11	23	96	0.60	2.95	0.02	24	99

NOTES: The table shows some quantities for the static and dynamic implementation with different sparse representations of the data represented by K, which is the number of clusters (expressed as % of n). For each approximation, different sample sizes (0.1,1,5 in % of n) are considered. The quantities are the mean RED_1 and RED_2 in (3.9) measured with respect to computing time and average number of evaluations, respectively. Furthermore, $\bar{\sigma}_R$ is the mean (over MCMC iterations) standard deviation of $\log(R_m)$. Finally, α_1 and α_2 are the acceptance probabilities in (3.1) and (3.3) (expressed in %), where the latter is computed conditional on acceptance in the first stage. The corresponding MH algorithm has an acceptance rate of ≈23%.

the prediction is a dot product that we assign the same cost as computing a single log-likelihood contribution (which is typically a function of a dot product). For a Gaussian process, the prediction requires evaluating the kernel for N_{train} observations and we let that define the number of evaluations for a single prediction.

5.3. Implementation Details

We consider the following two examples. Example 1 estimates the model with DA-MH implemented with our efficient control variates and compares it to the implementation in Payne and Mallick (2015). Example 2 estimates the model with DA-BPMMH and compare to the block PMMH algorithm in Quiroz et al. (2016). To make comparisons fair for Example 1, we use without replacement sampling (as in Payne and Mallick 2015). This sampling scheme is typically used together with the Horvitz–Thompson estimator (Horvitz and Thompson 1952): see Särndal, Swensson, and Wretman (2003) on how to modify the formulas in Section 2.3 for without replacement sampling.

Two main implementations of the difference estimator are considered. The first computes q_k with the second-order term evaluated at β, which we call *dynamic*. The second, which we call *static*, fixes the second-order term at the optimum $\beta^\star$. The dynamic approach clearly provides a better approximation but is more expensive to compute. For both the dynamic and static approaches, we compare four different sparse representations of the data for computing q in (2.2), each with a different number of clusters. The clusters are obtained using Algorithm 1 in Quiroz et al. (2016) on the observations for which $y = 0$ (4,706,523 observations). We note that, as more clusters are used to represent the data, the approximation of the likelihood is more accurate, although it is more expensive to compute.

We consider a Random walk MH proposal for β where we learn the proposal scale during the first $N_{\text{train}} = 5000$ (and also train $f(\theta)$) iterations to reach an acceptance probability of ≈ 0.23 for MH (Roberts, Gelman, and Gilks 1997) and ≈ 0.10 for BPMMH. For the delayed acceptance algorithms, we have the same targets but for α_1, that is, the first-stage acceptance probability. We discard the training samples and also a subsequent burn-in period of 10% of the remaining samples (20,000) when doing inference. However, the computing costs (CPU and number of evaluations) include all N iterations.

Finally, the delayed acceptance algorithms are implemented with an update of u with probability 0.01.

Table 2. Delayed acceptance MH without control variates (Payne and Mallick 2015).

	RED_1	RED_2	$\bar{\sigma}_R$	α_1	α_2
0.1	0.00	0.00	101.94	21	0
1	0.36	0.37	13.81	18	3
5	1.39	1.45	3.52	22	29
50	1.13	1.33	0.63	24	80
80	0.91	1.08	0.31	24	90

NOTES: The table shows some quantities for different sample sizes (0.1,1,5,50,80, in % of n) to estimate the likelihood. The quantities are the mean RED_1 and RED_2 in (3.9) measured with respect to computing time and number of evaluations, respectively. Furthermore, $\bar{\sigma}_R$ is the mean (over MCMC iterations) standard deviation of $\log(R_m)$. Finally, α_1 and α_2 are the acceptance probabilities in (3.1) and (3.3) (expressed in %), where the latter is computed conditional on acceptance in the first stage. The corresponding MH algorithm has an acceptance rate of ≈23%.

Table 3. Delayed acceptance block PMMH.

	RED_2	α_1	α_2
BPMMH	14.03		
DA-BPMMH (GP)	25.53	9.6	95
DA-BPMMH (LR)	30.19	9.2	99

NOTES: The table shows some quantities for Block PMMH (BPMMH, Quiroz et al. 2016) and delayed acceptance BPMMH. The latter is implemented with a Gaussian process (GP) and linear regression (LR) for learning $f(\theta)$ to predict $e(\theta)$ in (4.5). The block PMMH uses a sample size of 0.5%, which corresponds to $\sigma^2 \approx 10$. The approximations used by BPMMH for computing the likelihood estimate is based on $K = 3.68$ (expressed as % of n) number of clusters. The delayed acceptance version uses an approximation based on $K^{(1)} = 0.71$ (expressed as % of n) in the first stage (and the same as BPMMH for the second stage). The quantities are the mean RED_2 in (3.9) measured with respect to number of evaluations. Moreover, α_1 and α_2 are the acceptance probabilities in (3.1) and (3.3) (expressed in %), where the latter is computed conditional on acceptance in the first stage. The corresponding BPMMH algorithm targets an acceptance rate of ≈10%, obtaining 8.4% in this run.

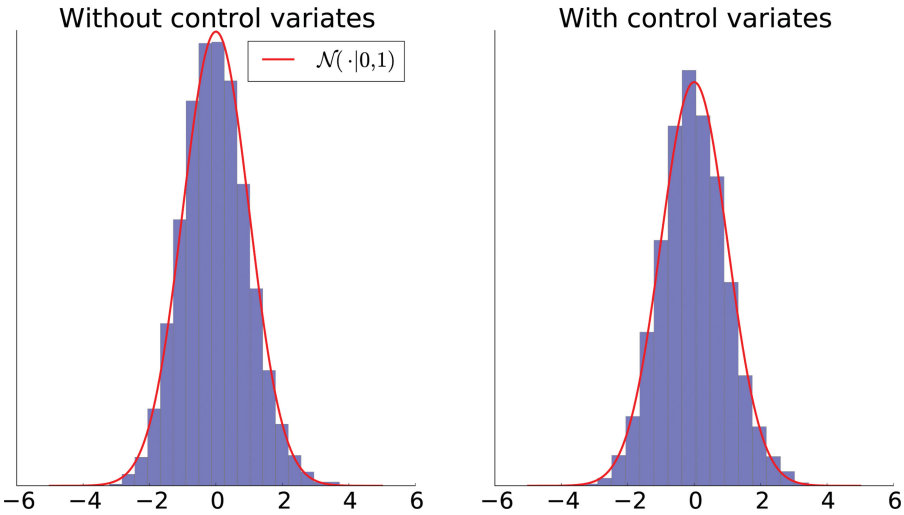

Figure 1. Addressing the normality assumption in Theorem 1. The figure shows a histogram of (standardized) $\hat{l}_{m,n}(\theta_c, \theta_p)$ in (3.4) (10,000 Monte Carlo replicates) without control variates (left, Payne and Mallick 2015) and with control variates (right). The red solid line is the density function of a standard normal variable. The validation is for the smallest value of m (0.1% of n) for both cases and for the smallest value of K (less accurate, 0.03% of n) for the control variates. The values of the parameters are $\theta_c = \theta^\star$, where $\theta^\star$ is the mode, and θ_p is sampled from the proposal distribution. We have verified the assumption for several values of θ_c and θ_p (not shown here).

5.4. Example 1: DA-MH

Tables 1 and 2 summarize the results, respectively, for the difference estimator with control variates (DE) and the estimator in Payne and Mallick (2015) (PM). It is evident that the difference estimator has a larger second-stage acceptance probability α_2 (for a given sample size), which is a consequence of Theorem 1 because it has a lower $\sigma_R^2 = V[\log(R_m)]$. Figure 1 confirms that the normality assumptions, for the smallest value of m and K (the worst case scenario), are adequate for both methods. We also note from Table 2 that for some sample sizes Payne and Mallick (2015) performed more poorly than the standard Metropolis–Hastings algorithm. One possible explanation is that the applications in Payne and Mallick (2015) have a small number of continuous covariates (one in the first application and three in the second) and the rest are binary. It is clear that the continuous covariate case results in more variation among the log-likelihood contributions, which is detrimental for SRS. In this application, we have eight continuous covariates, which explains why SRS without covariates performs poorly for small sampling fractions. As an example, for a subsample of 0.1% of the data, not a single effective sample was obtained.

5.5. Example 2: DA-BPMMH

Our second example explores how the state-dependent delayed acceptance BPMMH improves the BPMMH proposed in Quiroz et al. (2016). The results are presented in Table 3. Quiroz et al. (2016) in turn showed that they outperform other subsampling approaches and, in particular, that many of these approaches perform more poorly than the standard MH (see also Bardenet, Doucet, and Holmes 2015) in terms of efficiency and/or can give a very poor approximation of the posterior. Here, we find that the delayed acceptance BPMMH is 30 times more efficient than

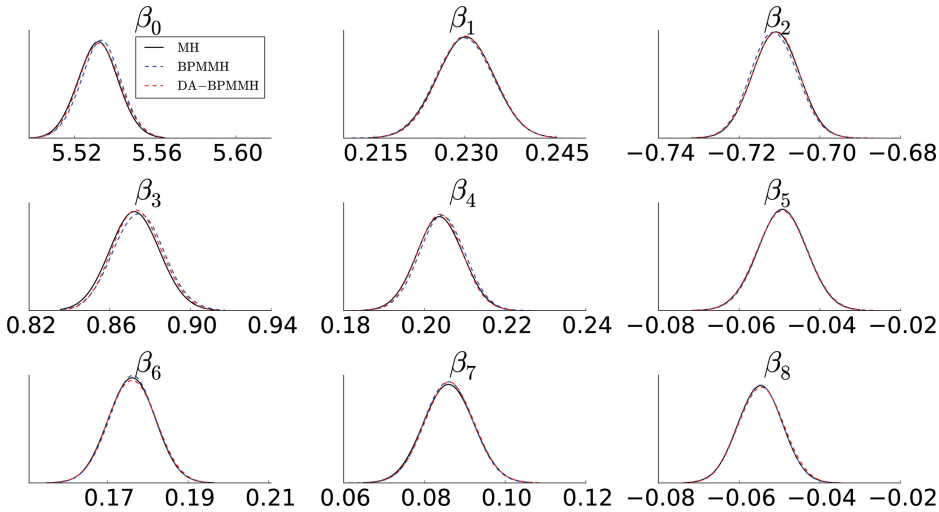

Figure 2. Kernel density estimations of marginal posteriors. The figure shows the marginal posteriors with the different algorithms MH (standard MH, solid black line), BPMMH (block PMMH, dashed blue line), and DA-BPMMH (delayed acceptance BPMMH, dashed red line). The approximations used by BPMMH for computing the likelihood estimate is based on $K = 3.68$ (expressed as % of n) number of clusters with $\sigma^2 \approx 10$. The delayed acceptance version uses an approximation based on $K^{(1)} = 0.71$ (expressed as % of n) in the first stage.

MH, which is a huge improvement considering the aforementioned facts. Moreover, we find that the approximate posterior produced is very close to the true posterior (simulated by MH), as illustrated in Figure 2.

6. Conclusions

We explore the use of the efficient and robust difference estimator in a delayed acceptance MH setting. The estimator incorporates auxiliary information about the contribution to the log-likelihood function while keeping the computational complexity low by operating on a sparse set of the data. We demonstrate that the estimator is more efficient than that of Payne and Mallick (2015) in terms of having a much lower variance. Moreover, we prove that a lower variability implies that the delayed acceptance algorithm is more efficient, as measured by the probability of accepting the second-stage conditional that the first stage is accepted. In an application to modeling of firm-bankruptcy, we find that the proposed delayed acceptance algorithm is feasible in the sense that it improves on the standard MH algorithm, which is sometimes not true for Payne and Mallick (2015). We argue that previous approaches (discussed in the introduction) of exact simulation by MCMC are either (i) only possible under unfeasible assumptions or (ii) very inefficient (compared to MH). We therefore believe that exact simulation by subsampling MCMC is only possible by a delayed acceptance approach, and the implementation provided here is crucial for success.

Next, we realize that a delayed acceptance approach has the caveat of scanning the complete data when deciding upon final acceptance. We propose a state-dependent delayed acceptance that replaces the second-stage evaluation with an estimate. This algorithm inherently allows for correlating the subsamples used for estimating the likelihood, and we can leverage on recent advances in the pseudo-marginal literature to reduce the computational cost. Moreover, we show that it is a special case of the state-dependent delayed acceptance in Christen and Fox (2005) and thus convergence to an invariant distribution follows. This distribution is perturbed because the second stage estimate is biased, but we can control the error and ensure that it is within $O(m^{-2})$ of the true posterior. We demonstrate that the approximation is very accurate and we can improve on MH by a factor of 30 in terms of a measure that balances statistical and computational efficiency.

Appendix: Proof of Theorem 1

Proof of Theorem 1. It follows from the normality assumption that $X = R_m \sim \log\mathcal{N}(0, \sigma_R^2)$ with density

$$f(x) = \frac{1}{x}\frac{1}{\sqrt{2\pi\sigma_R^2}}\exp\left(-\frac{1}{2\sigma_R^2}\log(x)^2\right), \quad x > 0.$$

The expectation of the acceptance probability $\alpha_2(\theta_c \to \theta_p)$ with respect to X is

$$\mathrm{E}[\min(1, X)] = \int_0^1 x f(x)dx + \int_1^\infty f(x)dx.$$

Since $\mathrm{median}(X) = 1$ we obtain $\int_1^\infty f(x)dx = 0.5$. Now,

$$\begin{aligned}\int_0^1 x f(x)dx &= \int_0^1 \frac{1}{\sqrt{2\pi\sigma_R^2}}\exp\left(-\frac{1}{2\sigma_R^2}\log(x)^2\right)dx \\ &= \exp\left(\sigma_R^2/2\right)\int_{-\infty}^0 \frac{1}{\sqrt{2\pi\sigma_R^2}}\exp\left(-\frac{1}{2\sigma_R^2}\left(y-\sigma_R^2\right)^2\right)dy,\end{aligned}$$

with $y = \log(x)$. The integrand is the pdf of $Y \sim \mathcal{N}(\sigma_R^2, \sigma_R^2)$ and thus

$$\mathrm{E}[\min(1, X)] = \exp\left(\sigma_R^2/2\right)(1 - \Phi(\sigma_R)) + 0.5.$$

We now show that $\mathrm{E}[\min(1, X)]$ is decreasing in σ_R. We have that

$$\frac{d}{d\sigma_R}\mathrm{E}[\min(1, X)] = \exp\left(\sigma_R^2/2\right)\left(\sigma_R - \sigma_R\Phi(\sigma_R) - \frac{1}{\sqrt{2\pi}}\right),$$

and we can (numerically) compute the maximum of the expression in brackets on the right which is ≈ -0.23. Since $\exp(\sigma_R^2/2) > 0$ it follows that $\frac{d}{d\sigma_R}\mathrm{E}[\min(1, X)] < 0$ and hence $\mathrm{E}[\alpha_2(\theta_c \to \theta_p)]$ decreases as a function of σ_R. □

Acknowledgments

Quiroz was partially supported by VINNOVA grant 2010-02635. Tran was partially supported by a Business School Pilot Research grant. Villani was partially supported by Swedish Foundation for Strategic Research (Smart Systems: RIT 15-0097). Kohn was partially supported by Australian Research Council Centre of Excellence grant CE140100049. The views expressed in this article are solely the responsibility of the authors and should not be interpreted as reflecting the views of the Executive Board of Sveriges Riksbank. The authors thank the editor, the associate editor, and the reviewers for their comments that helped to improve the manuscript.

References

Andrieu, C., and Roberts, G. O. (2009), "The Pseudo-Marginal Approach for Efficient Monte Carlo Computations," *The Annals of Statistics*, 37, 697–725. [13,17]

Banterle, M., Grazian, C., and Robert, C. P. (2014), "Accelerating Metropolis-Hastings Algorithms: Delayed Acceptance With Prefetching," *arXiv preprint arXiv:1406.2660*. [12,15]

Bardenet, R., Doucet, A., and Holmes, C. (2014), "Towards Scaling up Markov Chain Monte Carlo: An Adaptive Subsampling Approach," in *Proceedings of The 31st International Conference on Machine Learning*, pp. 405–413. [12]

——— (2015), "On Markov Chain Monte Carlo Methods for Tall Data," *arXiv preprint arXiv:1505.02827*. [12,13,14,15,20]

Beaumont, M. A. (2003), "Estimation of Population Growth or Decline in Genetically Monitored Populations," *Genetics*, 164, 1139–1160. [13]

Ceperley, D., and Dewing, M. (1999), "The Penalty Method for Random Walks With Uncertain Energies," *The Journal of Chemical Physics*, 110, 9812–9820. [14]

Christen, J. A., and Fox, C. (2005), "MCMC Using an Approximation," *Journal of Computational and Graphical Statistics*, 14, 795–810. [14,15,17,21]

Cui, T., Fox, C., and O'Sullivan, M. (2011), "Bayesian Calibration of a Large-Scale Geothermal Reservoir Model by a New Adaptive Delayed Acceptance Metropolis Hastings Algorithm," *Water Resources Research*, 47. [15,17]

Deligiannidis, G., Doucet, A., and Pitt, M. K. (2016), "The Correlated Pseudo-Marginal Method," *arXiv preprint arXiv:1511.04992v3*. [16]

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, 295–313. [17]

Fox, C., and Nicholls, G. (1997), "Sampling Conductivity Images via MCMC," in *The Art and Science of Bayesian Image Analysis*, eds. K.

Mardia, C. Gill, and R. Aykroyd, Leeds, UK: Leeds University Press, pp. 91–100. [15]

Giordani, P., Jacobson, T., Von Schedvin, E., and Villani, M. (2014), "Taking the Twists Into Account: Predicting Firm Bankruptcy Risk With Splines of Financial Ratios," *Journal of Financial and Quantitative Analysis*, 49, 1071–1099. [18]

Golightly, A., Henderson, D. A., and Sherlock, C. (2015), "Delayed Acceptance Particle MCMC for Exact Inference in Stochastic Kinetic Models," *Statistics and Computing*, 25, 1039–1055. [15,17]

Hansen, M. H., and Hurwitz, W. N. (1943), "On the Theory of Sampling From Finite Populations," *The Annals of Mathematical Statistics*, 14, 333–362. [14]

Hastings, W. K. (1970), "Monte Carlo Sampling Methods using Markov Chains and Their Applications," *Biometrika*, 57, 97–109. [12]

Horvitz, D. G., and Thompson, D. J. (1952), "A Generalization of Sampling Without Replacement From a Finite Universe," *Journal of the American Statistical Association*, 47, 663–685. [19]

Jacob, P. E., and Thiery, A. H. (2015), "On Nonnegative Unbiased Estimators," *The Annals of Statistics*, 43, 769–784. [13]

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)*, pp. 181–189. [12]

Liu, J. S. (2008), *Monte Carlo Strategies in Scientific Computing*, New York: Springer Science & Business Media. [15]

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, 443–467. [13]

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)*, pp. 543–552. [12,13]

Maire, F., Friel, N., and Alquier, P. (2015), "Light and Widely Applicable MCMC: Approximate Bayesian Inference for Large Datasets," *arXiv preprint arXiv:1503.04178*. [12]

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, 1087–1092. [12]

Minsker, S., Srivastava, S., Lin, L., and Dunson, D. (2014), "Scalable and Robust Bayesian Inference via the Median Posterior," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1656–1664. [12]

Neiswanger, W., Wang, C., and Xing, E. (2014), "Asymptotically Exact, Embarrassingly Parallel MCMC," *in Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, pp. 623–632. [12]

Nemeth, C., and Sherlock, C. (2016), "Merging MCMC Subposteriors Through Gaussian-Process Approximations," *arXiv preprint arXiv:1605.08576*. [12]

Nicholls, G. K., Fox, C., and Watt, A. M. (2012), "Coupled MCMC With a Randomized Acceptance Probability," *arXiv preprint arXiv:1205.6857*. [14]

Payne, R. D., and Mallick, B. K. (2017), "Two-Stage Metropolis-Hastings for Tall Data," *Journal of Classification*. [12,14,15,16,18,19,20,21]

Pitt, M. K., Silva, R. d. S., 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, 134–151. [17]

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006), "Coda: Convergence Diagnosis and Output Analysis for MCMC," *R News*, 6, 7–11. [16]

Quiroz, M., Tran, M.-N., Villani, M., and Kohn, R. (2017), "Exact Subsampling MCMC," *arXiv preprint arXiv:1603.08232*. [12,13]

Quiroz, M., Villani, M., Kohn, R., and Tran, M.-N. (2016), "Speeding up MCMC by Efficient Data Subsampling," *arXiv preprint arXiv:1404.4178v4*. [12,13,14,15,16,17,19,20]

Rhee, C., and Glynn, P. W. (2015), "Unbiased Estimation With Square Root Convergence for SDE Models," *Operations Research*, 63, 1026–1043. [13]

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, 110–120. [19]

Särndal, C.-E., Swensson, B., and Wretman, J. (2003), *Model Assisted Survey Sampling*, New York: Springer. [13,19]

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H., George, E., and McCulloch, R. (2013), "Bayes and Big Data: The Consensus Monte Carlo Algorithm," in *EFaBBayes 250 Conference* (Vol. 16). [12]

Sherlock, C., Golightly, A., and Henderson, D. A. (2017), "Adaptive, Delayed-Acceptance MCMC for Targets With Expensive Likelihoods," *Journal of Computational and Graphical Statistics*, 26, 434–444. [15,17]

Sherlock, C., Thiery, A., and Golightly, A. (2015a), "Efficiency of Delayed-Acceptance Random Walk Metropolis Algorithms," *arXiv preprint arXiv:1506.08155*. [15,17]

Sherlock, C., Thiery, A. H., Roberts, G. O., and Rosenthal, J. S. (2015b), "On the Efficiency of Pseudo-Marginal Random Walk Metropolis Algorithms," *The Annals of Statistics*, 43, 238–275. [17]

Smith, M. (2011), "Estimating Nonlinear Economic Models Using Surrogate Transitions," manuscript, Federal Reserve Board. Available at *https://www.economicdynamics.org/meetpapers/2012/paper_494.pdf*. [15]

Solonen, A., Ollinaho, P., Laine, M., Haario, H., Tamminen, J., Järvinen, H. (2012), "Efficient MCMC for Climate Model Parameter Estimation: Parallel Adaptive Chains and Early Rejection," *Bayesian Analysis*, 7, 715–736. [15]

Tran, M.-N., Kohn, R., Quiroz, M., and Villani, M. (2016), "Block-Wise Pseudo-Marginal Metropolis-Hastings," *arXiv preprint arXiv:1603.02485v3*. [16,17]

Wang, X., and Dunson, D. B. (2013), "Parallel MCMC via Weierstrass Sampler," *arXiv preprint arXiv:1312.4605v2*. [12]