On Markov chain Monte Carlo methods for tall data

Rémi Bardenet, Arnaud Doucet, Chris Holmes

On Markov chain Monte Carlo methods for tall data

Rémi Bardenet[1,*], Arnaud Doucet[2], Chris Holmes[2]
[1] CNRS & CRIStAL, Université de Lille, 59651 Villeneuve d'Ascq, France
[2] Department of Statistics, University of Oxford, Oxford OX1 3TG, UK

May 13, 2015

Abstract

Markov chain Monte Carlo methods are often deemed too computationally intensive to be of any practical use for big data applications, and in particular for inference on datasets containing a large number n of individual data points, also known as tall datasets. In scenarios where data are assumed independent, various approaches to scale up the Metropolis-Hastings algorithm in a Bayesian inference context have been recently proposed in machine learning and computational statistics. These approaches can be grouped into two categories: divide-and-conquer approaches and, subsampling-based algorithms. The aims of this article are as follows. First, we present a comprehensive review of the existing literature, commenting on the underlying assumptions and theoretical guarantees of each method. Second, by leveraging our understanding of these limitations, we propose an original subsampling-based approach which samples from a distribution provably close to the posterior distribution of interest, yet can require less than $\mathcal{O}(n)$ data point likelihood evaluations at each iteration for certain statistical models in favourable scenarios. Finally, we have only been able so far to propose subsampling-based methods which display good performance in scenarios where the Bernstein-von Mises approximation of the target posterior distribution is excellent. It remains an open challenge to develop such methods in scenarios where the Bernstein-von Mises approximation is poor.

Contents

*Corresponding author: remi.bardenet@gmail.com

1 Introduction

Performing inference on tall datasets, that is datasets containing a large number n of individual data points, is a major aspect of the big data challenge. Statistical models, and Bayesian methods in particular, commonly demand Markov chain Monte Carlo (MCMC) algorithms to make inference, yet running MCMC on such tall datasets is often far too computationally intensive to be of any practical use. Indeed, MCMC algorithms such as the Metropolis-Hastings

(MH) algorithm require at each iteration to sweep over the whole dataset. Frequentist or variational Bayes approaches are thus usually preferred to a fully Bayesian analysis in the tall data context on computational grounds. However, they might be difficult to put in practice or justify in scenarios where the likelihood function is complex; e.g. non-differentiable (Chernozhukov & Hong, 2003). Moreover, some applications require precise quantification of uncertainties and a full Bayesian approach might be preferable in those instances. This is the case for example for applications from experimental sciences, such as cosmology (Trotta, 2006) or genomics (Wright, 2014), were such big data problems abound. Consequently, much efforts have been devoted over recent years to develop scalable MCMC algorithms. These approaches can be broadly classified into two groups: divide-and-conquer approaches and subsampling-based algorithms. Divide-and-conquer approaches divide the initial dataset into batches, run MCMC on each batch separately, and then combine these results to obtain an approximation of the posterior: Subsampling approaches aim at reducing the number of individual data point likelihood evaluations necessary at each iteration of the MH algorithm.

After briefly reviewing the limitations of MCMC for tall data, introducing our notation and two running examples in Section 2, we first review the divide-and-conquer literature in Section 3. The rest of the paper is devoted to subsampling approaches. In Section 4, we discuss pseudo-marginal MH algorithms. These approaches are exact in the sense that they target the right posterior distribution. In Section 5, we review other exact approaches, before relaxing exactness in Section 6. Throughout, we focus on the assumptions and guarantees of each method. We also illustrate key methods on two running examples. Finally, in Section 7, we improve over our so-called confidence sampler in (Bardenet *et al.* , 2014), which samples from a controlled approximation of the target. We demonstrate these improvements yield significant reductions in computational complexity at each iteration in Section 8. In particular, our improved confidence sampler can break the $\mathcal{O}(n)$ barrier of number of individual data point likelihood evaluations per iteration in favourable cases. Its main limitation is the requirement for cheap-to-evaluate proxies for the log-likelihood, with a known error. We provide examples of such proxies relying on Taylor expansions.

All examples can be rerun or modified using the companion IPython notebook[1] to the paper, available as supplementary material.

2 Bayesian inference, MCMC, and tall data

In this section, we describe the inference problem of interest and the associated MH algorithm. We also detail the two running examples on which we benchmark key methods in Section 4, 5 and 6.

2.1 Bayesian inference

Consider a dataset

$$\mathcal{X} = \{x_1, ..., x_n\} \subset \mathbb{X} \subset \mathbb{R}^d, \tag{1}$$

[1]The IPython notebook and a static html render of it can both be found at http://www.2020science.net/research/scaling-mcmc-methods.

and a parameter space Θ. We assume the data are conditionally independent with associated likelihood $\prod_{i=1}^{n} p(x_i|\theta)$ given a parameter value θ and we denote $\ell(\theta)$ the associated average log-likelihood

$$\ell(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log p(x_i|\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(\theta). \tag{2}$$

We follow a Bayesian approach where one assigns a prior $p(\theta)$ to the unknown parameter, so that inference relies on the posterior distribution

$$\pi(\theta) = p(\theta|x) \propto \gamma(\theta) \triangleq p(\theta)e^{n\ell(\theta)}, \tag{3}$$

where γ denotes an unnormalized version of π. In most applications, π is intractable and we will focus here on Markov chain Monte Carlo methods (MCMC; Robert & Casella, 2004) and, in particular, on the Metropolis-Hastings (MH) algorithm to approximate it.

2.2 The Metropolis-Hastings algorithm

A standard approach to sample approximately from $\pi(\theta)$ is to use MCMC algorithms. To illustrate the limitation of MCMC in the tall data context, we focus here on the MH algorithm (Robert & Casella, 2004, Chapter 7.3). The MH algorithm simulates a Markov chain $(\theta_k)_{k\geq 0}$ of invariant distribution π. Then, under weak assumptions, see e.g. (Douc *et al.* , 2014, Theorem 7.32), the following central limit theorem holds for suitable test functions h

$$\sqrt{N_{\text{iter}}}\left[\frac{1}{N_{\text{iter}}}\sum_{k=0}^{N_{\text{iter}}} h(\theta_k) - \int h(\theta)\,\pi(\theta)d\theta\right] \to \mathcal{N}(0, \sigma^2_{\text{lim}}(h)), \tag{4}$$

where convergence is in distribution.

MH$\big(\gamma(\cdot),\, q(\cdot|\cdot),\, \theta_0,\, N_{\text{iter}}\big)$

for $k \leftarrow 1$ **to** N_{iter}
  $\theta \leftarrow \theta_{k-1}$,
  $\theta' \sim q(.|\theta)$,
  $u \sim \mathcal{U}_{(0,1)}$
  $\alpha(\theta, \theta') \leftarrow \frac{\gamma(\theta')}{\gamma(\theta)} \times \frac{q(\theta|\theta')}{q(\theta'|\theta)}$
  if $u < \alpha(\theta, \theta')$
    $\theta_k \leftarrow \theta'$ ▷ *Accept*
  else $\theta_k \leftarrow \theta$ ▷ *Reject*
return $(\theta_k)_{k=1,\ldots,N_{\text{iter}}}$

Figure 1: The pseudocode of the MH algorithm targeting the distribution π. Note that π is only involved in ratios, so that one only needs to know an unnormalized version γ of π.

The pseudocode of MH targeting a generic distribution π is given in Figure 1. In the case of Bayesian inference with independent data (3), Step 5 is equivalent to setting

$$\log \alpha(\theta, \theta') \quad = \quad \log \left[\frac{p\left(\theta'\right)}{p\left(\theta\right)} \frac{q(\theta|\theta')}{q(\theta'|\theta)} \right] + n \left[\ell(\theta') - \ell(\theta) \right]. \tag{5}$$

When the dataset is tall ($n \gg 1$), evaluating the log likelihood ratio in (5) is too costly an operation and rules out the applicability of such a method. As we shall see, two possible options are to either divide the dataset into tractable batches, or approximate the acceptance ratio in (5) using only part of the dataset.

2.3 Running examples

We will evaluate some of the described approaches on two illustrative simple running examples. We fit a one-dimensional normal distribution $p(\cdot|\mu, \sigma) = \mathcal{N}(\cdot|\mu, \sigma^2)$ to 10^5 i.i.d. points drawn according to $X_i \sim \mathcal{N}(0, 1)$ and lognormal observations $X_i \sim \log\mathcal{N}(0, 1)$, respectively. The latter example illustrates a misspecification of the model. We assign a flat prior $p(\mu, \log\sigma) \propto 1$. For all algorithms, we start the chain at the maximum a posteriori (MAP) estimate. The MH proposal is an isotropic Gaussian random walk, whose stepsize is first set proportional to $1/\sqrt{n}$ and then adapted during the first $1\,000$ iterations so as to reach 50% acceptance. When applicable, we also display the number of likelihood evaluations per iteration, and compare it to the n evaluations required at each iteration by the MH algorithm.

In Figure 2, we illustrate the results of $10\,000$ iterations of vanilla MH on each of the two datasets. MH does well, as the posterior coincides with that of a longer reference run of $50\,000$ iterations in each case, and the autocorrelations show a fast exponential decrease. The Bernstein-von Mises approximation (van der Vaart, 2000, Chapter 10.2), a Gaussian centered at the true value, with covariance minus the scaled inverse Fisher information, is a very good approximation to the posterior in both cases. We are thus in simple cases of heavy concentration of the posterior, where subsampling should help a lot if it is to be of any help in tackling tall data problems.

3 Divide-and-conquer approaches

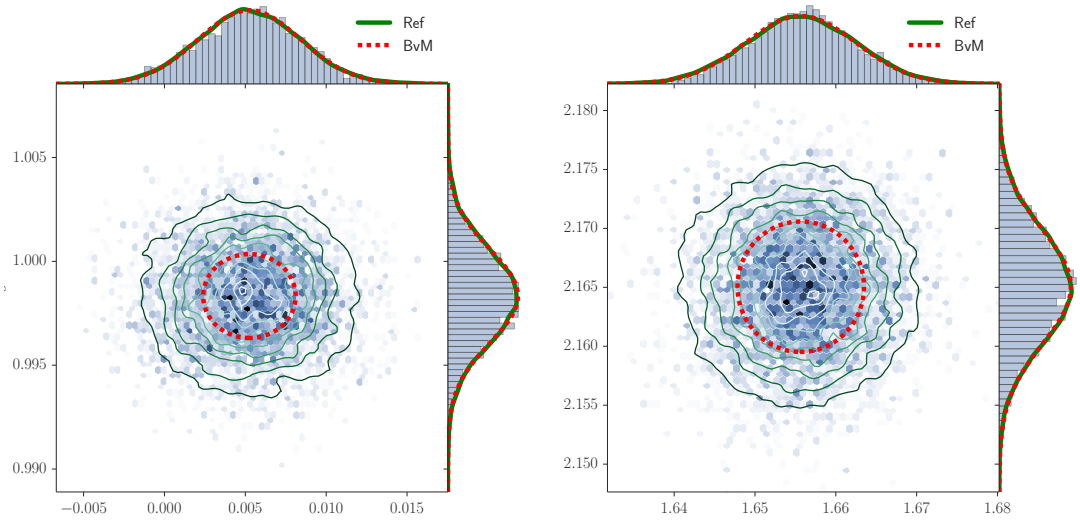
A natural way to tackle tall data problems is to divide the data into batches, run MH on each batch separately, and then combine the results.

3.1 Representing the posterior as a combination of batch posteriors

Assume data $\mathcal{X}$ are divided in B batches $\mathbf{x}_1, \ldots, \mathbf{x}_B$. Relying on the equality

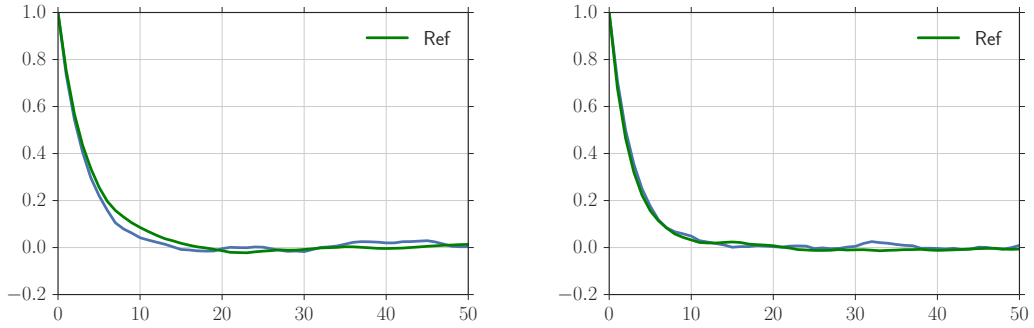
$$p(\theta|\mathcal{X}) \propto \prod_{i=1}^{B} p(\theta)^{1/B} p(\mathbf{x}_i|\theta), \tag{6}$$

Huang & Gelman (2005) propose to combine the batch posterior approximations using Gaussian approximations or importance sampling. Scott *et al.* (2013) propose to average samples



(a) Chain histograms, $X_i \sim \mathcal{N}(0, 1)$

(b) Chain histograms, $X_i \sim \log \mathcal{N}(0, 1)$



(c) Autocorr. of $\log \sigma$, $X_i \sim \mathcal{N}(0, 1)$

(d) Autocorr. of $\log \sigma$, $X_i \sim \log \mathcal{N}(0, 1)$

Figure 2: Results of 10 000 iterations of vanilla MH fitting a Gaussian model to one-dimensional Gaussian and lognormal synthetic data, on the left and right panel, respectively. Figures 2(a) and 2(b) show the chain histograms, joint and marginals; the x-axis corresponds to the mean of the fitted Gaussian, the y-axis to the standard deviation. We have superimposed a kernel density estimator of a long MH chain for reference in green and the Bernstein-von Mises Gaussian approximation in red. Figures 2(c) and 2(d) show the marginal autocorrelation of $\log \sigma$ in blue. The green curves are baselines that correspond to the long MH reference run depicted in green in the top panel; although the green autocorrelation functions are of limited interest when comparing them to vanilla MH, we use them as reference in all similar later figures.

across batches, noting this is exact under Gaussian assumptions. Neiswanger *et al.* (2014) propose to run an MCMC chain on each batch $\mathbf{x}_i$ targeting an artificial batch posterior

$$\pi_i(\theta) \propto p(\theta)^{1/B} p(\mathbf{x}_i|\theta),$$

fit a smooth approximation to each batch posterior, and multiply them. These methods are however theoretically justified only when batch posteriors are Gaussian, or when the size *of each batch* goes to infinity, to guarantee that the used smooth approximation of each batch posterior is accurate.

There are few results available on how the properties of combined estimators scale with the number of batches B. Neiswanger *et al.* (2014) fit a kernel density estimator to the samples of each batchwise chain, and multiply the resulting kernel density estimators. A sample from this mixture approximation to π is then obtained through an additional MCMC step. Under simplifying assumptions (all MCMC chains are assumed being independent draws from their targets, for example), a bound on the MSE of the final estimator is obtained. However, this bound explodes as the kernel bandwidth goes to zero, and more importantly, it is exponential in the number of batches B. In a tall data context, the number of batches is expected to grow with n to ensure that the size of each batch is less than $\mathcal{O}(n)$. Thus, the proposed bound is currently not informative for tall data.

As pointed out by Wang & Dunson (2013), if the supports of the π_i are almost disjoint, then the product of their approximations will be a poor approximation to π. To improve the overlap between the approximations of the π_i's, Wang & Dunson (2013) propose to replace the posterior in (6) by the product of the Weierstrass transforms of each batch posterior. When the approximation of π_i is an empirical measure, its Weierstrass transform corresponds to a kernel density estimator. The product of the Weierstrass transforms can be interpreted as the marginal distribution of an extended distribution on Θ^{B+1}, where the first B copies of θ are associated to the B batches, and the remaining copy is conditionally Gaussian around a weighted mean of the first B copies. Unfortunately, sampling from the posterior of this artificial model is difficult when one only has access to approximate samples of each π_i.

Although it is not strictly speaking a Monte Carlo method, we note that Xu *et al.* (2014) and Gelman *et al.* (2014) propose an expectation-propagation-like algorithm that similarly tackles the issue of disjoint approximate batch posterior supports. Each batch of data points is represented by its individual likelihood times a *cavity* distribution. The cavity distribution is itself the product of the prior and a number of terms that represent the contributions of other batches to the likelihood. The algorithm iterates between 1) simulating from each batchwise likelihood times a batch-specific cavity distribution, and 2) fitting each batch-specific cavity component. Again, while these approaches are computationally feasible and appear to perform well experimentally, it is difficult to assert the characteristics of the proposed approximation of the posterior and there is no convergence guarantee available for this iterative algorithm.

3.2 Replacing the posterior by a geometric combination of batch posteriors

Another avenue of research consists in avoiding multiplying the batch posteriors by replacing the target by a different combination of the latter.

By introducing a suitable metric on the space of probability measures such as the Wasserstein metric, it is for example possible to define the barycenter or the median of a set of probability measures. Minsker *et al.* (2014) propose to output the median of the batch posteriors, while Srivastava *et al.* (2014) use the Wasserstein barycenter, which can be computed efficiently in practice using the techniques developed by Cuturi & Doucet (2014). While this idea has some appeal, the statistical meaning of these median or mean measures is unclear, and the robustness of the median estimate advocated in (Minsker *et al.* , 2014) may also be a drawback, as in some circumstances valuable information contained in some batches may be lost.

To conclude, divide-and-conquer techniques appear as a natural approach to handle tall data. However, the crux is how to efficiently combine the batch posterior approximations. The main issues are that the batch posterior approximations potentially have disjoint supports, that the multiplicative structure of the posterior (3) leads to poor scaling with the number of batches, that theoretical guarantees are often asymptotic in the batch size, and that cheap-to-sample combinations of batch posteriors are difficult to interpret.

4 Exact subsampling approaches: Pseudo-marginal MH

Pseudo-marginal MH (Lin *et al.* , 2000; Beaumont, 2003; Andrieu & Roberts, 2009) is a variant of MH, which relies on unbiased estimators of an unnormalized version of the target. Pseudo-marginal MH is useful to help understand several potential approaches to scale up MCMC. We start by describing pseudo-marginal MH in Section 4.1. Then, we present two pseudo-marginal approaches to tall data in Section 4.2 and Section 4.3.

4.1 Pseudo-marginal Metropolis-Hastings

Assume that instead of being able to evaluate $\gamma(\theta)$, we have access to an unbiased, almost-surely *non-negative* estimator $\hat{\gamma}(\theta)$ of the unnormalized target $\gamma(\theta)$. Pseudo-marginal MH substitutes a realization of $\hat{\gamma}(\theta')$ to $\gamma(\theta')$ in Step 5. Similarly, it replaces $\gamma(\theta)$ in Step 5 by the realization of $\hat{\gamma}(\theta)$ that was computed when the parameter value θ was last proposed. Pseudo-marginal MH is of considerable practical importance, with applications such as particle marginal MH (Andrieu *et al.* , 2010) and MCMC versions of the approximate Bayesian computation paradigm (Marin *et al.* , 2012). It is thus worth investigating its use in the context of tall data problems.

The possibility to use an unbiased estimator of γ comes at a price: first, the asymptotic variance σ^2_{lim} in (4) of an MCMC estimator based on a pseudo-marginal chain will always be larger than that of an estimator based on the underlying "marginal" MH (Andrieu & Vihola, 2015). Second, the qualitative properties of the underlying MH may not be preserved, meaning that the rate of convergence to the invariant distribution may go from geometric to subgeometric, for instance; see Andrieu & Roberts (2009) and Andrieu & Vihola (2015) for a detailed discussion. In practice, if the variance of $\hat{\gamma}(\vartheta)$ is large for some value $\vartheta \in \Theta$, then an MH move to ϑ might be accepted while $\hat{\gamma}(\vartheta)$ largely overestimates $\gamma(\vartheta)$. In that case, it is difficult for the chain to leave ϑ, and pseudo-marginal MH chains thus tend to get stuck if the variance of the involved estimators is not controlled. When some tunable parameter allows to control this variance, Doucet *et al.* (2015) show that, in order to minimize the variance of MCMC estimates for a fixed com-

putational complexity, the variance of the log-likelihood estimator should be kept around 1.0 when the ideal MH having access to the exact likelihood generates quasi-i.i.d samples from π; or set to around 3.0 when it exhibits very large integrated autocorrelation times. In practice, the integrated autocorrelation times of averages under the ideal MH are unknown as this algorithm cannot be implemented. In this common scenario, Doucet *et al.* (2015) recommend keeping the variance around 1.5 as this is a value which ensures a small penalty in performance even in scenarios where 1.0 or 3.0 are actually optimal. They also show that the penalty incurred for having a variance too small (i.e. inferior to 0.2) or too large (i.e. superior to 10) is very large. When mentioning pseudo-marginal MH algorithms, we will thus comment on the variance of the logarithm of the involved estimators $\hat{\gamma}(\theta)$, or, if not available, of their relative variance.

4.2 Unbiased estimation of the likelihood using unbiased estimates of the log-likelihood

As described in Section 4.1, pseudo-marginal MH requires an almost-surely nonnegative unbiased estimator $\hat{\gamma}(\theta)$ of the unnormalized posterior at θ, for any θ in Θ. It is easy to check that, by sampling $x_1^*, \ldots, x_t^*$ from the dataset $\mathcal{X}$ with or without replacement, we obtain the following unbiased estimator of the log-likelihood $n\ell(\theta)$

$$n\hat{\ell}(\theta) = \frac{n}{t}\sum_{i=1}^{t} \log p(x_i^*|\theta). \tag{7}$$

We denote by $\hat{\ell}(\theta)$ the subsampling estimate of the average log-likelihood and denote by $\sigma_t(\theta)^2$ its variance. Obviously, exponentiating (7) does not provide an unbiased estimate of the likelihood $e^{n\ell(\theta)}$. However, an interesting question is whether one can design a procedure which outputs an unbiased, almost-surely nonnegative estimate of $e^{n\ell(\theta)}$ using unbiased estimates of $n\ell(\theta)$ such as $n\hat{\ell}(\theta)$. Without making any further assumption about $n\hat{\ell}(\theta)$, it was recently shown by Jacob & Thiery (2013) that it is not possible. However, this can be done if one further assumes, for instance, that there exists $a(\theta)$ such that $\ell_i(\theta) > a(\theta)$ for all i, see (Jacob & Thiery, 2013, Section 3.1) who rely on a technique by Rhee & Glynn (2013) generalizing (Bhanot & Kennedy, 1985). Unfortunately, as we shall see, the resulting estimator $\hat{\gamma}(\theta)$ typically has a very large relative variance, resulting in very poor performance of the associated pseudo-marginal chain.

We apply (Rhee & Glynn, 2013, Theorem 1) to build an unbiased non-negative estimator of $\gamma(\theta)/p(\theta)$, which is equivalent to defining $\hat{\gamma}(\theta)$. For $j \geq 1$, let

$$D_j^* = \frac{n}{t}\sum_{i=1}^{t} \log p(x_{i,j}^*|\theta) - na(\theta), \tag{8}$$

be an unbiased estimator of $n\left(\ell(\theta) - a(\theta)\right)$, where the $x_{i,j}^*$'s are drawn with replacement from $\mathcal{X}$ for each i, and are further independent across j. In (Jacob & Thiery, 2013, Section 3.1), N is an integer-valued random variable whose tails do not decrease too fast, in the sense that $\mathbb{P}(N \geq k) \geq C(1+\epsilon)^{-k}$. To ease computations, we take N to be geometric with parameter

$\epsilon/(1+\epsilon)$. This corresponds to the lightest tails allowed by (Jacob & Thiery, 2013, Section 3.1), since $\mathbb{P}(N \geq k) = (1+\epsilon)^{-k}$. Finally, let

$$Y \triangleq e^{na(\theta)} \left[1 + \sum_{k=1}^{N} \frac{1}{\mathbb{P}(N \geq k)} \frac{1}{k!} \prod_{j=1}^{k} D_j^* \right]. \tag{9}$$

By (Rhee & Glynn, 2013, Theorem 1), Y is a non-negative unbiased estimator of the likelihood $e^{n\ell(\theta)}$. As mentioned in Section 4.1, it is crucial, if we want to plug $\hat{\gamma}(\theta) = Y \times p(\theta)$ in a pseudo-marginal algorithm, to control the variance of its logarithm. The variance of $\log Y$ is difficult to compute, so we use here the relative variance of Y as a proxy.

Proposition 4.1 *Let $\theta \in \Theta$ and Y be the almost surely non-negative estimator of $e^{n\ell(\theta)}$ defined in (9). Then its relative variance satisfies*

$$\frac{\mathrm{Var} Y}{e^{2n\ell(\theta)}} \geq \frac{e^{-2n(\ell(\theta)-a(\theta))+2n\sqrt{(1+\epsilon)[\sigma_t(\theta)^2+(\ell(\theta)-a(\theta))^2]}}}{n\sqrt{(1+\epsilon)[\sigma_t(\theta)^2+(\ell(\theta)-a(\theta))^2]}} + \mathcal{O}(1). \tag{10}$$

The proof of Proposition 4.1 can be found in Appendix A. We can interpret (10) as follows: in order for the relative variance of Y not to increase exponentially with n, it is necessary that $n\sigma_t(\theta)$ is of order 1. But $\sigma_t(\theta)$ if of order $t^{-1/2}$, so that the batchsize t would have to be of order n^2, which is impractical. It is also necessary that $\sqrt{1+\epsilon}$ is of order $1+n^{-1}$ to control the term in $(\ell(\theta) - a(\theta))$. This means that ϵ should be taken of order n^{-1}, but then the mean $(1+\epsilon)\epsilon^{-1}$ of the geometric variable N will be of order n. This entails that the number of terms in the randomly truncated series (9) should be of order n, which defeats the purpose of using this estimator.

Hence for the reasons outlined in Section 4.1, we expect the pseudo-marginal MH relying on Y to be highly inefficient. Indeed, we have not been able to obtain reasonably mixing chains even on our Gaussian running example. We have experimented with various choices of ϵ, and with various values of t, but none yielded satisfactory results. We conclude that this approach is not a viable solution to MH for tall data.

We note that Strathmann *et al.* (2015) have recently proposed a different way to exploit the methodology of Rhee & Glynn (2013) in the context of tall data. However, their methodology does not provide unbiased estimates of the posterior expectations of interest. It only provides unbiased estimates of some biased MCMC estimates of these expectations, these MCMC estimates corresponding indeed to running an MCMC kernel on the whole dataset for a finite number of iterations. Strathmann *et al.* (2015) suggest that it might be possible to combine their algorithm with the recent scheme of Glynn & Rhee (2014) to obtain unbiased estimates of the posterior expectations. It is yet unclear whether this could be achieved under realistic assumptions on the MCMC kernel.

4.3 Building $\hat{\gamma}(\theta)$ with auxiliary variables

In (MacLaurin & Adams, 2014), the authors propose an alternative MCMC to sample from π which, similarly to the method described previously, only requires evaluating the likelihood of

a subset of the data at each iteration. Assume a bound $\ell_i(\theta) \geq b_i(\theta)$ is available for each i and θ. For simplicity, we further assume that $b_i(\theta) = b(\theta, x_i)$ only depends on i through x_i. This is the case in the experiments of (MacLaurin & Adams, 2014), as well as ours. Note also that in Section 4.2, we used a bound that was uniform in the data index i; we could have used similarly a non-uniform bound, but this would have made the derivation of Proposition 4.1 unnecessarily heavy.

As noted in (MacLaurin & Adams, 2014), we can then define the following extended target $\tilde{\pi}$ distribution on $\Theta \times \{0,1\}^n$

$$\begin{aligned} \tilde{\pi}(\theta, z) &\propto p(\theta) \prod_{i=1}^{n} \left[\exp(\ell_i(\theta) - \exp(b_i(\theta))\right]^{z_i} \exp(b_i(\theta))^{1-z_i} \\ &= p(\theta) \prod_{i=1}^{n} \exp(b_i(\theta)) \prod_{i=1}^{n} \left[\exp(\ell_i(\theta) - b_i(\theta)) - 1\right]^{z_i}. \end{aligned} \tag{11}$$

This distribution satisfies two important features: it admits $\pi(\theta)$ as a marginal distribution, and its pointwise evaluation only requires to evaluate $\ell_i(\theta)$ for those i's for which $z_i = 1$. Note that evaluating $\tilde{\pi}(\theta, z)$ however requires to evaluate $\prod_{i=1}^{n} \exp(b_i(\theta))$, and the bounds $b_i(\theta)$ thus must be chosen so that this computation is cheap. This is the case for the lower bound of the logistic regression log-likelihood model discussed in (MacLaurin & Adams, 2014), which is a quadratic form in $t_i \theta^T x_i$, where t_i is the ± 1 label of datum x_i. The idea of replacing the evaluation of the target by a Bernoulli draw and the evaluation of a lower bound has been exploited previously; see e.g. (Mak, 2005).

Any MCMC sampler could be used to sample from $\tilde{\pi}$. MacLaurin & Adams (2014) propose an MH-within-Gibbs sampler that leverages the known conditional $\tilde{\pi}(z|\theta)$. The expected cost of one conditional MH iteration on θ at equilibrium, that is the average number of indices i such that $z_i = 1$, is $\mathcal{O}(n)$, and the constant is related to the expected relative tightness of the bound, see (MacLaurin & Adams, 2014, Section 3.1). The number of likelihood evaluations for an update of z conditional on θ is explicitly controlled in (MacLaurin & Adams, 2014) by either specifying a maximum number of attempted flips, or implicitly specifying the fraction of flips to 1.

The authors of MacLaurin & Adams (2014) remarked that their methodology is related to pseudo-marginal techniques but did not elaborate. We show here how it is indeed possible to exploit the extended target distribution $\tilde{\pi}$ in (11) to obtain an unbiased estimate of an unnormalized version of π. More precisely, we have

$$p(x_i|\theta) = \sum_{z_i \in \{0,1\}} p\left(x_i, z_i|\theta\right)$$

where $p\left(z_i|\theta, x_i\right) = \{1 - \exp(b_i(\theta) - \ell_i(\theta))\}^{z_i} \exp(b_i(\theta) - \ell_i(\theta))^{1-z_i}$. Hence, the marginal distribution of z_i under this extended model is given by

$$\begin{aligned} p(z_i = 1|\theta) &= \int p(z_i = 1, x_i|\theta) dx_i \\ &= \int \left[\exp(\ell_i(\theta)) - \exp(b_i(\theta))\right] dx_i \\ &= 1 - I_\theta, \end{aligned} \tag{12}$$

where $I_\theta \triangleq \int \exp(b(\theta, x))dx$. Using Bayes' theorem, we obtain accordingly

$$p(x_i|\theta, z_i = 1) = \frac{\exp(\ell_i(\theta)) - \exp(b_i(\theta))}{1 - I_\theta},\ p(x_i|\theta, z_i = 0) = \frac{\exp(b_i(\theta))}{I_\theta}.$$

An obvious unbiased estimator of the unnormalized posterior is thus given by

$$\hat{\gamma}(\theta) = p(\theta) \prod_{i=1}^{n} p(x_i|\theta, z_i) \tag{13}$$

where each z_i is drawn independently given θ from (12). Note that in the case of logistic regression, if $b_i(\theta)$ is chosen to be the quadratic lower bound given in (MacLaurin & Adams, 2014), its integral I_θ is a Gaussian integral and can thus be computed. Finally, similarly to the Firefly algorithm of MacLaurin & Adams (2014), the number of evaluations of the likelihood per iteration is nI_θ, loosely speaking.

Although the pseudo-marginal variant of Firefly we propose has the disadvantage of requiring the integrals I_θ to be tractable, it comes with two advantages. First, the sampling of z does not require to evaluate the likelihood at all. If computing all bounds does not become a bottleneck, this avoids the need to explicitly state a resampling fraction at the risk of augmenting the variance of the likelihood estimator. Second, the properties of this variant are easier to understand, as it is a 'standard' pseudo-marginal MH and hence the results from Section 4.1 apply. In particular, although it has the correct target distribution, the asymptotic variance of ergodic averages is inflated compared to the ideal algorithm.

As explained in Section 4.1, we consider the variance of the log likelihood estimator.

Proposition 4.2 *Let $\theta \in \Theta$. With the notations introduced in Section 4.3,*

$$\mathrm{Var}_z \left[\sum_{i=1}^{n} \log p(x_i|\theta, z_i) \right] = I_\theta(1 - I_\theta) \sum_{i=1}^{n} \log^2 \left[\frac{I_\theta}{1 - I_\theta} \left(e^{\ell_i(\theta) - b_i(\theta)} - 1 \right) \right] \tag{14}$$

The proof of Proposition 4.2 can be found in Appendix B. Proposition 4.2 can be interpreted as follows: the variance is related to how tight the bound is. In general, obtaining a variance of order 1 will only be possible if *most* bounds $b_i(\theta)$ are very tight, and the bigger n, the tighter the bounds have to be. These conditions will typically not be met when a fixed fraction of "outlier" x_i's correspond to untight bounds.

We give the results of the original Firefly MH on our running Gaussian and log normal examples in Figure 3. We bound each $\ell_i(\theta)$ using a 2nd order Taylor expansion at the MLE and the Taylor-Lagrange inequality, see Section 7.2.1 for further details. This bound is very tight in both cases, so that we are in the favourable case where only a few components of z are 1 at each iteration, and the number of likelihood evaluations per full joint iteration is thus roughly the fraction of points for which z_i has been resampled. We chose the fraction of resampled points to be 10% here, and initialized z to have 10% of ones. Trying smaller fractions led to very slowly mixing chains even for the Gaussian example. Estimating the number of likelihood evaluations per full joint iteration as the sum of the number of resampled z_i's and the number of "bright" points, we obtained in both the Gaussian and lognormal case an almost constant

number of likelihood evaluations close to 10%, so that only a few points are bright. This can be explained by the tightness of the Taylor bound, which leads Firefly MH to almost exclusively replace the evaluation of the likelihood by that of the Taylor bound. Finally, unlike the other algorithms we applied, we observed that a bad choice of the initial value of z can easily take θ out of the posterior mode. To be fair, we thus discarded the first $1\,000$ iterations as a burn-in before plotting.

As expected, the algorithm behaves erratically in the lognormal case, as failure to attempt a flip of each z_i draws the μ-component of the chain towards the few large values of $(x_i - \mu)^2$ which are bright. Since the bright points are rarely updated, the chain mixes very slowly.

5 Other exact approaches

Other exact approaches have been proposed, which do not rely on pseudo-marginal MH.

5.1 Forgetting about acceptance: stochastic approximation approaches

Welling & Teh (2011) proposed an algorithm based on stochastic gradient Langevin dynamics (SGLD). This is an iterative algorithm which at iteration $k+1$ uses the following update rule

$$\theta_{k+1} = \theta_k + \frac{\epsilon_{k+1}}{2}\left[\nabla \log p(\theta) + \frac{n}{t}\sum_{i=1}^{t} \nabla \log p(x^*_{i,k}|\theta)\right] + \sqrt{\epsilon_{k+1}}\eta_{k+1}, \tag{15}$$

(ϵ_k) is a sequence of time steps, (η_k) are independent $\mathcal{N}(0, I_d)$ vectors and

$$\frac{n}{t}\sum_{i=1}^{t} \nabla \log p(x^*_{i,k}|\theta)$$

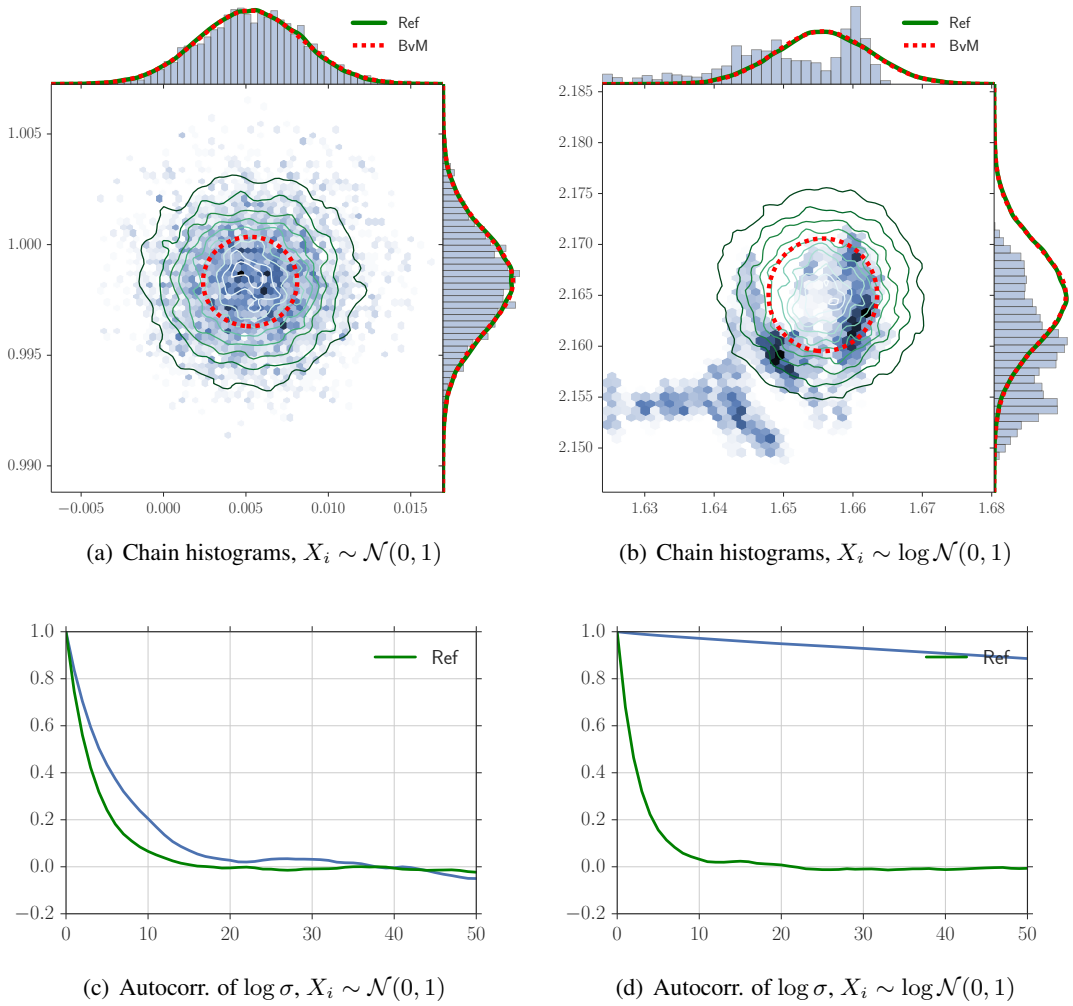
is an unbiased estimate of the score computed at each iteration using a random subsample $\left\{x^*_{i,k}\right\}$ of the observations. This approach is reminiscent of the Metropolis-adjusted Langevin algorithm (MALA; Robert & Casella 2004, Section 7.8.5), where the proposal given by

$$\theta' = \theta + \frac{\epsilon}{2}\left[\nabla \log p(\theta) + \sum_{i=1}^{n} \nabla \log p(x_i|\theta)\right] + \sqrt{\epsilon}\eta,$$

is used in an MH acceptance step, where $\epsilon \sim \mathcal{N}(0, I_d)$. The point of Welling & Teh (2011) is that if one suppresses the MH acceptance step, computes an unbiased estimate of the score but introduces a sequence of stepsizes (ϵ_k) that decreases to zero at the right rate, then

$$\left(\sum_{k=0}^{N_{\text{iter}}} \epsilon_k\right)^{-1} \sum_{k=0}^{N_{\text{iter}}} \epsilon_k \delta_{\theta_k}$$

is an approximation to π. The algorithm has been analyzed recently in (Teh *et al.* , 2014), where it has been established that it provides indeed a consistent estimate of the target. Additionally,



(a) Chain histograms, $X_i \sim \mathcal{N}(0, 1)$

(b) Chain histograms, $X_i \sim \log \mathcal{N}(0, 1)$

(c) Autocorr. of $\log \sigma$, $X_i \sim \mathcal{N}(0, 1)$

(d) Autocorr. of $\log \sigma$, $X_i \sim \log \mathcal{N}(0, 1)$

Figure 3: Results of 10 000 iterations of Firefly MH (MacLaurin & Adams, 2014) on our Gaussian and lognormal running examples. See Section 4.3 and the caption of Figure 2 for details.

a central limit theorem holds with convergence rate $N_{\text{iter}}^{-1/3}$, which is slower than the traditional Monte Carlo rate $N_{\text{iter}}^{-1/2}$. It is yet unclear how SGLD compares to other subsampling schemes in theory: it may require a smaller fraction of the dataset per iteration, but more iterations are needed to reach the same accuracy.

In practice, we show the results of SGLD on our two running examples in Figure 5.1. The stepsize ϵ_k is chosen proportional to $k^{-1/3}$, following the recommendations of Teh *et al.* (2014). We show the results of two choices for the subsample size t: 10% and 1% of the data, with respectively $10\,000$ and $100\,000$ iterations, so that both runs amount to the same 10% fraction of the budget of the vanilla MH in Figure 2. Both runs are still far from convergence on the lognormal example: subsampling draws the chain away from the support of the posterior, and one has to wait for smaller stepsizes to avoid overconfident moves. But then, the variance of the final estimate gets bigger. Constant stepsizes lead to comparable results (not shown).

Finally, we note that subsampling for Hamiltonian Monte Carlo (HMC; Duane *et al.* , 1987) has also been recently considered. Chen *et al.* (2014) propose a modification of HMC that is inspired by the SGLD with decreasing stepsize of Welling & Teh (2011), while Betancourt (2014) explores why naive approaches suffer from unacceptable biases. The algorithm of Chen *et al.* (2014) is however a heuristic, which further relies on the subsampling noise being Gaussian. As demonstrated in (Bardenet *et al.* , 2014) and in Section 6.2, relying on a Gaussian noise assumption can yield arbitrarily poor performance when this assumption is violated.

5.2 Delayed acceptance

Banterle *et al.* (2015) remarked that if we decompose the acceptance ratio in a product of positive functions

$$\alpha(\theta, \theta') = \prod_{i=1}^{B} \rho_i(\theta, \theta')$$

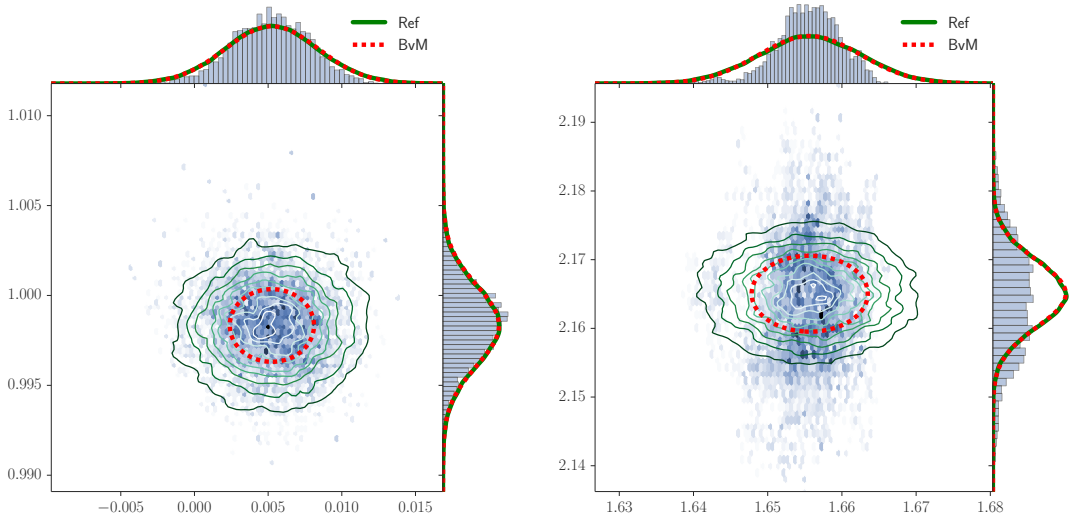
then the MH-like algorithm that accepts the move from θ to θ' with probability

$$\prod_{i=1}^{B} \min\left[\rho_i(\theta, \theta'), 1\right]$$

still admits π as invariant distribution. In practice, in the case of tall data, we can divide the dataset into B batches and use for example

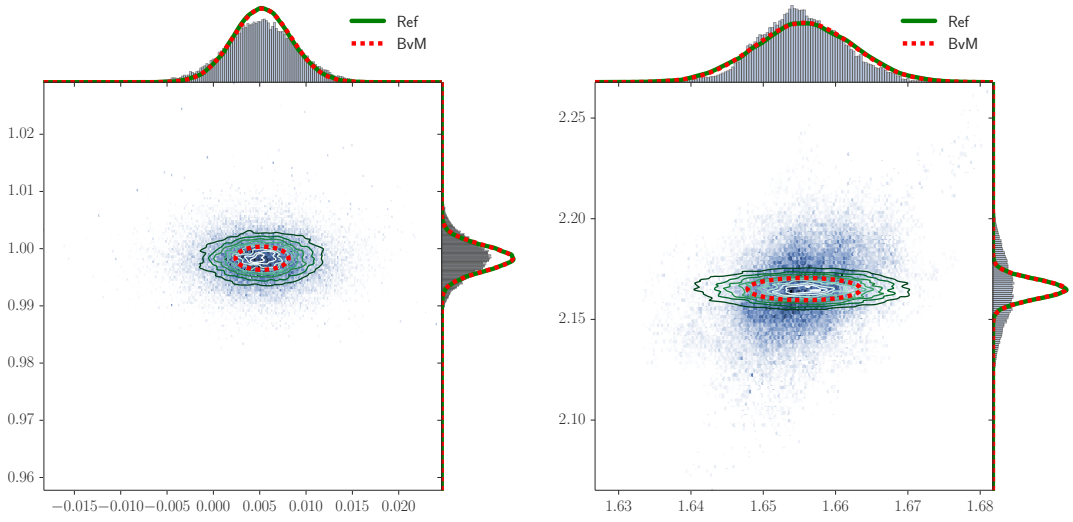
$$\rho_i\left(\theta, \theta'\right) = \frac{p(\theta')^{1/B} p(\mathbf{x}_i|\theta') q(\theta|\theta')}{p(\theta)^{1/B} p(\mathbf{x}_i|\theta) q(\theta'|\theta')}.$$

This allows us to reject candidate θ' without having to compute the full likelihoods and the calculations of $\rho_i(\theta, \theta')$ can be done in parallel. However, as remarked by Banterle *et al.* (2015), the resulting Markov chain has a larger asymptotic variance σ_{lim}^2 in (4) than the original MH, and it does not necessarily inherit the ergodicity of the original MH. Furthermore, by construction, every accepted point has to be evaluated on the whole dataset, and the average proportion of data points used is thus lower bounded by the acceptance rate of the algorithm, which in practice is often around 25%. Overall, it is an easy-to-implement feature that does not add any bias, but its benefits are inherently limited, and speed of convergence might be affected.



(a) Chain hist., 10^4 iter. at 10%, $X_i \sim \mathcal{N}(0, 1)$

(b) Chain hist., 10^4 iter. at 10%, $X_i \sim \log \mathcal{N}(0, 1)$



(c) Chain hist., 10^5k iter. at 1%, $X_i \sim \mathcal{N}(0, 1)$

(d) Chain hist., 10^5k iter. at 1%, $X_i \sim \log \mathcal{N}(0, 1)$

Figure 4: Results of SGLD (Welling & Teh, 2011) on our Gaussian and lognormal running examples. See Section 5.1 and the caption of Figure 2 for details.

6 Approximate subsampling approaches

In this Section, we consider again subsampling approaches where, at each MH iteration, a subset of data points is used to approximate the acceptance ratio (5). As mentioned in Section 4.2, it is very simple to obtain an unbiased estimator of the log-likelihood $n\ell(\theta)$ based on random samples $x_1^*, \ldots, x_t^*$ from the dataset $\mathcal{X}$; see (7). Similarly, one can also easily obtain an unbiased estimator of the average log likelihood ratio $[\ell(\theta') - \ell(\theta)]$

$$\Lambda_t^*(\theta, \theta') \triangleq \frac{1}{t} \sum_{i=1}^{t} \log \frac{p(x_i^*|\theta')}{p(x_i^*|\theta)}. \tag{16}$$

Note that unlike the exact approaches of Sections 4 and 5, the methods reviewed in Section 6 do not attempt to sample exactly from π, but just from an approximation to π.

6.1 Naive subsampling

The first approach one could try is to only use a random fixed proportion of data points to estimate π at any newly drawn θ. We highlight that this leads to a nontrivial mixture target that is hard to interpret, where all subsampled posteriors appear, suitably rescaled. Assume that at each new θ drawn in an MH run, we draw n independent Bernoulli variables and let

$$\hat{\ell}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \frac{z_i}{\lambda} \ell_i(\theta) \tag{17}$$

be an unbiased estimator of the average log likelihood $\ell(\theta)$, where $z_i \sim B(1, \lambda)$ i.i.d. Now one could think of plugging estimates $\hat{\gamma}(\theta) = p(\theta) e^{n\hat{\ell}(\theta)}$ in Steps 2 and 3 of MH in Figure 1. However, as $\hat{\gamma}(\theta)$ is not an unbiased estimator of $\gamma(\theta)$, this algorithm samples from a target distribution which is proportional to $p(\theta)\mathbb{E}e^{n\hat{\ell}(\theta)} \neq \gamma(\theta)$, where the expectation is w.r.t the distributions of the Bernoulli random variables $\{z_i\}$. Now

$$\begin{aligned} \mathbb{E}e^{n\hat{\ell}(\theta)} &= \prod_{i=1}^{n} \left[\lambda \ell_i(\theta)^{1/\lambda} + (1 - \lambda) \right] \\ &= \sum_{r=0}^{n} \sum_{\#I_r = r} \lambda^r (1 - \lambda)^{n-r} p(x_{I_r}|\theta)^{1/\lambda}, \end{aligned}$$

where I_r denotes a set of r distinct indices in $\{1, \ldots, n\}$, $x_{I_r} = \{x_i; i \in I_r\}$, and with the convention $p(x_\emptyset|\theta) = 1$. Each subsampled likelihood contributes to the target, exponentiated to the power $1/\lambda$, resulting in a nontrivial mixture of rescaled data likelihood terms. To further simplify, assume $p(x_{I_r}|\theta) \approx p_r(\theta)$ for each set of indices I_r, that is, the variance of the likelihood under subsampling is small, then

$$\mathbb{E}e^{n\hat{\ell}(\theta)} \approx \sum_{r=0}^{n} C_n^r \lambda^r (1 - \lambda)^{n-r} p_r(\theta)^{1/\lambda} = \mathbb{E}_{R \sim B(n,\lambda)} p_R^{1/\lambda}(\theta), \tag{18}$$

where $B(n, \lambda)$ denotes the binomial distribution with parameters n and λ. Noticing that $p_r(\theta)$ is roughly exponentially decreasing in r, the values or r that are larger than the mode of the binomial probability mass function are unlikely to contribute a lot to (18). The largest subsample size contributing to (18) is thus roughly $n\lambda$, and the power $1/\lambda$ makes this term of the same scale as $p(x_1, \ldots, x_n|\theta)$. Broadly speaking, subsampling has a "broadening" effect due to the contribution of the likelihoods of small subsamples.

Alternately, if one starts with the biased estimator of the average log likelihood

$$\tilde{\ell}(\theta) = \frac{1}{n} \sum_{i=0}^{n} z_i \ell_i(\theta), \tag{19}$$

instead of (17) one ends up with

$$\begin{aligned} \mathbb{E} e^{n\tilde{\ell}(\theta)} &= \prod_{i=1}^{n} [\lambda \ell_i(\theta) + (1-\lambda)] \\ &= \sum_{r=0}^{n} \sum_{\#I_r = r} \lambda^r (1-\lambda)^{n-r} p(x_{I_r}|\theta) \\ &\approx \mathbb{E}_{R \sim B(n,\lambda)} p_R(\theta). \end{aligned} \tag{20}$$

Again, all subsampled likelihoods contribute, but without being further exponentiated. Still, the result is a much broadened target, as values of r that are larger than $n\lambda$ are unlikely to contribute a lot. In this case, the broadening effect of subsampling is not only due to the contribution of small subsamples, but also to the absence of bias correction in (19).

We have thus seen that naive subsampling is nontrivial tempering, so that the target is not preserved. Additionally, as mentioned in Section 4.1, the variance of the log likelihood estimator needs to be kept around a constant, 1 or 3 depending on hypotheses, in order for pseudo-marginal MH to be efficient. This means that λ should be such that

$$\frac{(1-\lambda)}{\lambda} \sum_{i=1}^{n} \ell_i(\theta)^2$$

is of order 1 in the case of (17). This entails that λ should be close to 1, so that there is no substantial gain in terms of number of likelihood evaluations. In the case of (19),

$$\lambda(1-\lambda) \sum_{i=1}^{n} \ell_i(\theta)^2$$

can be of order 1 if $\lambda \sim n^{-1}$. But then the leading terms in the mixture target (20) will be the subsampled likelihoods corresponding to small subsamples, so that the target will be very different from the actual target π.

Overall, naive subsampling is a very poor approach. However, it allows us to identify the main issues a good subsampling approach should tackle: guaranteeing its target, not loosing too much convergence speed compared to MH, and cutting the likelihood evaluation budget. As shown in (Bardenet *et al.* , 2014), the first point is an algorithmic design issue, while the last two points are related to controlling the variance of the log likelihood ratios.

6.2 Relying on the CLT

Several authors have appealed to the central limit theorem to justify their assumption that the average subsampled log likelihoods and log likelihood ratios in (7) and (16) are Gaussianly distributed.

If the noise of the log likelihood ratio estimate is normal *with known variance* and mean equal to the true log-likelihood ratio, Ceperley & Dewing (1999) have proposed an MH with a corrected acceptance ratio that is exact, i.e., that still targets π. When the variance of the noise is not known, and is rather estimated, the method becomes inexact. Nicholls *et al.* (2012) propose a heuristic argument to show that this inexact chain gives reasonable approximate results, but the Gaussian assumption remains crucial there. As shown in (Bardenet *et al.* , 2014) and in this paper in Figures 5 and 6, this assumption can be arbitrarily violated when subsampling tall data if the log likelihood ratios $\ell_i(\theta') - \ell_i(\theta)$ are heavy-tailed. Missing log likelihoods in the tails will lead to erroneous decisions, and uncontrolled results.

6.2.1 A pseudo-marginal approach with variance reduction under Gaussian assumption

Quiroz *et al.* (2014) propose a methodology to use MH for tall data which also relies on the assumption that the log-likelihood estimator is Gaussian with mean $\ell(\theta)$, for every θ. By introducing a bias correction providing an approximately unbiased estimate of the likelihood, this corresponds to a pseudo-marginal MH algorithm whose target distribution is proportional to $p(\theta)\mathbb{E}e^{n\hat{\ell}(\theta)-\hat{b}(\theta)}$, where $\hat{b}(\theta)$ is an estimate of the bias $b(\theta)$ satisfying $\mathbb{E}e^{n\hat{\ell}(\theta)} = e^{n\ell(\theta)+b(\theta)}$. They rightly notice that, ideally, if one wants to keep the variance of average subsampled log likelihoods small, one should not subsample data points with or without replacement, but one should rather perform importance sampling with the weight of data point i being proportional to $|\ell_i(\theta)|$. While this variance reduction approach obviously defeats the purpose of subsampling, Quiroz *et al.* (2014) propose to use as weights an approximation of the log-likelihood, based e.g. on a Gaussian process or splines trained on a small subset of computed likelihoods $\ell_i(\theta)$. Finally, a heuristic to adaptively choose the size of the total subsample so as to keep the variance of the log likelihood controlled is proposed. The method is demonstrated to work on a bivariate probit model using only 10% of the full dataset. However, as a general purpose method, it suffers from two limitations. First, it is based on Gaussian assumptions, which can be unreasonable as noted above and it is unclear whether it will be robust to these CLT approximations not being valid. Second, the proposed importance sampling step requires to learn a good proxy for $x \mapsto p(x|\theta)$ for each θ drawn during the MCMC run. The fitted proxies should thus be cheap to train and evaluate, but at the same time accurate if any variance reduction is to be obtained.

6.2.2 Adaptive subsampling with T-tests

Still assuming the noise of the log likelihood is Gaussian, given a drawn $\theta \in \Theta$, one can try to adaptively choose the size of the subsample $\{x_1^*, \dots, x_t^*\}$ to be used in the unbiased estimators (7) or (16), so as to take the correct acceptance decision with high probability. Upon noting that

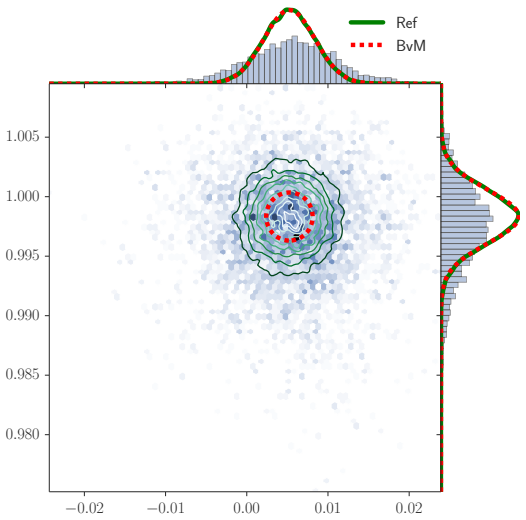
the MH acceptance decision is equivalent to deciding whether $\log \alpha(\theta, \theta') > u$, or equivalently

$$\frac{1}{n}\sum_{i=1}^{n} \log \frac{p(x_i|\theta')}{p(x_i|\theta)} > \frac{1}{n}\log u - \frac{1}{n}\log\left[\frac{p(\theta')q(\theta|\theta')}{p(\theta)q(\theta'|\theta)}\right] \tag{21}$$

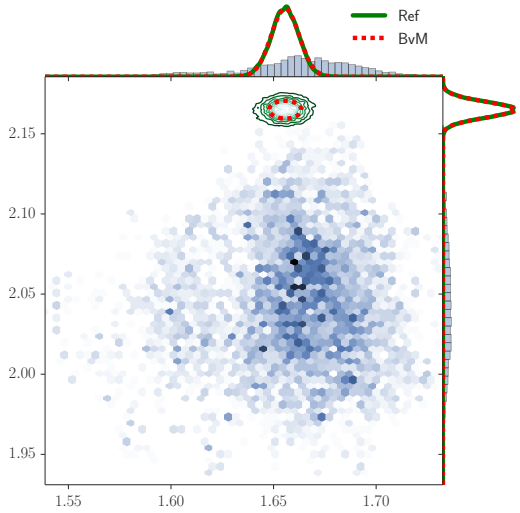
with $u \sim \mathcal{U}_{[0,1]}$ drawn beforehand, statistical tests can be used to assert whether (21) holds with a given level of "confidence". As far as we are aware, Bulgak & Sanders (1988) were the first to consider such a procedure. They used it in a simulated annealing algorithm maximizing a function defined as an expectation w.r.t a probability distribution, and approximated using Monte Carlo. Simulated annealing is a simple non-homogeneous variant of the MH algorithm where the the target distribution is annealed over the iterations. The same application received more attention later (Alkhamis *et al.* , 1999; Wang & Zhang, 2006). Applied to the standard MH, the method has been considered by Singh *et al.* (2012), and more recently by Korattikara *et al.* (2014) specifically for tall data. Korattikara *et al.* (2014) propose an MH-like algorithm called *Austerity MH* that incorporates a sequential T-test to check (21) for each pair (θ, θ'), thus relying on *several* CLTs. They demonstrate dramatic reductions in the average number of subsamples used per MCMC iteration on particular applications. However, as noted in (Korattikara *et al.* , 2014; Bardenet *et al.* , 2014), the results can be arbitrarily far from the original MH when the CLT approximations are not valid.

We show the results of $10\,000$ iterations of Austerity MH on our two running examples in Figure 5. The parameters are $\epsilon = 0.05$, corresponding to the p-value threshold in the aforementioned T-test, and an initial subsample size of 100 at each iteration. In the Gaussian case, the posterior is rightly centered, but is slightly too wide. This is a tempering effect due to too small subsamples, while the CLT-based Student approximation seems reasonable, as shown in Figure 6. In the lognormal case, the departure of the chain from the actual posterior is more remarkable, and relatedly the CLT approximations of Austerity MH are inaccurate for the chosen initial subsample size of 100, as we demonstrate in Figure 6. This explains the strong mismatch of the chain and the posterior in Figure 5(b). The standard deviation of the fitted Gaussian is largely underestimated, due to small subsamples which do not include enough of the tails of the log likelihood ratios, which coincide with the tails of $\mathcal{X}$. Finally, the reductions in the number of samples needed per iteration are quite interesting: half of the iterations require less than 4% of the dataset for the lognormal case, but this is at the price of a large error in the posterior approximation. Augmenting the initial size of the subsample will likely make the CLT approximations tighter, but there is no generic answer as to which size to choose: any fixed choice will fail on an example where the log likelihood ratios have heavy enough tails. In both the Gaussian and the lognormal example, it is actually safer to go with the Bernstein-von Mises approximation, which costs little more than a run of stochastic gradient descent, and only requires one CLT approximation, for a sample of size $n \gg 1$. This illustrates the danger of using CLT-based approximations for small sample sizes, which is related to asymptotic arguments on small batches in Section 3.

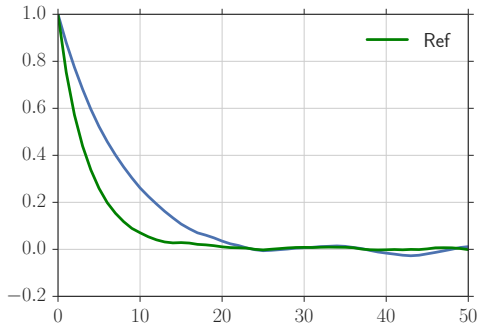
Overall, CLT-based approaches to MH with tall data lead to heuristics with interesting reductions in the number of samples used, but they have little theoretical backing so far and they are not robust to the involved CLT approximations being inaccurate. We note also that the CLT is assumed to provide a good approximation for the log likelihood or log likelihood ratio for



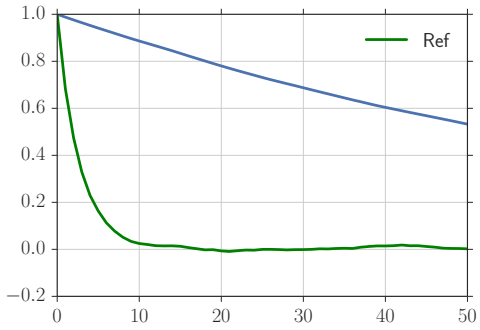
(a) Chain histograms, $X_i \sim \mathcal{N}(0, 1)$



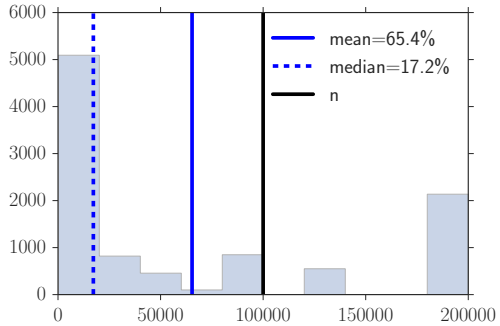
(b) Chain histograms, $X_i \sim \log \mathcal{N}(0, 1)$



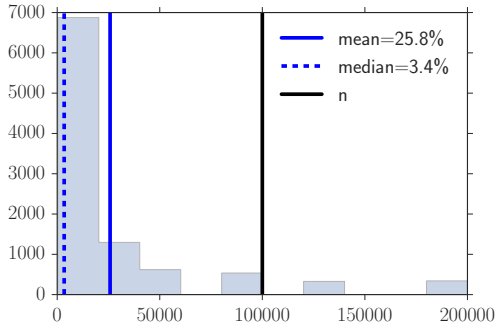
(c) Autocorr. of $\log \sigma$, $X_i \sim \mathcal{N}(0, 1)$



(d) Autocorr. of $\log \sigma$, $X_i \sim \log \mathcal{N}(0, 1)$

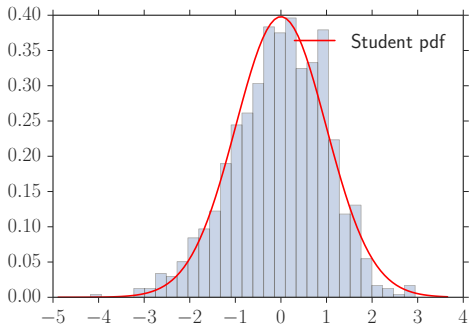


(e) Num. lhd. evals, $X_i \sim \mathcal{N}(0, 1)$

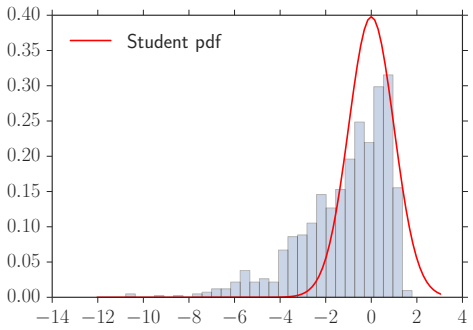


(f) Num. lhd. evals, $X_i \sim \log \mathcal{N}(0, 1)$

Figure 5: Results 10 000 iterations of Austerity MH (Korattikara *et al.* , 2014). See Section 6.2 and the caption of Figure 2 for details.



(a) Student around MAP, $X_i \sim \log \mathcal{N}(0, 1)$



(b) Student around MAP, $X_i \sim \log \mathcal{N}(0, 1)$

Figure 6: Histogram of 1000 realizations of the Student statistic required in Austerity MH, taken at $\theta = \theta_{\text{MAP}}$ and $\theta' \sim q(\cdot|\theta)$. The theoretical Student pdf is plotted in red.

any $\theta, \theta' \in \Theta$, which amounts to more than one Gaussian assumption. The approaches in this section should thus be applied with care. As a minimal sanity-check, we recommend using tests of Gaussianity across $\Theta \times \Theta$ to make sure the CLT assumptions are realistic. Note that even then, there is no guarantee the above algorithms have π for target, if any.

6.3 Exchanging acceptance noise for subsampling noise

This section is an original contribution, which illustrates a way to obtain subsampling algorithms with guarantees under weaker assumptions than Gaussianity. This approach is impractical, but it is of methodological and illustrative interest. First it illustrates a potentially useful technique to *take advantage of* subsampling noise. Second, it is our first illustration of the seemingly inevitable $\mathcal{O}(n)$ average number of subsamples required per MCMC iteration as soon as we do not use any CLT-based approximation and require theoretical guarantees.

Let $\theta, \theta' \in \Theta$, and let $x_1^*, \dots, x_t^*$ be drawn independently with replacement from $\mathcal{X}$. Let $\Lambda_t^*(\theta, \theta')$ be the average subsampled log likelihood ratio defined in (16). Now, we remark that MH has some inherent noise in its acceptance decision (21), encapsulated by the uniform variable $u \sim \mathcal{U}[0, 1]$. Why, then, not rely on the subsampling noise to guarantee exploration, and accept a move if and only if

$$\Lambda_t^*(\theta, \theta') + \frac{1}{n} \log \left[\frac{p(\theta')q(\theta|\theta')}{p(\theta)q(\theta'|\theta)} \right] > 0 \tag{22}$$

instead of (21)? This idea has been first used by Branke *et al.* (2008) to develop heuristics for simulated annealing in the presence of noise. We formalize this argument here in the context of subsampling. For the sake of simplicity, assume for a moment we have a flat prior and a symmetric proposal, so that (22) becomes

$$\Lambda_t^*(\theta, \theta') > 0.$$

We do not assume that the $\Lambda_t^*(\theta,\theta')$'s are Gaussianly distributed, but we make the parametric assumption that the second and third absolute moments σ^2 and ρ of $-\log p(x_i^*|\theta') + \log p(x_i^*|\theta)$ are known and independent of θ, θ'. Applying the Berry-Esseen inequality (van der Vaart & Wellner, 1996) to the variables $-\log p(x_i^*|\theta') + \log p(x_i^*|\theta)$ yields

$$\left|\mathbb{P}(-\Lambda_t^*(\theta,\theta') \leq u) - \Phi\left(\frac{u + \Lambda_n(\theta,\theta')}{\sigma/\sqrt{t}}\right)\right| \leq \frac{K(\sigma,\rho)}{\sqrt{t}} \tag{23}$$

for any $u \in \mathbb{R}$, where Φ is the cdf of a $\mathcal{N}(0,1)$ variable, and

$$\Lambda_n(\theta,\theta') \triangleq \frac{1}{n}\sum_{i=1}^{n} \log \frac{p(x_i|\theta')}{p(x_i|\theta)}$$

is the average log likelihood ratio. When $u = 0$, (23) yields

$$\left|\mathbb{P}(\Lambda_t^*(\theta,\theta') \geq 0) - \Phi\left(\frac{\Lambda_n(\theta,\theta')}{\sigma/\sqrt{t}}\right)\right| \leq \frac{K(\sigma,\rho)}{\sqrt{t}}. \tag{24}$$

Now let $C, \lambda > 0$ be such that for any $x \in \mathbb{R}$,

$$\left|\Phi(x) - \frac{1}{1+e^{-\lambda x}}\right| \leq C.$$

Bowling *et al.* (2009) for instance, empirically found $C = 0.0095$ and $\lambda = 1.702$. Combining this bound with (24), we obtain

$$\left|\mathbb{P}(\Lambda_t^*(\theta,\theta') \geq 0) - \frac{1}{1+e^{-\frac{\lambda\Lambda_n(\theta,\theta')}{\sigma/\sqrt{t}}}}\right| \leq C + \frac{K(\sigma,\rho)}{\sqrt{t}}.$$

Hence, the acceptance probability of an algorithm that would accept the move from θ to θ' by checking whether $\Lambda^*(\theta,\theta') > 0$ is close to the acceptance probability of an MCMC algorithm with a Baker acceptance criterion (Robert & Casella, 2004, Section 7.8.1) that targets π^β with temperature $\beta = \frac{\lambda\sqrt{t}}{n\sigma}$. Arguments such as (Bardenet *et al.* , 2014, Lemma 3.1, Proposition 3.2) could then help concluding that the distance between the kernels of both Markov chains is controlled, which would yield positive ergodicity results, in the line of (Bardenet *et al.* , 2014, Proposition 3.2). This reasoning shows again a close relation between subsampling and tempering, as in Section 6.1, with a clear link between the variance of the subsampled log likelihood ratios and the temperature.

Now, from a practical point of view, in simple applications such as logistic regression, σ is of the order of $\|\theta - \theta'\|$, which in turn should be of order $\mathcal{O}_p(n^{-1/2})$ if the MCMC proposal is a Gaussian random walk with covariance similar to that of π, see Bardenet *et al.* (2014). This means that t has to be of order n for the temperature β to be of order 1, and this approach is thus bound to use $\mathcal{O}(n)$ subsamples per iteration! In conclusion, robustness to non-Gaussianity leads to requiring a fixed proportion of the whole dataset on average, even in the favourable case when one controls the first three moments of the subsampling noise and one swaps subsampling noise for the inherent MCMC acceptance noise.

6.4 Confidence samplers

In (Bardenet *et al.* , 2014), we proposed a controlled approximation of the acceptance decision (21). Indeed, let us fix θ, θ' and momentarily assume that $x \mapsto \log[p(x|\theta')/p(x|\theta)]$ was Lipschitz with known constant. Then, having observed the log likelihood ratio at some points $\{x_i^*, i = 1, \dots, t\} \subset \mathcal{X}$, one could build a lower and an upper bound for the complete log likelihood ratio

$$\frac{1}{n}\sum_{i=1}^{n} \log\left[\frac{p(x_i|\theta')}{p(x_i|\theta)}\right],$$

simply by associating each x_i with the nearest point among $\{x_1^*, \dots, x_t^*\}$. These bounds could be refined by augmenting the set of observed log likelihoods ratios, until eventually one knows for sure whether (21) holds.

Now, concentration inequalities allow softer bounds and require less than this Lipschitz assumption. If one knows a bound for the range

$$C_{\theta,\theta'} \triangleq \max_{i=1}^{n} \left|\log\left[\frac{p(x_i|\theta')}{p(x_i|\theta)}\right]\right|, \tag{25}$$

then concentration inequalities such as Hoeffding's or Bernstein's, yield confidence bounds $c_t(\delta)$ such that

$$\mathbb{P}\left(\left|\frac{1}{t}\sum_{i=1}^{t} \log\left[\frac{p(x_i^*|\theta')}{p(x_i^*|\theta)}\right] - \frac{1}{n}\sum_{i=1}^{n} \log\left[\frac{p(x_i|\theta')}{p(x_i|\theta)}\right]\right| > c_t(\delta)\right) \geq 1 - \delta, \tag{26}$$

where the probability is taken over $x_1^*, \dots, x_t^*$ drawn uniformly from $\mathcal{X}$, with or without replacement. Borrowing from the bandit literature, we explain in (Bardenet *et al.* , 2014) how to leverage such confidence bounds to automatically select a subsample size T such that the right MH acceptance decision is taken with a user-specified probability $1 - \delta$. Note that for our algorithm to bring any improvement over the ideal MH, the range (25) must be cheap to compute, i.e. cheaper than $\mathcal{O}(n)$. This is the case for logistic regression, for example, but it is the major limitation of the approach in Bardenet *et al.* (2014). We showed in (Bardenet *et al.* , 2014, Proposition 3.2) that if the ideal MH sampler is uniformly ergodic then the resulting algorithm inherits the uniform ergodicity of the ideal MH sampler, with a convergence speed that is within $\mathcal{O}(\delta)$ of that of the ideal MH. Importantly, we showed that our sampler then admits a limiting distribution, which is also within $\mathcal{O}(\delta)$ of π. Uniform ergodicity is a very strong assumption and it would be worth extending these results to the geometrically ergodic scenario. There has recently been work in this direction (Alquier *et al.* , 2014; Pillai & Smith, 2014; Rudolf & Schweizer, 2015).

On the negative side, we demonstrated in (Bardenet *et al.* , 2014) that vanilla confidence samplers still require $\mathcal{O}(n)$ samples at each iteration at equilibrium, where the proportionality constant is the variance of the log likelihood ratio under subsampling. This statement relies on the leading term in $c_t(\delta)$ being of order $t^{-1/2}$. In practice, the results of the vanilla confidence sampler on our running examples are shown in Figure 7. We set $\delta = 0.1$ and we place ourselves in the favourable scenario where the algorithm has access to the actual range of each log likelihood ratio. The number of likelihood evaluations is estimated as follows: we take by default

twice the detected value T for the subsample size in general, but only *once* when the previous iteration required computing all n likelihoods at the current state of the chain. Still, even in these favourable conditions, the algorithm basically requires essentially the whole dataset at each iteration.

Concentration inequalities are "worst-case" guarantees, and the theoretical results come at the price of a smaller reduction in the number of samples required. When the target is locally Gaussian, e.g. when Bernstein-von Mises yields a good approximation, there is potentially a lot to be gained, as empirically demonstrated by Korattikara *et al.* (2014), for example. In the current paper, we propose in Section 7 a modified confidence sampler that can leverage concentration of the target to yield dramatic empirical gains while not sacrificing any theoretical guarantee of the confidence sampler. The basic tool is a cheap proxy for the log likelihood ratio that acts as a control variate in the concentration inequality (26). Using a 2nd order Taylor expansion centered at the maximum of the likelihood – obtained with a stochastic gradient descent for example – allows to replace many likelihood evaluations by the evaluation of this Taylor expansion. Figure 8 shows the results of this new confidence sampler with proxy on our running Gaussian and lognormal examples. Our algorithm outperforms all preceding methods, using almost no sample in the Gaussian case where it *automatically* detects that a quadratic form is enough to represent the log likelihood ratio. Finally, we demonstrate in Sections 7.2.3 and 8 that this new algorithm can require less than $\mathcal{O}(n)$ likelihood evaluations per iteration. Combined with the statements in Bardenet *et al.* (2014) that each iteration is almost as efficient as the ideal MH, which is further supported by the match of the autocorrelation functions in Figures 8(c) and 8(d), this opens up big data horizons. We give full details on the confidence algorithm with proxy in Section 7.

7 An improved confidence sampler

In this section, we build upon the confidence sampler in (Bardenet *et al.* , 2014) by introducing likelihood proxies, which act as control variates for the individual likelihoods.

7.1 Introducing proxies in the confidence sampler

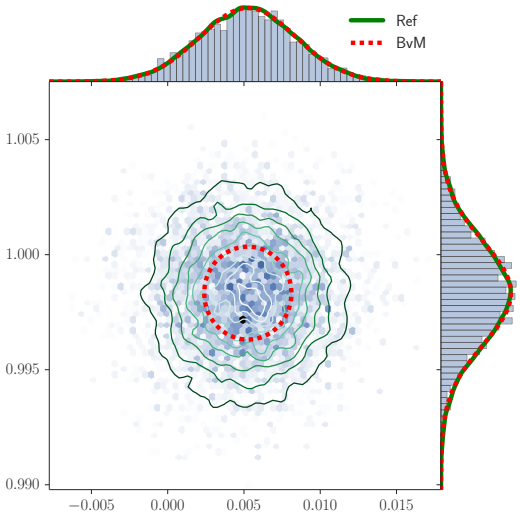
We start by recalling the pseudocode of the confidence sampler in (Bardenet *et al.* , 2014) in Figure 9, using sampling with replacement and a generic empirical concentration bound $c_t(\delta)$. In practice, one can think of the empirical Bernstein bound of Audibert *et al.* (2009)

$$c_t(\delta) = \hat{\sigma}_{t,\theta,\theta'}\sqrt{\frac{2\log(3/\delta)}{t}} + \frac{6C_{\theta,\theta'}\log(3/\delta)}{t}, \tag{27}$$

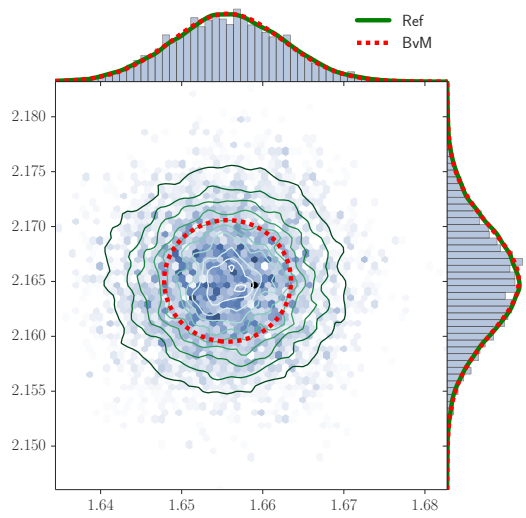
where $\hat{\sigma}_{t,\theta,\theta'}$ is the sample standard deviation of the log likelihood ratio

$$\left\{\log\left[\frac{p(x_i^*|\theta')}{p(x_i^*|\theta)}\right], i = 1,\dots,t\right\},$$

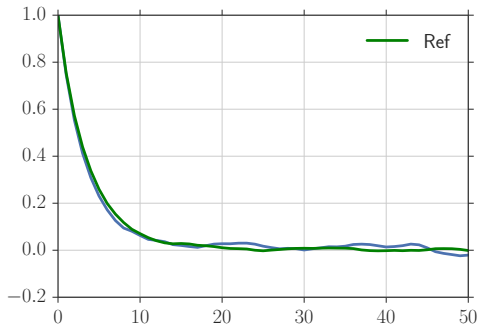
and $C_{\theta,\theta'}$ is their range, defined in (25). We emphasize that other choices of sampling procedure and concentration inequalities are valid, as long as they guarantee a concentration like (26). We



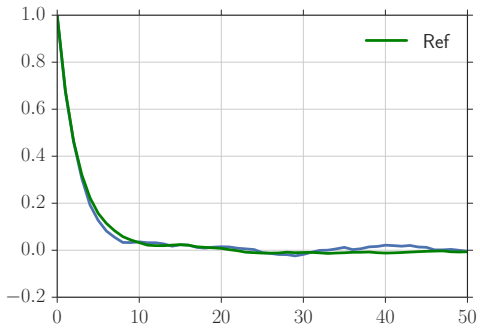
(a) Chain histograms, $X_i \sim \mathcal{N}(0, 1)$



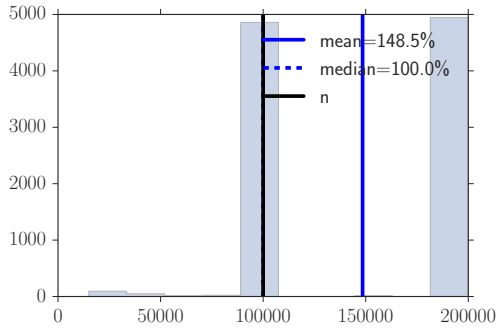
(b) Chain histograms, $X_i \sim \log \mathcal{N}(0, 1)$



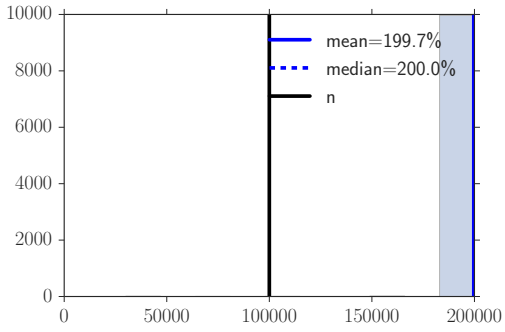
(c) Autocorr. of $\log \sigma$, $X_i \sim \mathcal{N}(0, 1)$



(d) Autocorr. of $\log \sigma$, $X_i \sim \log \mathcal{N}(0, 1)$

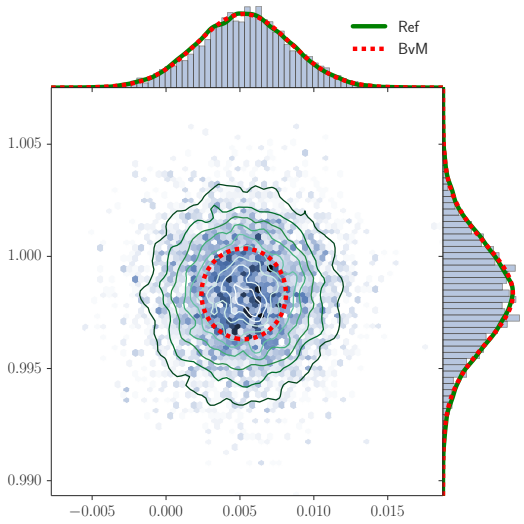


(e) Number of likelihood evals, $X_i \sim \mathcal{N}(0, 1)$

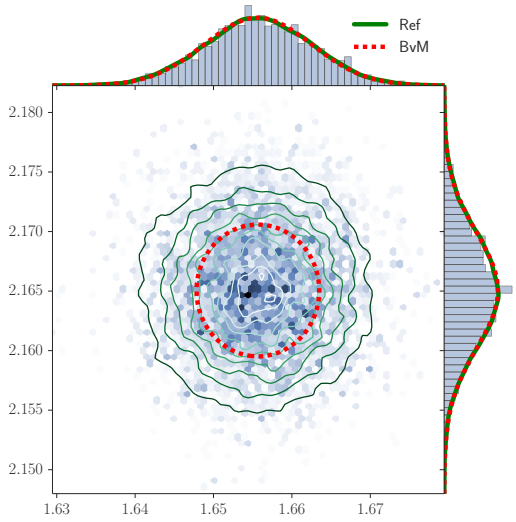


(f) Number of likelihood evals, $X_i \sim \log \mathcal{N}(0, 1)$

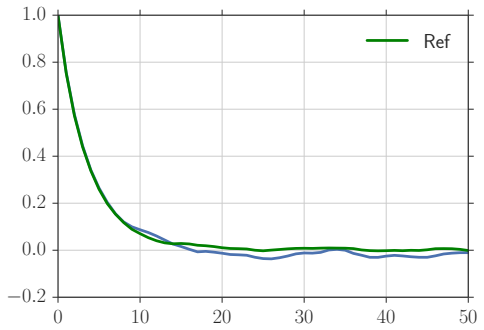
Figure 7: Results of 10 000 iterations of the vanilla confidence sampler (Bardenet *et al.* , 2014), see Section 6.4 and the caption of Figure 2 for details.



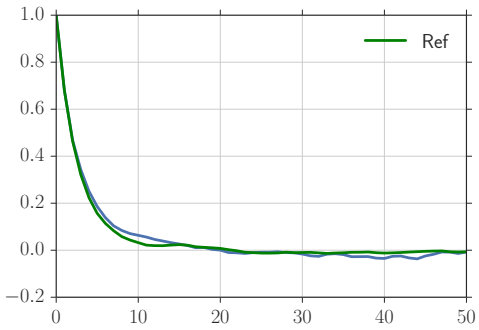
(a) Chain histograms, $X_i \sim \mathcal{N}(0, 1)$



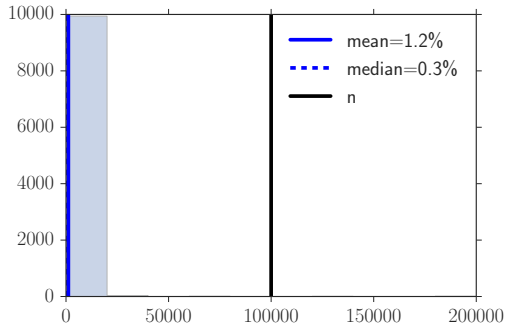
(b) Chain histograms, $X_i \sim \log \mathcal{N}(0, 1)$



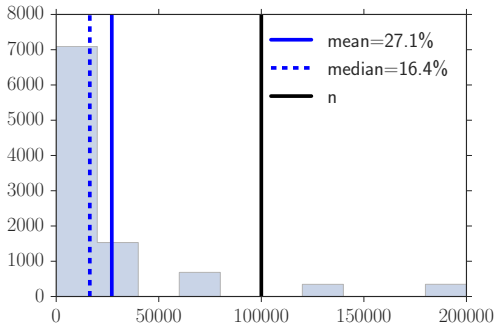
(c) Autocorr. of $\log \sigma$, $X_i \sim \mathcal{N}(0, 1)$



(d) Autocorr. of $\log \sigma$, $X_i \sim \log \mathcal{N}(0, 1)$



(e) Number of likelihood evals, $X_i \sim \mathcal{N}(0, 1)$



(f) Number of likelihood evals, $X_i \sim \log \mathcal{N}(0, 1)$

Figure 8: Results of 10 000 iterations of the confidence sampler of Section 7 with a single 2nd order Taylor proxy at θ_{MAP}.

refer the reader to (Bardenet *et al.* , 2014) for a proof of the correctness of the confidence sampler and implementation details.

CONFIDENCESAMPLER$\big(p(x|\theta),\, p(\theta),\, q(\theta'|\theta),\, \theta_0,\, N_{\text{iter}},\, \mathcal{X},\, (\delta_t),\, C_{\theta,\theta'},\big)$

1 **for** $k \leftarrow 1$ **to** N_{iter}
2 $\quad\theta \leftarrow \theta_{k-1}$
3 $\quad\theta' \sim q(.|\theta)$, $u \sim \mathcal{U}_{(0,1)}$,
4 $\quad\psi(u,\theta,\theta') \leftarrow \frac{1}{n}\log\left[u\frac{p(\theta)q(\theta'|\theta)}{p(\theta')q(\theta|\theta')}\right]$
5 $\quad t \leftarrow 0$
6 $\quad t_{\text{look}} \leftarrow 0$
7 $\quad\Lambda^* \leftarrow 0$
8 $\quad\mathcal{X}^* \leftarrow \emptyset$ ▷ *Keeping track of points already used*
9 $\quad b \leftarrow 1$ ▷ *Initialize batchsize to 1*
10 $\quad$DONE $\leftarrow$ FALSE
11 $\quad$**while** DONE $==$ FALSE **do**
12 $\quad\quad x^*_{t+1},\dots,x^*_b \sim_{\text{w/ repl.}} \mathcal{X}\setminus\mathcal{X}^*$ ▷ *Sample new batch with replacement*
13 $\quad\quad\mathcal{X}^* \leftarrow \mathcal{X}^* \cup \{x^*_{t+1},\dots,x^*_b\}$
14 $\quad\quad\Lambda^* \leftarrow \frac{1}{b}\left(t\Lambda^* + \sum_{i=t+1}^{b}\log\left[\frac{p(x^*_i|\theta')}{p(x^*_i|\theta)}\right]\right)$
15 $\quad\quad t \leftarrow b$
16 $\quad\quad c \leftarrow c_t(\delta_{t_{\text{look}}})$
17 $\quad\quad t_{\text{look}} \leftarrow t_{\text{look}} + 1$
18 $\quad\quad b \leftarrow n \wedge \lceil\gamma t\rceil$ ▷ *Increase batchsize geometrically*
19 $\quad\quad$**if** $|\Lambda^* - \psi(u,\theta,\theta')| \geq c$ **or** $b > n$
20 $\quad\quad\quad$DONE $\leftarrow$ TRUE
21 $\quad$**if** $\Lambda^* > \psi(u,\theta,\theta')$
22 $\quad\quad\theta_k \leftarrow \theta'$ ▷ *Accept*
23 $\quad$**else** $\theta_k \leftarrow \theta$ ▷ *Reject*
24 **return** $(\theta_k)_{k=1,\dots,N_{\text{iter}}}$

Figure 9: Pseudocode of the confidence MH from (Bardenet *et al.* , 2014). Our contribution is a modification of Steps 14, 19 and 21 to introduce proxies for the log likelihood ratios, see Section 7.

The bottleneck for the performance of the confidence sampler was identified in (Bardenet *et al.* , 2014) as the expectation w.r.t. $\pi(\theta)q(\theta'|\theta)$ of the variance of the log likelihood ratio $\log p(x|\theta')/p(x|\theta)$ w.r.t. to the empirical distribution of the observations. We now propose a control variate technique inspired from the Firefly MH of MacLaurin & Adams (2014) to lower this variance down when an accurate and cheap proxy of the log likelihood is known.

We require a proxy for the log likelihood ratio that may decrease the variance of the log

likelihood ratio or its range. More precisely, let $\wp_i(\theta, \theta')$ be such that for any $\theta, \theta' \in \Theta$,

1. $\wp_i(\theta, \theta') \approx \ell_i(\theta') - \ell_i(\theta)$
2. $\sum_{i=1}^n \wp_i(\theta, \theta')$ can be computed cheaply.
3. $|\ell_i(\theta') - \ell_i(\theta) - \wp_i(\theta, \theta')|$ can be bounded uniformly in $1 \leq i \leq n$, and the bound is cheap to compute.

We now simply remark that the acceptance decision (21) in MH is equivalent to checking whether

$$\frac{1}{n}\sum_{i=1}^n \left[\log \frac{p(x_i|\theta')}{p(x_i|\theta)} - \wp_i(\theta, \theta')\right] > \frac{1}{n}\log u - \frac{1}{n}\log\left[\frac{p(\theta')q(\theta|\theta')}{p(\theta)q(\theta'|\theta)}\right] - \frac{1}{n}\sum_{i=1}^n \wp_i(\theta, \theta'). \quad (28)$$

Building the confidence sampler on (28) leads to the same pseudocode as in Figure 9, except that Step 14 is replaced by

$$\Lambda^* \leftarrow \frac{1}{b}\left(t\Lambda^* + \sum_{i=t+1}^{b}\left[\log \frac{p(x_i^*|\theta')}{p(x_i^*|\theta)} - \wp_i(\theta, \theta')\right]\right),$$

the condition in Step 19 is replaced by

$$\left|\Lambda^* + \frac{1}{n}\sum_{i=1}^n \wp_i(\theta, \theta') - \psi(u, \theta, \theta')\right| \geq c,$$

and the condition in Step 21 becomes

$$\Lambda^* > \psi(u, \theta, \theta') - \frac{1}{n}\sum_{i=1}^n \wp_i(\theta, \theta').$$

For completeness, we restate here in Proposition 7.1 that the vanilla confidence sampler inherits the uniform ergodicity of the underlying MH sampler, that its target is within $\mathcal{O}(\delta)$ of π, and that the difference in speed of convergence is also controlled by δ. Let P be the underlying MH kernel, and $\tilde{P}$ the kernel of the confidence sampler described in this section.

Proposition 7.1 *Let P be uniformly geometrically ergodic, i.e., there exists an integer m, a probability measure ν on $(\Theta, \mathcal{B}(\Theta))$ and $0 \leq \rho < 1$ such that for all $\theta \in \Theta$, $P^m(\theta, \cdot) \geq (1-\rho)\nu(\cdot)$. Hence there exists $A < \infty$ such that*

$$\forall \theta \in \Theta, \forall k > 0, \ \|P^k(\theta, \cdot) - \pi\|_{TV} \leq A\rho^{\lfloor k/m \rfloor}. \quad (29)$$

Then there exists $B < \infty$ and a probability distribution $\tilde{\pi}$ on $(\Theta, \mathcal{B}(\Theta))$ such that for all $\theta \in \Theta$ and $k > 0$,

$$\|\tilde{P}^k(\theta, \cdot) - \tilde{\pi}\|_{TV} \leq B[1 - (1-\delta)^m(1-\rho)]^{\lfloor k/m \rfloor}. \quad (30)$$

Furthermore, $\tilde{\pi}$ satisfies

$$\|\pi - \tilde{\pi}\|_{TV} \leq \frac{Am\delta}{1-\rho}. \quad (31)$$

Even in the presence of proxies, the proofs of (Bardenet *et al.* , 2014, Lemma 3.1, Proposition 3.2) apply with straightforward modifications, so that we can extend Proposition 7.1 to the proxy case. The major advantage of this new algorithm is that the sample standard deviation $\hat{\sigma}_{t,\theta,\theta'}$ and range $C_{\theta,\theta'}$ in the concentration inequality (27) are replaced by those of

$$\left\{ \log\left[\frac{p(x_i^*|\theta')}{p(x_i^*|\theta)}\right] - \wp_i(\theta,\theta'), i = 1,\dots,t \right\}.$$

If $\wp_i(\theta,\theta')$ is a good proxy for the log likelihood ratio, one can thus expect significantly more accurate confidence bounds, leading in turn to reduction in the number of samples used.

7.2 An example proxy: Taylor expansions

In general, the choice of proxy $\wp$ will be problem-dependent, and the availability of a good proxy at all is a *strong* assumption, although not as strong as our previous requirement in Bardenet *et al.* (2014) that the range (25) can be computed cheaply, which basically corresponds to $\wp_i(\theta,\theta')$ being identically zero for all i. Indeed, we show in this section that all models that possess up to third derivatives can typically be tackled using Taylor expansions as proxies. In Section 8, we detail the case of logistic regression and gamma linear regression.

7.2.1 Taylor expansions

We expand ℓ_i around some reference value $\theta_\star$ to obtain an estimate

$$\hat{\ell}_i(\theta) = \ell_i(\theta_\star) + g_{i,\star}^T(\theta - \theta_\star) + \frac{1}{2}(\theta - \theta_\star)^T H_{i,\star}(\theta - \theta_\star),$$

where $g_{i,\star}$ and $H_{i,\star}$ are respectively the gradient and the Hessian of ℓ_i at $\theta_\star$. The choice of $\theta_\star$ is deferred to Section 7.2.2. Let us now define $\wp_i(\theta,\theta') = \hat{\ell}_i(\theta') - \hat{\ell}_i(\theta)$. The average $\frac{1}{n}\sum_{i=1}^n \wp_i(\theta,\theta')$ can be computed in $\mathcal{O}(1)$ time if one has precomputed

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^n g_{i,\star}$$

and

$$\hat{S} = \frac{1}{n}\sum_{i=1}^n H_{i,\star}.$$

Indeed, the following holds

$$\frac{1}{n}\sum_{i=1}^n \wp_i(\theta,\theta') = \hat{\mu}^T(\theta' - \theta) + \frac{1}{2}(\theta' - \theta)^T\hat{S}(\theta + \theta' - 2\theta_\star).$$

Finally, assuming

$$\left.\frac{\partial \ell_i}{\partial\theta^{(j)}\partial\theta^{(k)}\partial\theta^{(l)}}\right|_{\theta_\star}$$

can be bounded uniformly in i,j,k,l, the absolute difference $\ell_i(\theta') - \ell_i(\theta) - \wp_i(\theta,\theta')$ can be bounded using the Taylor-Lagrange inequality. To conclude, all conditions of Section 7.1 are satisfied by the proxy $\wp_i$.

7.2.2 Drop proxies along the way

When the mass of the posterior is concentrated around the maximum likelihood estimator θ_{MLE}, a single proxy – say a Taylor proxy centered at $\theta_\star = \theta_{\text{MLE}}$ – can represent the target quite accurately. This is the proxy we used in the running examples of Section 6, see Figure 8. When the posterior does not concentrate, or the proposal is not local enough, such a proxy will be inaccurate, potentially resulting in insufficient subsampling gains. There are various tricks that can be applied. One can either precompute proxies across Θ if one has an idea where the modes of π are, and then use the closest proxy to the current state of the chain at each iteration. Alternately, if one agrees to look at the whole dataset every α iterations, we can *drop proxies along the way*, i.e. set $\theta_\star$ to the current state of the chain every α MH iterations. The whole dataset needs to be browsed at each change of the reference point $\theta_\star$, since there is typically some preprocessing to do in order to compute later bounds. In the case of 2nd order Taylor expansions, for example, one has to compute the full gradient, Hessian, and any other quantity needed to bound the third derivatives. What the user should aim at is to sacrifice a proportion α of the budget of the ideal MH to make the remaining iterations cheaper. The proof of Proposition 7.1 easily generalizes to the case of proxies dropped every constant number of iterations. We demonstrate the empirical performance of such an approach in Sections 8.1.3 and 8.2.2.

7.2.3 A heuristic on the subsampling gain

In (Bardenet *et al.* , 2014), we presented a heuristic that showed the original confidence sampler required $\mathcal{O}(n)$ likelihood evaluations per iteration. At the time, it seemed every attempt at marrying subsampling and MH was fundamentally $\mathcal{O}(n)$. We first repeat here the heuristic from (Bardenet *et al.* , 2014), before arguing that the contributions of this paper can lower this budget to $o(n)$, even $\mathcal{O}(1)$ up to polylogarithmic factors in very favourable conditions.

Assuming a symmetric proposal and a flat prior, the stopping rule of the **while** loop in the original confidence sampler in Figure 9 is met whenever

$$\frac{1}{t}\sum_{i=1}^{t} \log\left[\frac{p(x_i^*|\theta')}{p(x_i^*|\theta)}\right] - \frac{1}{n}\log u$$

is of the same order as the confidence bound $c_t(\delta)$, that is, when $c_t(\delta)$ is of order $1/n$. We consider the Bernstein bound in (27) and we assume the range $C_{t,\theta,\theta'}$ grows with n strictly slower than $\sqrt{n}$. The latter assumption is realistic: $C_{t,\theta,\theta'}$ is often dominated by some power of $\max \|x_i\|_\infty$, and if $x_1, \dots, x_n$ are drawn i.i.d. from a subgaussian distribution, then $\mathbb{E}\max_{i=1}^n x_i = \mathcal{O}(\sqrt{\log n})$ (Cesa-Bianchi & Lugosi, 2006, Lemma A.13). The leading term of the Bernstein bound is proportional to $\hat{\sigma}_{t,\theta,\theta'}/\sqrt{t}$. In simple models such as logistic regression, $\hat{\sigma}_{t,\theta,\theta'}$ is proportional to $\|\theta - \theta'\|$.

Assuming n is large enough that standard asymptotics apply and the target is approximately Gaussian, the results of Roberts & Rosenthal (2001) lead to choose the covariance matrix of the proposal such that $\|\theta - \theta'\|$ is of order $n^{-1/2}$. Summing up, we exit the **while** loop when

$$\frac{1}{n} \sim \frac{1}{\sqrt{t}\sqrt{n}},$$

which leads to $t \sim n$.

Now consider the confidence sampler with second-order Taylor proxies introduced in Section 7.2.1. $\hat{\sigma}_{t,\theta,\theta'}$ and $C_{t,\theta,\theta'}$ now correspond to the standard deviation and range of

$$\left\{ \log \left[\frac{p(x_i^*|\theta')}{p(x_i^*|\theta)} - \wp_i^*(\theta, \theta') \right] ; 1 \leq i \leq t \right\}.$$

Now let us assume the third-order derivatives at the reference point $\theta_\star$ can be bounded, say by some constant times $\max_i \|X_i\|_\infty^3$ as will be the case for the exponential family models of Section 8. Then $\hat{\sigma}_{t,\theta,\theta'}$ and $C_{t,\theta,\theta'}$ are dominated by

$$\max_i \|X_i\|_\infty^3 \left(\|\theta - \theta_\star\|^3 + \|\theta' - \theta_\star\|^3 \right). \tag{32}$$

But $\|\theta - \theta_\star\|$ is of order $n^{-1/2}$ if standard asymptotics (van der Vaart, 2000) yield good approximations and $\theta_\star$ is set to the maximum of the posterior. Alternatively, if one has implemented the strategy of dropping proxies regularly, then $\|\theta - \theta_\star\|$ should be of order $n^{-1/2}$ since we assume the covariance matrix of the proposal distribution is of order $1/n$. Again assuming that $\max_i \|X_i\|_\infty^3$ grows, say, like $\rho(n) = o(n^{1/3})$, we now exit the while loop when

$$\frac{1}{n} \sim \frac{\rho(n)^3}{\sqrt{t} n^{3/2}} = \frac{o(1)}{\sqrt{t}\sqrt{n}}.$$

Thus, when the target is approximately Gaussian and the chain is in the mode, the cost in likelihood evaluations per iteration of the confidence sampler with proxy is likely to be $o(n)$. The actual order of convergence depends on the rate of growth of the bounds on the third derivatives. For example, in the case of independent Gaussian data and still assuming (32), we have $t = \mathcal{O}(1)$ up to polylogarithmic factors.

8 Experiments

As a proof of concept, all experiments in this section avoid loading the dataset or proxy-related quantities into memory by building, maintaining and querying from a disk-based database using *SQLite*[2].

8.1 Logistic regression

8.1.1 A Taylor proxy for logistic regression

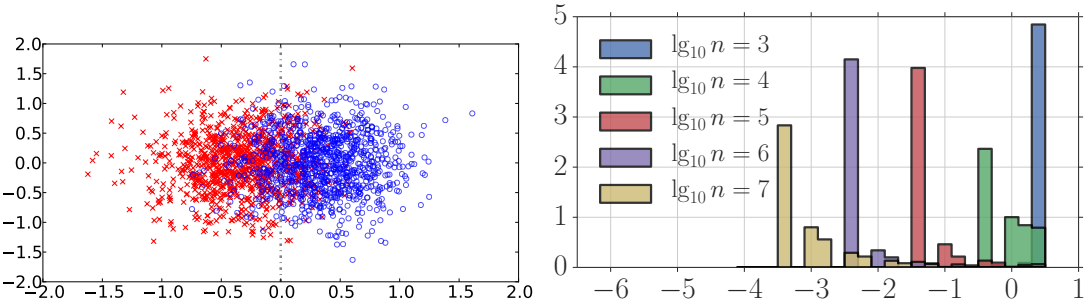
In logistic regression, the likelihood is defined by $\ell_i(\theta) = \phi(t_i x_i^T \theta)$, where

$$\phi(z) = -\log(1 + e^{-z})$$

and the label t_i is in $\{-1, +1\}$. We can use the Taylor expansion proxy of Section 7.2.1, using

$$g_{i,\star} = \phi'(t_i x_i^T \theta_\star) t_i x_i$$

[2] http://www.sqlite.org/



(a) Synthetic logistic regression dataset

(b) log Fraction $\log_{10}(L/n)$ of number of likelihood evals

Figure 10: Results of 10 000 iterations of confidence MH with a single Taylor proxy, applied to a synthetic logistic regression dataset vs. n

and

$$H_{i,\star} = \phi''(t_i x_i^T \theta_\star) x_i x_i^T$$

Furthermore,

$$\frac{\partial}{\partial\theta^{(j)}\partial\theta^{(k)}\partial\theta^{(l)}}\ell_i(\theta) = t_i\phi'''(t_i x_i^T\theta)x_i^{(j)}x_i^{(k)}x_i^{(l)}$$

and

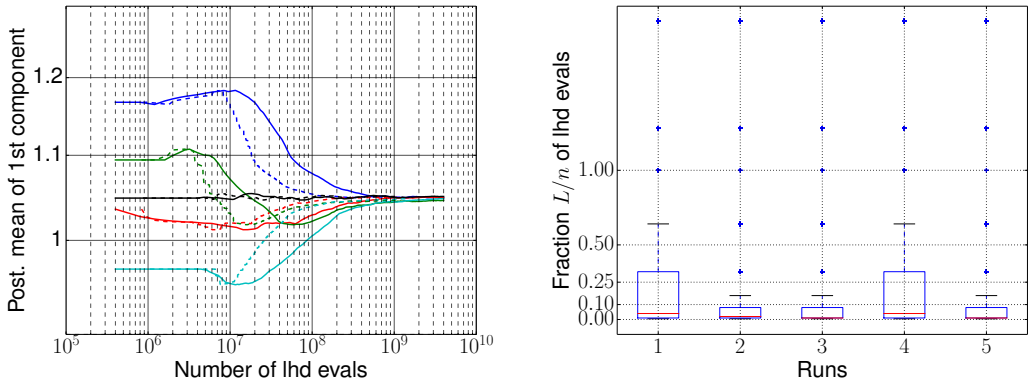
$$\left|\phi'''(z)\right| = \frac{1}{4}\left|\frac{\tanh(z/2)}{\cosh^2(z/2)}\right| \leq \frac{1}{4},$$

so that

$$\begin{aligned}&|\ell_i(\theta') - \ell_i(\theta) - \wp_i(\theta,\theta')| \\ &\qquad \leq \frac{1}{24}\max_{i=1}^{n}\|x_i\|^3\left\{\|\theta-\theta_\star\|^3 + \|\theta'-\theta_\star\|^3\right\}.\end{aligned}$$

8.1.2 A toy example that requires $\mathcal{O}(1)$ likelihood evaluations

In this section, we consider the simple two-dimensional logistic regression dataset in (Bardenet *et al.* , 2014, Section 4.2.2), where the features within each class are drawn from a Gaussian. The dataset is depicted in Figure 10(a). We consider subsets of the dataset with increasing size $\log_{10} n \in \{3,4,5,6,7\}$, run a confidence MH chain for each n, started at the MAP, with $\delta = 0.1$ and a single proxy around the MAP. We report the numbers of likelihood evaluations L at each iteration in Figure 10(b). The fraction of likelihood evaluations compared to MH roughly decreases by a factor 10 when the size of the dataset is multiplied by 10: the number of likelihood evaluations is constant for n large enough. In other words, 1 000 random data points at each iteration are enough to get within $\mathcal{O}(\delta)$ of the actual posterior, the rest of the dataset appears to be superfluous. There is a *saturation* phenomenon. By relaxing the goal of sampling from π into sampling from a controlled approximation, we can break the $\mathcal{O}(n)$ barrier and in this particular example reach a cost per iteration of $\mathcal{O}(1)$.



(a) Posterior mean vs. iteration number

(b) Fraction of likelihood evaluations

Figure 11: Results of 5 runs of a confidence sampler with Taylor proxies dropped every 10 iterations, applied to logistic regression on *covtype*. In Figure 11(a), a solid line corresponds to the online posterior mean of the 1st component of the chain vs. the budget of MH, while a dashed line of the same color corresponds to the budget of the confidence sampler.

8.1.3 The *covtype* dataset

We consider the dataset *covtype.binary*[3] described in Collobert *et al.* (2002). The dataset consists of 581,012 points, of which we pick $n = 400,000$ as a training set, following the maximum training size in Collobert *et al.* (2002). The original dimension of the problem is 54, with the first 10 attributes being quantitative. To illustrate our point without requiring a more complex sampler than MH, we only consider the 10 quantitative attributes. We use the preprocessing and Cauchy prior recommended by Gelman *et al.* (2008).

We run 5 independent chains for 10 000 iterations, dropping proxies every 10 iterations as explained in Section 7.2.2. We obtain a Gelman-Rubin statistic of 1.01 (Robert & Casella, 2004, Section 12.3.4), which suggests the between-chain variance is low enough that we can stop sampling.

We estimate the number of likelihood evaluations L_k at MH iteration k as follows. First, note that –dropping proxies or not– on a regular iteration where the proxy is not necessarily recomputed, L_k can take values up to $2n$, unlike MH, which can store the evaluation of the likelihood at the current state of the chain from one iteration to the next, and thus only requires n likelihood evaluations per iteration. Second, at an iteration where the proxy is recomputed, the whole data has to be read anyway, so that we choose here to perform a normal MH iteration. This requires the maximum $2n$ likelihood evaluations, Assuming the cost of the likelihood evaluation is the bottleneck, we neglect here the additional cost of computing the proxy itself, and only report $L_k = 2n$ when the proxy is recomputed. Third, whenever we compute the full likelihood at a state of the chain, we store it until the chain leaves that state, similarly to any implementation of MH. Thus, at an iteration that follows a full read of the data, i.e. $L_{k-1} = 2n$,

[3] available at http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/binary.html

we only count the likelihood evaluations of the proposed state.

We summarize the results in Figure 11. All runs use on average 27 to 42% of n likelihood evaluations per iteration. Since we compute the proxy every $\alpha = 10$ iterations, there is a necessary $2 \times 10 = 20\%$ of n that is due to recomputing the proxy. We manually assessed the value of α, and recomputing the proxy less often increases the average number of likelihood evaluations (not shown). Thanks to these forced 20%, the rest of the iterations are considerably cheaper than n, since 50% of the iterations require less than 5% of the dataset, as shown in Figure 11(b). Relatedly, although subsampling implies a forced $2n$ likelihood evaluations to start and thus shows an initial delay in Figure 11(a), it quickly catches up and converges faster. The gains are two- or threefold, which is of limited overall practical interest, but we know from Section 7.2.3 and Figure 10(b) that increasing n will also improve the gain.

8.2 Gamma linear regression

8.2.1 A Taylor proxy for gamma regression

In gamma regression, the nonnegative response y is assumed to be gamma-distributed

$$y \sim \Gamma\left(\kappa, \frac{e^{x^T\theta}}{\kappa}\right)$$

where $\Gamma(\kappa, s)$ is the gamma distribution with shape parameter κ and scale parameters s. Assuming κ is known, the log likelihood is thus given by

$$\log p(y|x,\theta) \propto -\kappa y e^{-\theta^T x} - \kappa\theta^T x$$

up to an additive constant, so that

$$\nabla \log p(y|x,\theta) = \kappa\left(ye^{-x^T\theta} - 1\right)x,$$

$$\text{Hess}(\log p(y|x,\theta)) = -\kappa y e^{-x^T\theta} x x^T$$

and

$$\frac{\partial}{\partial\theta^{(j)}\partial\theta^{(k)}\partial\theta^{(l)}} \log p(y|x,\theta) = \kappa y e^{-x^T\theta} x^{(j)}x^{(k)}x^{(l)}.$$

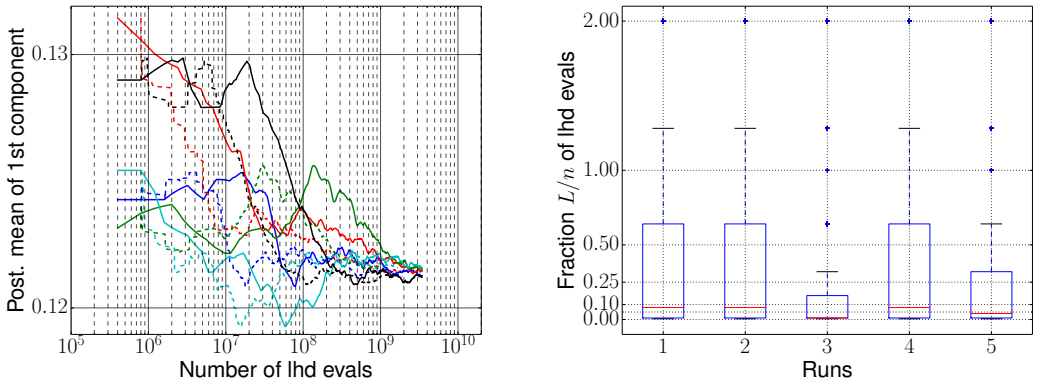
Furthermore, we can bound

$$\left|\frac{\partial}{\partial\theta^{(j)}\partial\theta^{(k)}\partial\theta^{(l)}} \log p(y|x,\theta)\right| \leq \kappa \max_{i=1}^{n} |y| \exp\left(-\min_{i=1}^{n} x_i^T\theta\right) \max_{1\leq i\leq n} \|x_i\|_\infty^3.$$

The Taylor proxies of Section 7.2.1 can thus be applied.

8.2.2 The *covtype* dataset

As an application, we consider the *covtype* dataset again and regress the nonnegative feature "horizontal distance to nearest wildfire ignition" onto the other quantitative features. We run 5 independent chains for 10 000 iterations, dropping proxies every 10 iterations as explained in



(a) Posterior mean vs. iteration number

(b) Fraction of likelihood evaluations

Figure 12: Results of 5 runs of a confidence sampler with Taylor proxies dropped every 10 iterations, applied to gamma linear regression on *covtype*. In Figure 12(a), a solid line corresponds to the online posterior mean of the 1st component of the chain vs. the budget of MH, while a dashed line of the same color corresponds to the budget of the confidence sampler.

Section 7.2.2. We obtain a Gelman-Rubin statistic of 1.001, which again suggests we can stop sampling. We estimate the evaluation budget as in Section 8.1.3. We summarize the results in Figure 12.

All runs use on average 33 to 54% of n likelihood evaluations per iteration, from which $2 \times 10 = 20\%$ are due to recomputing the proxy every 10 iterations. Recomputing the proxy less often increases the average number of likelihood evaluations (not shown). Thanks to these forced 20% the rest of the iterations are considerably cheaper than n, since, as in Section 8.1.3, 50% of the iterations require less than 10% of the dataset. Relatedly, and similarly to the logistic regression task in Section 8.1.3 subsampling converges two or three times faster in this example. Again, this is a proof of concept that subsampling works, and we know from Section 7.2.3 and Figure 10(b) that increasing n will also improve the gain.

9 Discussion

We have reviewed recent advances in applying MCMC to tall datasets. Divide-and-conquer approaches have yet to solve the recombination problem, i.e. how to obtain a *meaningful* distribution in a *stable* manner from the output of individual chains on a growing number of smaller datasets. Subsampling approaches face different issues, namely that of approaching the right target at a known speed, and of keeping the overall budget in terms of likelihood evaluations per iteration low.

In this paper, we have proposed an original subsampling approach. We have showed that under strong ergodicity assumptions on the original MH sampler, our algorithm samples from a controlled approximation of the posterior target. While these strong assumptions are rarely satisfied in practice, our experiments suggest that our results extend to more general scenarios.

In terms of scaling, the introduced methodology is even able to lower the natural cost of $\mathcal{O}(n)$ subsamples per iteration to as low as $\mathcal{O}(1)$ in favourable scenarios. However, we have yet only observed these dramatic gains in contexts where the Bernstein-von Mises approximation is already excellent. On the positive side, our algorithm improves on other proposed subsampling approaches in this context. On the negative side, computing the Bernstein-von Mises approximation for regular models can be typically achieved in only a couple of passes over the data, using for example stochastic gradient to compute the maximum likelihood estimator, and the observed information matrix at this point to estimate the Hessian.

Further work should thus now focus on demonstrating the applicability of subsampling approaches to cases where it is either difficult to compute Bernstein-von Mises even if it is a good approximation (Chernozhukov & Hong, 2003), or – more importantly – cases where n is not big enough that Bernstein-von Mises yields a good approximation.

Acknowledgments

The authors acknowledge Louis Aslett, Nando de Freitas, Pierre Jacob, François Septier, Matti Vihola, and Sebastian Vollmer for their comments and discussions on this paper and topic.

Appendix A: proof of Proposition 4.1

Define

$$S_n = e^{na(\theta)} \left[1 + \sum_{k=1}^{n} \frac{1}{k!} \prod_{j=1}^{k} D_j^* \right]. \tag{33}$$

By construction and the monotone convergence theorem, $\mathbb{E}S_n \to \mathbb{E}S = e^{n\ell(\theta)}$, where

$$S = e^{na(\theta)} \left[1 + \sum_{k=1}^{\infty} \frac{1}{k!} \prod_{j=1}^{k} D_j^* \right].$$

From (Rhee & Glynn, 2013, Theorem 1), the second moment of Y is

$$\mathbb{E}Y^2 = \sum_{k=0}^{\infty} \frac{1}{\mathbb{P}(N \geq k)} \left[\mathbb{E}(S - S_{k-1})^2 - \mathbb{E}(S - S_k)^2 \right],$$

with the convention $S_{-1} = 0$ and $S_0 = e^{na(\theta)}$. We note that

$$(S - S_k)^2 = e^{2na(\theta)} \sum_{p=k+1}^{\infty} \sum_{q=k+1}^{\infty} \frac{1}{p!q!} \prod_{u=1}^{p} D_u^* \prod_{v=1}^{q} D_v^*.$$

Hence

$$
\begin{aligned}
e^{-2na(\theta)}&\left[\mathbb{E}(S-S_{k-1})^2-\mathbb{E}(S-S_k)^2\right]\\
&=\frac{1}{k!k!}\mathbb{E}\prod_{u=1}^{k}D_u^*\prod_{v=1}^{k}D_v^*+2\sum_{j=k+1}^{\infty}\frac{1}{k!j!}\mathbb{E}\prod_{u=1}^{j}D_u^*\prod_{v=1}^{k}D_v^*\\
&=\frac{1}{k!k!}\left[n^2\sigma_t(\theta)^2+n^2(\ell(\theta)-a(\theta))^2\right]^k\\
&\quad+2\sum_{j=k+1}^{\infty}\frac{1}{k!j!}\left[n(\ell(\theta)-a(\theta))\right]^{j-k}\left[n^2\sigma_t(\theta)^2+n^2(\ell(\theta)-a(\theta))^2\right]^k\\
&\geq\frac{\left[n^2\sigma_t(\theta)^2+n^2(\ell(\theta)-a(\theta))^2\right]^k}{k!k!}.
\end{aligned}
$$

Now, since $k!k!\leq 4^{-k}(2k+1)!$, letting

$$A_n\triangleq(1+\epsilon)[n^2\sigma_t(\theta)^2+n^2(\ell(\theta)-a(\theta))^2],$$

and by definition of N, it comes

$$
\begin{aligned}
\frac{\mathrm{Var}Y}{e^{2n\ell(\theta)}} &\geq e^{-2n(\ell(\theta)-a(\theta))}\sum_{k=0}^{\infty}\frac{[2\sqrt{A_n}]^{2k}}{(2k+1)!}-1\\
&= \frac{e^{-2n(\ell(\theta)-a(\theta))}}{2\sqrt{A_n}}\sinh(2\sqrt{A_n})-1\\
&= \frac{e^{-2n(\ell(\theta)-a(\theta))}}{4\sqrt{A_n}}\left[e^{2\sqrt{A_n}}-e^{-2\sqrt{A_n}}\right]-1\\
&= \frac{e^{-2n(\ell(\theta)-a(\theta))+2n\sqrt{(1+\epsilon)[\sigma_t(\theta)^2+(\ell(\theta)-a(\theta))^2]}}}{n\sqrt{(1+\epsilon)[\sigma_t(\theta)^2+(\ell(\theta)-a(\theta))^2]}}+\mathcal{O}(1).
\end{aligned}
$$

Appendix B: proof of Proposition 4.2

We write

$$\begin{aligned}
\mathrm{Var}_z\left[\sum_{i=1}^n \log p(x_i|\theta,z_i)\right] &= \sum_{i=1}^n \left[\mathbb{E}\log^2 p(x_i|\theta,z_i) - (\mathbb{E}\log p(x_i|\theta,z_i))^2\right] \\
&= \sum_{i=1}^n \left[(1-I_\theta)\log^2\left(\frac{e^{\ell_i(\theta)}-e^{b_i(\theta)}}{1-I_\theta}\right) + I_\theta \log^2\left(\frac{e^{b_i(\theta)}}{I_\theta}\right)\right] \\
&\quad - \left[(1-I_\theta)\log\left(\frac{e^{\ell_i(\theta)}-e^{b_i(\theta)}}{1-I_\theta}\right) + I_\theta \log\left(\frac{e^{b_i(\theta)}}{I_\theta}\right)\right]^2 \\
&= I_\theta(1-I_\theta)\sum_{i=1}^n \left[\log\left(\frac{e^{\ell_i(\theta)}-e^{b_i(\theta)}}{1-I_\theta}\right) - \log\left(\frac{e^{b_i(\theta)}}{I_\theta}\right)\right]^2 \\
&= I_\theta(1-I_\theta)\sum_{i=1}^n \log^2\left[\frac{I_\theta}{1-I_\theta}\left(e^{\ell_i(\theta)-b_i(\theta)}-1\right)\right].
\end{aligned}$$

References

Alkhamis, T. M., Ahmed, M. A., & Tuan, V. K. 1999. Simulated annealing for discrete optimization with estimation. *European Journal of Operational Research*, **116**, 530–544.

Alquier, P., Friel, N., Everitt, R., & Boland, A. 2014. Noisy Monte Carlo: convergence of Markov chains with approximate transition kernels. *Statistics and Computing*.

Andrieu, C., & Roberts, G. O. 2009. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, **37**(2), 697–725.

Andrieu, C., & Vihola, M. 2015. Convergence properties of pseudo-marginal Markov chain Monte Carlo algorithms. *to appear in Annals of Applied Probability, available as http://arxiv.org/abs/1210.1484*.

Andrieu, C., Doucet, A., & Holenstein, R. 2010. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society B*.

Audibert, J.-Y., Munos, R., & Szepesvári, Cs. 2009. Exploration-exploitation trade-off using variance estimates in multi-armed bandits. *Theoretical Computer Science*.

Banterle, M., Grazan, C., Lee, A., & Robert, C. P. 2015. Accelerating Metropolis-Hastings algorithms by Delayed acceptance. *Preprint, available as http://arxiv.org/abs/1503.00996*.

Bardenet, R., Doucet, A., & Holmes, C. 2014. Towards scaling up MCMC: an adaptive subsampling approach. *In: Proceedings of the International Conference on Machine Learning (ICML)*. http://jmlr.org/proceedings/papers/v32/bardenet14-supp.pdf.

Beaumont, M. A. 2003. Estimation of population growth or decline in genetically monitored populations. *Genetics*, **164**, 11391160.

Betancourt, M. J. 2014. The Fundamental Incompatibility of Hamiltonian Monte Carlo and Data Subsampling. *Preprint, available as http://arxiv.org/abs/1502.01510*.

Bhanot, G., & Kennedy, A. D. 1985. Bosonic lattice gauge theory with noise. *Physics Letters B*, **157**(1), 70 – 76.

Bowling, S. R., Khasawneh, M. T., Kaewkuekool, S., & Cho, B. R. 2009. A logistic approximation to the cumulative normal distribution. *Journal of industrial engineering and management*, **2**(1), 114–127.

Branke, J., Meisel, S., & Schmidt, C. 2008. Simulated annealing in the presence of noise. *Journal of Heuristics*, **14**, 627–654.

Bulgak, A. A., & Sanders, J. L. 1988. Integrating a modified simulated annealing algorithm with the simulation of a manufacturing system to optimize buffer sizes in automatic assembly systems. *In: Proceedings of the 20th Winter Simulation Conference*.

Ceperley, D. M., & Dewing, M. 1999. The Penalty Method for Random Walks with Uncertain Energies. *Journal of Chemical Physics*, **110**.

Cesa-Bianchi, N., & Lugosi, G. 2006. *Prediction, Learning, and Games*. New York, NY, USA: Cambridge University Press.

Chen, T., Fox, E. B., & Guestrin, C. 2014. Stochastic Gradient Hamiltonian Monte Carlo. *In: Proceedings of the International Conference on Machine Learning (ICML)*.

Chernozhukov, V., & Hong, H. 2003. An MCMC Approach to Classical Estimation. *Journal of Econometrics*.

Collobert, R., Bengio, S., & Bengio, Y. 2002. A Parallel Mixture of SVMs for Very Large Scale Problems. *Neural Computation*, **14**(5), 1105–1114.

Cuturi, M., & Doucet, A. 2014. Fast Computation of Wasserstein Barycenters. *In: Proceedings of The International Conference on Machine Learning (ICML)*.

Douc, R., Moulines, É., & Stoffer, D. 2014. *Nonlinear time series*. Chapman-Hall.

Doucet, A., Pitt, M., Deligiannidis, G., & Kohn, R. 2015. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika, to appear, available as http://arxiv.org/abs/1210.1871*.

Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. 1987. Hybrid Monte Carlo. *Physics Letters B*, 2774–2777.

Gelman, A., Jakulin, A, Pittau, M.G., & Su, Y-S. 2008. A weakly informative default prior distribution for logistic and other regression models. *Annals of applied Statistics)*.

Gelman, A., Vehtari, A., Jylänki, P., Robert, C., Chopin, N., & Cunningham, J. P. 2014. Expectation propagation as a way of life. *Preprint, available as http://arxiv.org/abs/1412.4869*.

Glynn, P. W., & Rhee, C.-H. 2014. Exact Estimation for Markov Chain Equilibrium Expectations. *Journal of Applied Probability*, **51A**, 377–389.

Huang, Z., & Gelman, A. 2005. *Sampling for Bayesian computation with large datasets*. Tech. rept. Department of Statistics, Columbia University.

Jacob, P. E., & Thiery, A. H. 2013. On non-negative unbiased estimators. *Preprint, available as http://arxiv.org/abs/1309.6473*.

Korattikara, A., Chen, Y., & Welling, M. 2014. Austerity in MCMC Land: Cutting the Metropolis-Hastings Budget. *In: Proceedings of the International Conference on Machine Learning (ICML)*.

Lin, L., Liu, K. F., & Sloan, J. 2000. A noisy Monte Carlo algorithm. *Physical Review D*, **61**(074505).

MacLaurin, D., & Adams, R. P. 2014. Firefly Monte Carlo: Exact MCMC with Subsets of Data. *In: Proceedings of the conference on Uncertainty in Artificial Intelligence (UAI)*.

Mak, C. H. 2005. Stochastic Potential Switching Algorithm for Monte Carlo Simulations of Complex Systems. *Journal of Chemical Physics*, **122**(21).

Marin, J.-M., Pudlo, P., Robert, C. P., & Ryder, R. 2012. Approximate Bayesian Computation methods. *Statistics and Computing*, **22**(6), 1167–1180.

Minsker, S., Srivastava, S., Lin, L., & Dunson, D. 2014. Scalable and Robust Bayesian Inference via the Median Posterior. *In: Proceedings of The International Conference on Machine Learning (ICML)*.

Neiswanger, W., Wang, C., & Xing, E. 2014. Asymptotically exact, embarassingly parallel MCMC. *In: Proceedings of the conference on Uncertainty in Artificial INtelligence (UAI)*.

Nicholls, G. K., Fox, C., & Muir-Watt, A. 2012. Coupled MCMC with a randomized acceptance probability. *Preprint, available as http://arxiv.org/abs/1307.5302*.

Pillai, N. S., & Smith, A. 2014. Ergodicity of Approximate MCMC Chains with Applications to Large Data Sets. *Preprint, available as http://arxiv.org/abs/1405.0182*.

Quiroz, M, Villani, M., & Kohn, R. 2014. Speeding Up MCMC by Efficient Data Subsampling. *Preprint, available as http://arxiv.org/abs/1404.4178*.

Rhee, C.-H., & Glynn, P. W. 2013. *Unbiased Estimation with Square Root Convergence for SDE Models*. Tech. rept. Stanford University.

Robert, C. P., & Casella, G. 2004. *Monte Carlo Statistical Methods*. New York: Springer-Verlag.

Roberts, G. O., & Rosenthal, J. S. 2001. Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, **16**, 351–367.

Rudolf, D., & Schweizer, N. 2015. Perturbation theory for Markov chains via Wasserstein distance. *Preprint, available as http://arxiv.org/abs/1503.04123*.

Scott, S. L., Blocker, A. W., & V., Bonassi F. 2013. Bayes and Big Data: The Consensus Monte Carlo Algorithm. *In: Proceedings of the Bayes 250 conference*.

Singh, S., Wick, M., & McCallum, A. 2012. Monte Carlo MCMC: Efficient Inference by Approximate Sampling. *In: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Srivastava, S., Cevher, V., Tran-Dinh, Q., & Dunson, D. B. 2014. WASP: scalable Bayes via barycenters of subset posteriors. *Preprint*.

Strathmann, H., Sejdinovic, D., & Girolami, M. 2015. Unbiased Bayes for Big Data: Paths of Partial Posteriors. *Preprint, available as http://arxiv.org/abs/1501.03326*.

Teh, Y. W., Thiery, A. H., & Vollmer, S. J. 2014. Consistency and fluctuations for stochastic gradient Langevin dynamics. *Preprint, available as http://arxiv.org/abs/1409.0578*.

Trotta, R. 2006. Bayes in the sky: Bayesian inference and model selection in cosmology. *Contemporary Physics*.

van der Vaart, A. W. 2000. *Asymptotic Statistics*. Cambridge University Press.

van der Vaart, A. W., & Wellner, J. A. 1996. *Weak convergence and empirical processes*. Springer.

Wang, L., & Zhang, L. 2006. Stochastic optimization using simulated annealing with hypothesis test. *Applied Mathematics and Computation*, **174**, 1329–1342.

Wang, X., & Dunson, D. B. 2013. Parallelizing MCMC via Weierstrass Sampler. *Preprint, available as http://arxiv.org/abs/1312.4605*.

Welling, M., & Teh, Y. W. 2011. Bayesian Learning via Stochastic Gradient Langevin Dynamics. *In: Proceedings of the International Conference on Machine Learning (ICML)*.

Wright, Jessica. 2014. Genetics: unravelling complexity. *Nature*, **508**(7494), S6–S7.

Xu, M., Lakshminarayanan, B., Teh, Y. W., Zhu, J., & Zhang, B. 2014. Distributed Bayesian posterior sampling via moment sharing. *In: Advances in Neural Information Processing Systems (NIPS)*.