HAL
open science

Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach

Rémi Bardenet, Arnaud Doucet, Chris Holmes

To cite this version:

Rémi Bardenet, Arnaud Doucet, Chris Holmes. Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach. International Conference on Machine Learning (ICML), 2014, Beijing, China. pp.405-413. hal-01254232

HAL Id: hal-01254232

https://hal.science/hal-01254232v1

Submitted on 11 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach

Rémi Bardenet REMI.BARDENET@GMAIL.COM
Arnaud Doucet DOUCET@STATS.OX.AC.UK
Chris Holmes CHOLMES@STATS.OX.AC.UK
Department of Statistics, University of Oxford, Oxford OX1 3TG, UK

Abstract

Markov chain Monte Carlo (MCMC) methods are often deemed far too computationally intensive to be of any practical use for large datasets. This paper describes a methodology that aims to scale up the Metropolis-Hastings (MH) algorithm in this context. We propose an approximate implementation of the accept/reject step of MH that only requires evaluating the likelihood of a random subset of the data, yet is guaranteed to coincide with the accept/reject step based on the full dataset with a probability superior to a user-specified tolerance level. This adaptive subsampling technique is an alternative to the recent approach developed in (Korattikara et al., 2014), and it allows us to establish rigorously that the resulting approximate MH algorithm samples from a perturbed version of the target distribution of interest, whose total variation distance to this very target is controlled explicitly. We explore the benefits and limitations of this scheme on several examples.

1. Introduction

Consider a dataset $\mathcal{X} = \{x_1, ..., x_n\}$ and denote by $p(x_1, ..., x_n|\theta)$ the associated likelihood for parameter $\theta \in \Theta$. Henceforth we assume that the data are conditionally independent, so that $p(x_1, ..., x_n|\theta) = \prod_{i=1}^n p(x_i|\theta)$. Given a prior $p(\theta)$, Bayesian inference relies on the posterior $\pi(\theta) \propto p(x_1, ..., x_n|\theta)p(\theta)$. In most applications, this posterior is intractable and we need to rely on Bayesian computational tools to approximate it.

A standard approach to sample approximately from $\pi(\theta)$ is the Metropolis-Hastings algorithm (MH; Robert & Casella (2004, Chapter 7.3)). MH consists in building an ergodic Markov chain of invariant distribution $\pi(\theta)$. Given a proposal $q(\theta'|\theta)$, the MH algorithm starts its chain at a user-defined θ_0, then at iteration $k+1$ it proposes a candidate state $\theta' \sim q(\cdot|\theta_k)$ and sets θ_{k+1} to θ' with probability

$$\begin{aligned}\alpha(\theta_k, \theta') &= 1 \wedge \frac{\pi(\theta')}{\pi(\theta_k)} \frac{q(\theta_k|\theta')}{q(\theta'|\theta_k)} \\ &= 1 \wedge \frac{p(\theta')}{p(\theta_k)} \frac{q(\theta_k|\theta')}{q(\theta'|\theta_k)} \prod_{i=1}^n \frac{p(x_i|\theta')}{p(x_i|\theta_k)}, \quad (1)\end{aligned}$$

while θ_{k+1} is otherwise set to θ_k. When the dataset is large ($n \gg 1$), evaluating the likelihood ratio appearing in the MH acceptance ratio (1) is too costly an operation and rules out the applicability of such a method.

The aim of this paper is to propose an approximate implementation of this "ideal" MH sampler, the maximal approximation error being pre-specified by the user. To achieve this, we first present the "ideal" MH sampler in a slightly non-standard way.

In practice, the accept/reject step of the MH step is implemented by sampling a uniform random variable $u \sim \mathcal{U}_{(0,1)}$ and accepting the candidate if and only if

$$u < \frac{\pi(\theta')}{\pi(\theta_k)} \frac{q(\theta_k|\theta')}{q(\theta'|\theta_k)}. \quad (2)$$

In our specific context, it follows from (2) and the independence assumption that there is acceptance of the candidate if and only if

$$\Lambda_n(\theta_k, \theta') > \psi(u, \theta_k, \theta'), \quad (3)$$

where for $\theta, \theta' \in \Theta$ we define the average log likelihood ratio $\Lambda_n(\theta, \theta')$ by

$$\Lambda_n(\theta, \theta') = \frac{1}{n} \sum_{i=1}^n \log \left[\frac{p(x_i|\theta')}{p(x_i|\theta)} \right] \quad (4)$$

and where

$$\psi(u, \theta, \theta') = \frac{1}{n} \log \left[u \frac{q(\theta'|\theta)p(\theta)}{q(\theta|\theta')p(\theta')} \right].$$

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

The pseudocode of MH is given in Figure 1, unusually formulated using the expression (3). The advantage of this presentation is that it clearly outlines that the accept/reject step of MH requires checking whether or not (3) holds.

```
MH(p(x|θ), p(θ), q(θ'|θ), θ_0, N_iter, X)
for k ← 1 to N_iter
  θ ← θ_{k-1}
  θ' ~ q(.|θ), u ~ U_(0,1),
  ψ(u, θ, θ') ← (1/n) log [ u p(θ)q(θ'|θ) / p(θ')q(θ|θ') ]
  Λ_n(θ, θ') ← (1/n) Σ_{i=1}^n log [ p(x_i|θ') / p(x_i|θ) ]
  if Λ_n(θ, θ') > ψ(u, θ, θ')
    θ_k ← θ'   ▷ Accept
  else θ_k ← θ   ▷ Reject
return (θ_k)_{k=1,...,N_iter}
```

Figure 1. The pseudocode of the MH algorithm targeting the posterior $\pi(\theta) \propto p(x_1, ..., x_n|\theta)p(\theta)$. The formulation singling out $\Lambda_n(\theta, \theta')$ departs from conventions (Robert & Casella, 2004, Chapter 7.3) but serves the introduction of our main algorithm MHSUBLHD in Figure 3.

So as to save computational efforts, we would like to be able to decide whether (3) holds using only a Monte Carlo approximation of $\Lambda_n(\theta, \theta')$ based on a subset of the data. There is obviously no hope to be able to guarantee that we will take the correct decision with probability 1 but we would like to control the probability of taking an erroneous decision. Korattikara et al. (2014) propose in a similar large datasets context to control this error using an approximate confidence interval for the Monte Carlo estimate. Similar ideas have actually appeared earlier in the operations research literature. Bulgak & Sanders (1988), Alkhamis et al. (1999) and Wang & Zhang (2006) consider maximizing a target distribution whose logarithm is given by an intractable expectation; in the large dataset scenario this expectation is w.r.t the empirical measure of the data. They propose to perform this maximization using simulated annealing, a non-homogeneous version of MH. This is implemented practically by approximating the MH ratio $\log \pi(\theta')/\pi(\theta)$ through Monte Carlo and by determining an approximate confidence interval for the resulting estimate; see also (Singh et al., 2012) for a similar idea developed in the context of inference in large-scale factor graphs. All these approaches rely on approximate confidence intervals so they do not allow to control rigorously the approximation error. Moreover, the use of approximate confidence intervals can yield seriously erroneous inference results as demonstrated in Section 4.1.

The method presented in this paper is a more robust alternative to these earlier proposals, which can be analyzed theoretically and whose properties can be better quantified. As shown in Section 2, it is possible to devise an adaptive sampling strategy which guarantees that we take the correct decision, i.e. whether (3) holds or not, with at worst a *user-specified* maximum probability of error. This sampling strategy allows us to establish in Section 3 various quantitative convergence results for the associated Markov kernel. In Section 4, we compare our approach to the one by Korattikara et al. (2014) on a toy example and demonstrate the performance of our methodology on a large-scale Bayesian logistic regression problem. All proofs are in the supplemental paper.

2. A Metropolis-Hastings algorithm with subsampled likelihoods

In this section, we use concentration bounds so as to obtain exact confidence intervals for Monte Carlo approximations of the log likelihood ratio (4). We then show how such bounds can be exploited so as to build an adaptive sampling strategy with desired guarantees.

2.1. MC approximation of the log likelihood ratio

Let $\theta, \theta' \in \Theta$. For any integer $t \geq 1$, a Monte Carlo approximation $\Lambda_t^*(\theta, \theta')$ of $\Lambda_n(\theta, \theta')$ is given by

$$\Lambda_t^*(\theta, \theta') = \frac{1}{t} \sum_{i=1}^{t} \log \left[\frac{p(x_i^*|\theta')}{p(x_i^*|\theta)} \right], \tag{5}$$

where $x_1^*, \ldots, x_t^*$ are drawn uniformly over $\{x_1, \ldots, x_n\}$ *without replacement*.

We can quantify the precision of our estimate $\Lambda_t^*(\theta, \theta')$ of $\Lambda_n(\theta, \theta')$ through concentration inequalities, i.e., a statement that for $\delta_t > 0$,

$$\mathbb{P}(|\Lambda_t^*(\theta, \theta') - \Lambda_n(\theta, \theta')| \leq c_t) \geq 1 - \delta_t, \tag{6}$$

for a given c_t. Hoeffding's inequality without replacement Serfling (1974), for instance, uses

$$c_t = C_{\theta,\theta'} \sqrt{\frac{2(1 - f_t^*) \log(2/\delta_t)}{t}}, \tag{7}$$

where

$$C_{\theta,\theta'} = \max_{1 \leq i \leq n} |\log p(x_i|\theta') - \log p(x_i|\theta)| \tag{8}$$

and $f_t^* = \frac{t-1}{n}$ is approximately the fraction of used samples. The term $(1 - f_t^*)$ in (7) decreases to $\frac{1}{n}$ as the number t of samples used approaches n, which is a feature of bounds corresponding to sampling without replacement. Let us add that $C_{\theta,\theta'}$ typically grows slowly with n. For instance, if the likelihood is Gaussian, then $C_{\theta,\theta'}$ is proportional to $\max_{i=1}^n |x_i|$, so that if the data actually were

sampled from a Gaussian, $C_{\theta,\theta'}$ would grow in $\sqrt{\log(n)}$ (Cesa-Bianchi & Lugosi, 2009, Lemma A.12).

If the empirical standard deviation $\hat{\sigma}_t$ of $\{\log p(x_i|\theta') - \log p(x_i|\theta)\}$ is small, a tighter bound known as the empirical Bernstein bound (Audibert et al., 2009)

$$c_t = \hat{\sigma}_t \sqrt{\frac{2\log(3/\delta_t)}{t}} + \frac{6C_{\theta,\theta'}\log(3/\delta_t)}{t}, \qquad (9)$$

applies. While the bound of Audibert et al. (2009) originally covers the case where the x_i^* are drawn *with replacement*, it was early remarked (Hoeffding, 1963) that Chernoff bounds, such as the empirical Bernstein bound, still hold when considering sampling without replacement. Finally, we will also consider the recent Bernstein bound of Bardenet & Maillard (2013, Theorem 3), designed specifically for the case of sampling without replacement.

2.2. Stopping rule construction

The concentration bounds given above are helpful as they can allow us to decide whether (3) holds or not. Indeed, on the event $\{|\Lambda_t^*(\theta,\theta') - \Lambda_n(\theta,\theta')| \leq c_t\}$, we can decide whether or not $\Lambda_n(\theta,\theta') > \psi(u,\theta,\theta')$ if $|\Lambda_t^*(\theta,\theta') - \psi(u,\theta,\theta')| > c_t$ additionally holds. This is illustrated in Figure 2. Combined to the concentration inequality (6), we thus take the correct decision with probability at least $1-\delta_t$ if $|\Lambda_t^*(\theta,\theta') - \psi(u,\theta,\theta')| > c_t$. In case $|\Lambda_t^*(\theta,\theta') - \psi(u,\theta,\theta')| \leq c_t$, we want to increase t until the condition$|\Lambda_t^*(\theta,\theta') - \psi(u,\theta,\theta')| > c_t$ is satisfied.

Let $\delta \in (0,1)$ be a user-specified parameter. We provide a construction which ensures that at the first random time T such that $|\Lambda_T^*(\theta,\theta') - \psi(u,\theta,\theta')| > c_T$, the correct decision is taken with probability at least $1-\delta$. This *adaptive stopping rule* adapted from (Mnih et al., 2008) is inspired by bandit algorithms, Hoeffding races (Maron & Moore, 1993) and procedures developed to scale up boosting algorithms to large datasets (Domingo & Watanabe, 2000). Formally, we set the stopping time

$$T = n \wedge \inf\{t \geq 1 : |\Lambda_t^*(\theta,\theta') - \psi(u,\theta,\theta')| > c_t\}, \quad (10)$$

where $a \wedge b$ denotes the minimum of a and b. In other words, if the infimum in (10) is larger than n, then we stop as our sampling without replacement procedure ensures $\Lambda_n^*(\theta,\theta') = \Lambda_n(\theta,\theta')$. Letting $p > 1$ and selecting $\delta_t = \frac{p-1}{pt^p}\delta$, we have $\sum_{t\geq 1}\delta_t \leq \delta$. Setting $(c_t)_{t\geq 1}$ such that (6) holds, the event

$$\mathcal{E} = \bigcap_{t\geq 1}\{|\Lambda_t^*(\theta,\theta') - \Lambda_n(\theta,\theta')| \leq c_t\} \qquad (11)$$

has probability larger than $1-\delta$ under sampling without replacement by a union bound argument. Now by definition of T, if $\mathcal{E}$ holds then $\Lambda_T^*(\theta,\theta')$ yields the correct decision, as pictured in Figure 2.

A slight variation of this procedure is actually implemented in practice; see Figure 3. The sequence (δ_t) is decreasing, and each time we check in Step 19 whether or not we should break out of the **while** condition, we have to use a smaller δ_t, yielding a smaller c_t. Every check of Step 19 thus makes the next check less likely to succeed. Thus, it appears natural not to perform Step 19 systematically after each new x_i^* has been drawn, but rather draw several new subsamples x_i^* between each check of Step 19. This is why we introduce the variable t_{look} is Steps 6, 16, and 17 of Figure 3. This variable simply counts the number of times the check in Step 19 was performed. Finally, as recommended in a related setting in (Mnih et al., 2008; Mnih, 2008), we augment the size of the subsample geometrically by a user-input factor $\gamma > 1$ in Step 18. Obviously this modification does not impact the fact that the correct decision is taken with probability at least $1-\delta$.

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

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 $\quad\quad x_{t+1}^*, \ldots, x_b^* \sim_{\text{w/o repl.}} \mathcal{X} \setminus \mathcal{X}^*$ ▷ *Sample new batch without replacement*
13 $\quad\quad\mathcal{X}^* \leftarrow \mathcal{X}^* \cup \{x_{t+1}^*, \ldots, x_b^*\}$
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 $\quad\quad c \leftarrow 2C_{\theta,\theta'}\sqrt{\frac{(1-f_t^*)\log(2/\delta_{t_{\text{look}}})}{2t}}$
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** $(\theta_k)_{k=1,\ldots,N_{\text{iter}}}$

Figure 3. Pseudocode of the MH algorithm with subsampled likelihoods. Step 16 uses a Hoeffding bound, but other choices of concentration inequalities are possible. See main text for details.

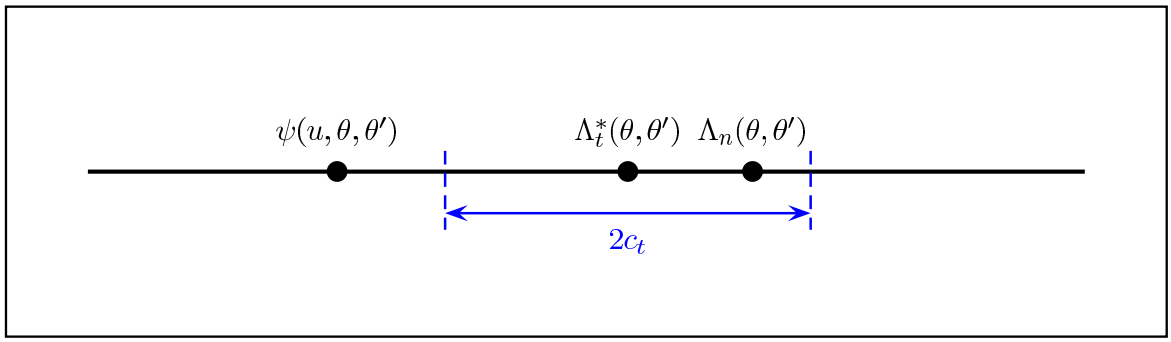


Figure 2. Schematic view of the acceptance mechanism of MHSUBLHD given in Figure 3: if $|\Lambda_t^*(\theta,\theta') - \psi(u,\theta,\theta')| > c_t$, then MHSUBLHD takes the acceptance decision based on $\Lambda_t^*(\theta,\theta')$, without requiring to compute $\Lambda_n(\theta,\theta')$.

3. Analysis

In Section 3.1 we provide an upper bound on the total variation norm between the iterated kernel of the approximate MH kernel and the target distribution π of MH applied to the full dataset, while Section 3.2 focuses on the number T of subsamples required by a given iteration of MHSUBLHD. We establish a probabilistic bound on T and give a heuristic to determine whether a user can expect a substantial gain in terms of number of samples needed for the problem at hand.

3.1. Properties of the approximate transition kernel

For $\theta, \theta' \in \Theta$, we denote by

$$P(\theta, d\theta') = \alpha(\theta,\theta')q(\theta'|\theta)d\theta' + \delta_\theta(d\theta')\left(1 - \int \alpha(\theta,\vartheta)q(\vartheta|\theta)d\vartheta\right) \quad (12)$$

the "ideal" MH kernel targeting π with proposal q, where the acceptance probability $\alpha(\theta,\theta')$ is defined in (1). Denote the acceptance probability of MHSUBLHD in Figure 3 by

$$\tilde{\alpha}(\theta,\theta') = \mathbb{E}1_{\{\Lambda_T^*(\theta,\theta') > \psi(u,\theta,\theta')\}}, \quad (13)$$

where the expectation in (13) is with respect to $u \sim \mathcal{U}_{(0,1)}$ and the variables T and $x_1^*, \ldots, x_T^*$ described in Section 2. Finally, denote by $\tilde{P}$ the MHSUBLHD kernel, obtained by substituting $\tilde{\alpha}$ to α in (12). The following Lemma states that the absolute difference between α and $\tilde{\alpha}$ is bounded by the user-defined parameter $\delta > 0$.

Lemma 3.1. *For any $\theta, \theta' \in \Theta$, we have $|\alpha(\theta,\theta') - \tilde{\alpha}(\theta,\theta')| \leq \delta$.*

Lemma 3.1 can be used to establish Proposition 3.2, which states that the chain output by the algorithm MHSUBLHD in Figure 3 is a *controlled* approximation to the original target π. For any signed measure ν on $(\Theta, \mathcal{B}(\Theta))$, let $\|\nu\|_{\text{TV}} = \frac{1}{2}\sup_{f:\Theta\to[-1,1]} \nu(f)$ denote the total variation norm, where $\nu(f) = \int_\Theta f(\theta)\nu(d\theta)$. For any Markov kernel Q on $(\Theta, \mathcal{B}(\Theta))$, we denote by Q^k be the k-th iterate kernel defined by induction for $k \geq 2$ through $Q^k(\theta, d\theta') = \int_\Theta Q(\theta, d\vartheta)Q^{k-1}(\vartheta, d\theta')$ with $Q^1 = Q$.

Proposition 3.2. *Let P be uniformly geometrically ergodic, i.e., there exists an integer m and a probability measure ν on $(\Theta, \mathcal{B}(\Theta))$ such that for all $\theta \in \Theta$, $P^m(\theta,\cdot) \geq (1-\rho)\nu(\cdot)$. Hence there exists $A < \infty$ such that*

$$\forall\theta \in \Theta, \forall k > 0,\ \|P^k(\theta,\cdot) - \pi\|_{TV} \leq A\rho^{\lfloor k/m \rfloor}. \quad (14)$$

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

$$\|\tilde{P}^k(\theta,\cdot) - \tilde{\pi}\|_{TV} \leq B[1 - (1-\delta)^m(1-\rho)]^{\lfloor k/m \rfloor}. \quad (15)$$

Furthermore, $\tilde{\pi}$ satisfies

$$\|\pi - \tilde{\pi}\|_{TV} \leq \frac{Am\delta}{1-\rho}. \quad (16)$$

One may obtain tighter bounds and ergodicity results by weakening the uniform geometric ergodicity assumption and using recent results on perturbed Markov kernels (Ferré et al., 2013), but this is out of the scope of this paper.

3.2. On the stopping time T

3.2.1. A BOUND FOR FIXED θ, θ'

The following Proposition gives a probabilistic upper bound for the stopping time T, conditionally on $\theta, \theta' \in \Theta$ and $u \in [0,1]$ and when c_t is defined by (7). A similar bound holds for the empirical Bernstein bound in (9).

Proposition 3.3. *Let $\delta > 0$ and $\delta_t = \frac{p-1}{pt^p}\delta$. Let $\theta, \theta' \in \Theta$ such that $C_{\theta,\theta'} \neq 0$ and $u \in [0,1]$. Let*

$$\Delta = \frac{|\Lambda_n(\theta,\theta') - \psi(u,\theta,\theta')|}{2C_{\theta,\theta'}} \quad (17)$$

and assume $\Delta \neq 0$. Then if $p > 1$, with probability at least $1-\delta$,

$$T \leq \left\lceil \frac{4}{\Delta^2} \left\{ p \log \left[\frac{4p}{\Delta^2} \right] + \log \left[\frac{2p}{\delta(p-1)} \right] \right\} \right\rceil \vee 1. \quad (18)$$

The relative distance Δ in (18) characterizes the difficulty of the step. Intuitively, at equilibrium, i.e., when $(\theta, \theta') \sim \pi(\theta) q(\theta'|\theta)$ and $u \sim \mathcal{U}_{[0,1]}$, if the log likelihood $\log p(x|\theta)$ is smooth in θ, the proposal could be chosen so that $\Lambda_n(\theta, \theta')$ has positive expectation and a small variance, thus leading to high values of Δ and small values of T.

3.2.2. A HEURISTIC AT EQUILIBRIUM

Integrating (18) with respect to θ, θ' to obtain an informative quantitative bound on the average number of samples required by MHSUBLHD at equilibrium would be desirable but proved difficult. However the following heuristic can help the user figure out whether our algorithm will yield important gains for a given problem. For large n, standard asymptotics (van der Vaart, 2000) yield that the log likelihood is approximately a quadratic form

$$\log p(x|\theta) \approx -(\theta - \theta^*)^T H_n (\theta - \theta^*)$$

with H_n of order n. Assume the proposal $q(\cdot|\theta)$ is a Gaussian random walk $\mathcal{N}(\cdot|\theta, \Gamma)$ of covariance Γ, then the expected log likelihood ratio under $\pi(\theta)q(\theta'|\theta)$ is approximately Trace$(H_n \Gamma)$. According to (Roberts & Rosenthal, 2001), an efficient random walk Metropolis requires Γ to be of the same order as H_n^{-1}, that is, of order $1/n$. Finally, the expected $\Lambda_n(\theta, \theta')$ at equilibrium is of order $1/n$, and can thus be compared to $\psi(u, \theta, \theta') = \log(u)/n$ in Line 19 of MHSUBLHD in Figure 3. The RHS of the first inequality in Step 19 is the concentration bound c_t, which has a leading term in $\hat{\sigma}_t/\sqrt{t}$ in the case of (9). In the applications we consider in Section 4, $\hat{\sigma}_t$ is typically proportional to $\|\theta - \theta'\|$, which is of order $\sqrt{n}$ since $\Gamma \approx H_n^{-1}$. Thus, to break out of the *while* loop in Line 19, we need $t \propto n$. At equilibrium, we thus should not expect gains of several orders of magnitude: gains are fixed by the constants in the proportionality relations above, which usually depend on the empirical distribution of the data. We provide a detailed analysis for a simple example in Section 4.3.

4. Experiments

All experiments were conducted using the empirical Bernstein-Serfling bound of Bardenet & Maillard (2013), which revealed equivalent to the empirical Bernstein bound in (9), and much tighter in our experience with MHSUBLHD than Hoeffding's bound in (7). All MCMC runs are adaptive Metropolis (Haario et al., 2001; Andrieu & Thoms, 2008) with target acceptance 25% when the dimension is larger than 2 and 50% else (Roberts & Rosenthal, 2001). Hyperparameters of MHSUBLHD were set to $p = 2$, $\gamma = 2$, and $\delta = 0.01$. The first two were found to work well with all experiments. We found empirically that the algorithm is very robust to the choice of δ.

4.1. On the use of asymptotic confidence intervals

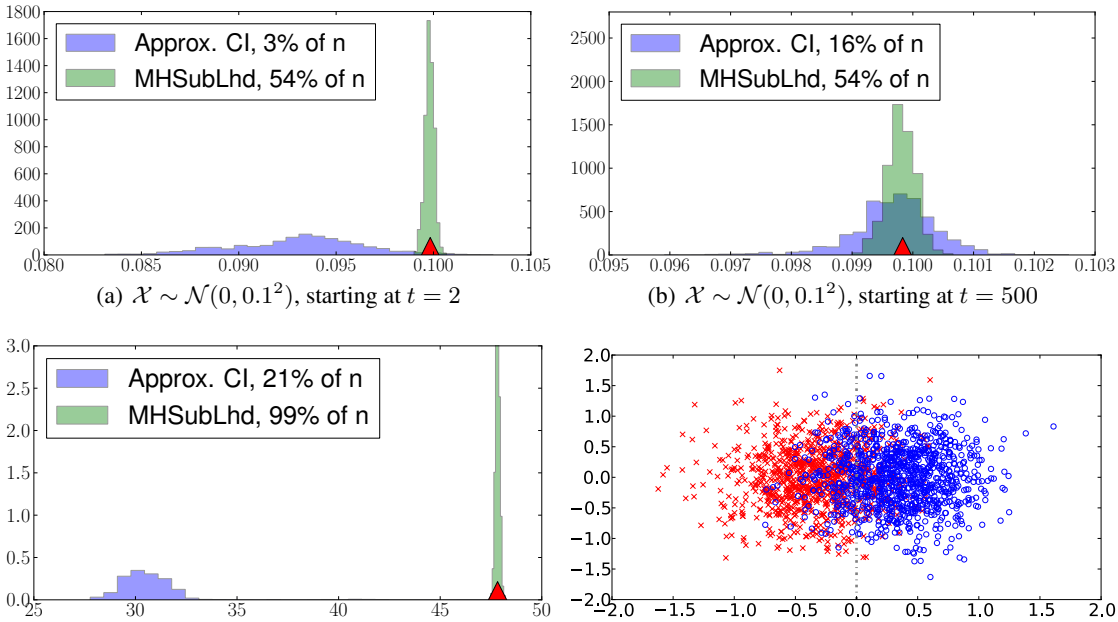
MCMC algorithms based on subsampling and asymptotic confidence intervals experimentally lead to efficient optimization procedures (Bulgak & Sanders, 1988; Alkhamis et al., 1999; Wang & Zhang, 2006), and perform well in terms of classification error when used, e.g., in logistic regression (Korattikara et al., 2014). However, in terms of approximating the original posterior, they come with no guarantee and can provide unreliable results.

To illustrate this, we consider the following setting. $\mathcal{X}$ is a synthetic sample of size 10^5 drawn from a Gaussian $\mathcal{N}(0, 0.1^2)$, and we estimate the parameters μ, σ of a $\mathcal{N}(\mu, \sigma^2)$ model, with flat priors. Analytically, we know that the posterior has its maximum at the empirical mean and variance of $\mathcal{X}$. Running the approximate MH algorithm of Korattikara et al. (2014), using a level for the test $\epsilon = 0.05$, and starting each iteration by looking at $t = 2$ points so as to be able to compute the variance of the log likelihood ratios, leads to the marginal of σ shown in Figure 4(a).

The empirical variance of $\mathcal{X}$ is denoted by a red triangle, and the maximum of the marginal is off by 7% from this value. Relaunching the algorithm, but starting each iteration with a minimum $t = 500$ points, leads to better agreement and a smaller support for the marginal, as depicted in Figure 4(b). Still, $t = 500$ works better for this example, but fails dramatically if $\mathcal{X}$ are samples from a lognormal log$\mathcal{N}(0, 2)$, as depicted in Figure 4(c). The asymptotic regime, in which the studentized statistic used in (Korattikara et al., 2014) actually follows a Student distribution, depends on the problem at hand and is left to the user to specify. In each of the three examples of Figure 4, our algorithm produces significantly better estimates of the marginal, though at the price of a significantly larger average number of samples used per MCMC iteration. In particular, the case $\mathcal{X} \sim$ log$\mathcal{N}(0, 2)$ in Figure 4(c) essentially requires to use the whole dataset.

4.2. Large-scale Bayesian logistic regression

In logistic regression, an accurate approximation of the posterior is often needed rather than minimizing the classification error, for instance, when performing Bayesian variable selection. This makes logistic regression for large datasets a natural application for our algorithm, since the constant $C_{\theta,\theta'}$ in concentration inequalities such as (9) can



(a) $\mathcal{X} \sim \mathcal{N}(0, 0.1^2)$, starting at $t = 2$

(b) $\mathcal{X} \sim \mathcal{N}(0, 0.1^2)$, starting at $t = 500$

(c) $\mathcal{X} \sim \ln\mathcal{N}(0, 2)$, starting at $t = 500$

(d) train error vs. time

Figure 4. (a,b,c) Estimates of the marginal posteriors of σ obtained respectively by the algorithm of Korattikara et al. (2014) using approximate confidence intervals and our algorithm MHSUBLHD given in Figure 3, for $\mathcal{X}$ sampled from each of the distributions indicated below the plots, and with different starting points for the number t of samples initially drawn from $\mathcal{X}$ at each MH iteration. On each plot, a red triangle indicates the true maximum of the posterior, and the legend indicates the proportion of $\mathcal{X}$ used on average by each algorithm. (d) The synthetic dataset used in Section 4.2.2. The dash-dotted line indicates the Bayes classifier.

be computed as follows. The log likelihood

$$\log p(y|x, \theta) = -\log(1 + e^{-\theta^T x}) - (1 - y)\theta^T x \quad (19)$$

is L-Lipschitz in θ with $L = \|x\|$, so that we can set

$$C_{\theta,\theta'} = \|\theta - \theta'\| \max_{1 \le j \le n} \|x_j\|.$$

We expect the Lipschitz inequality to be tight as (19) is almost linear in θ.

4.2.1. THE *covtype* DATASET

We consider the dataset *covtype.binary*[1] 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 consider a simple variant of the classification problem using the first $q = 2$ attributes only. We use the preprocessing and Cauchy prior recommended by Gelman et al. (2008).

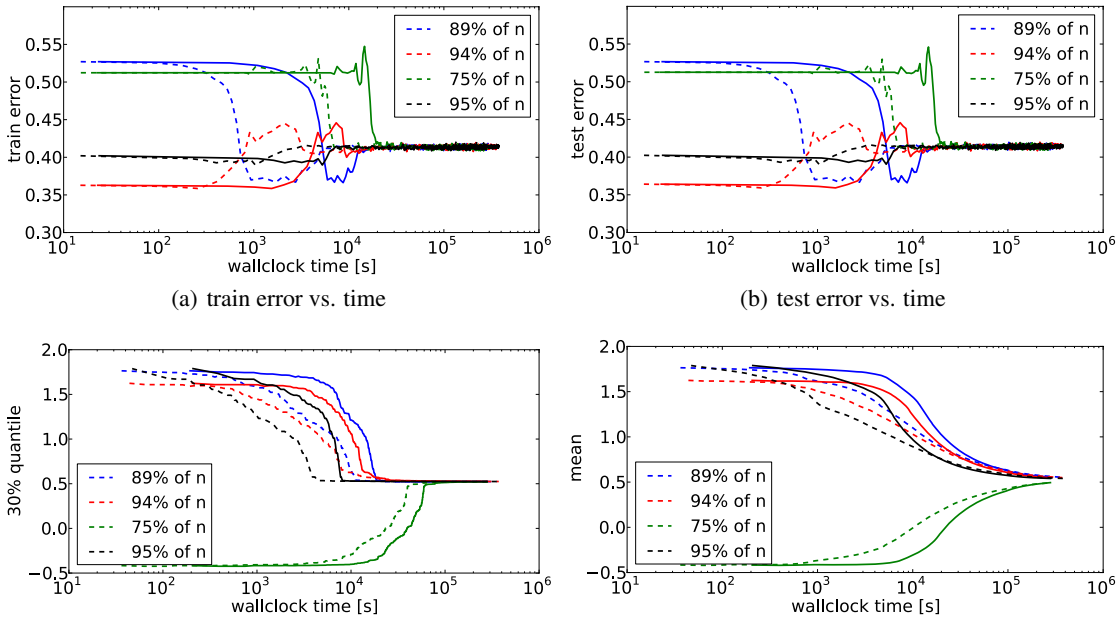
We draw four random starting points and launch independent runs of both the traditional MH in Figure 1 and our MHSUBLHD in Figure 3 at each of these four starting points. Figure 5 shows the results: plain lines indicate traditional MH runs, while dashed lines indicate runs of MHSUBLHD. Figures 5(c) and 5(d) confirm that MHSUBLHD accurately approximates the target posterior. In all Figures 5(a) to 5(d), MHSUBLHD reproduces the behaviour of MH, but converges up to 3 times faster. However, the most significant gains in number of samples used happen in the initial transient phase. This allows fast progress of the chains towards the mode but, once in the mode, the average number of samples required by MHSUBLHD is close to n. We observed the same behaviour when considering all $q = 10$ quantitative attributes of the dataset, as depicted by the train error in Figure 7(a).

4.2.2. SYNTHETIC DATA

To investigate the rôle of n in the gain, we generate a 2D binary classification dataset of size $n = 10^7$. Given the label, both classes are sampled from unit Gaussians centered on the x-axis, and a subsample of $\mathcal{X}$ is shown in Figure 4(d).

The results are shown in Figure 6. The setting appears more favorable than in Section 4.2.1, and MHSUBLHD chains converge up to 5 times faster. The average number of samples used is smaller, but it is still around 70% after the tran-

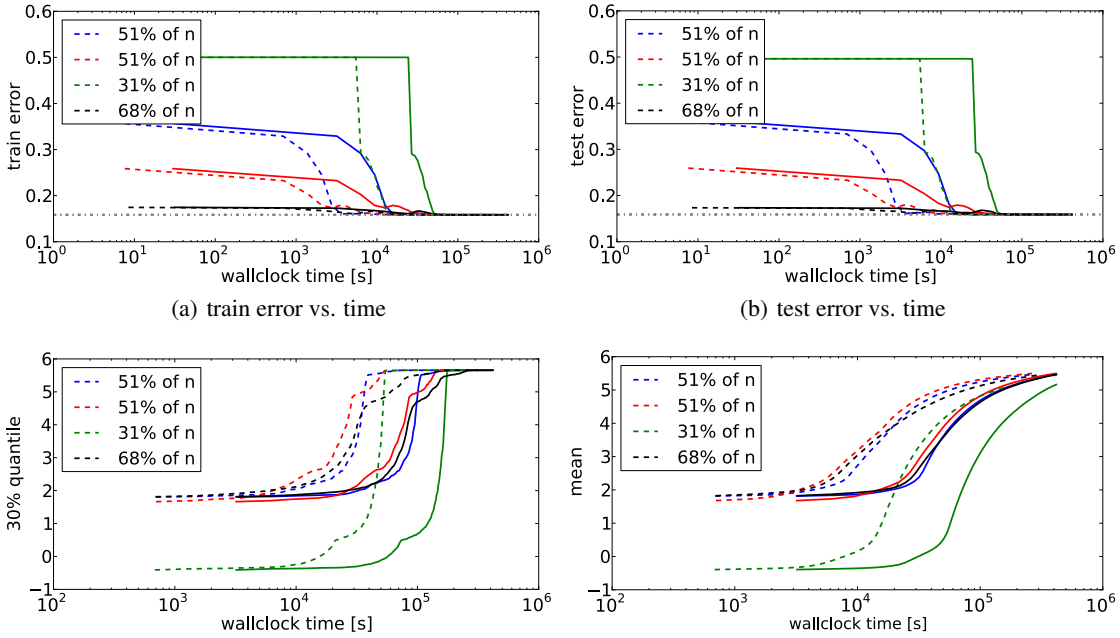
[1] available at http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html



(a) train error vs. time (b) test error vs. time

(c) 30% quantile of the chain vs. time (d) mean of the chain vs. time

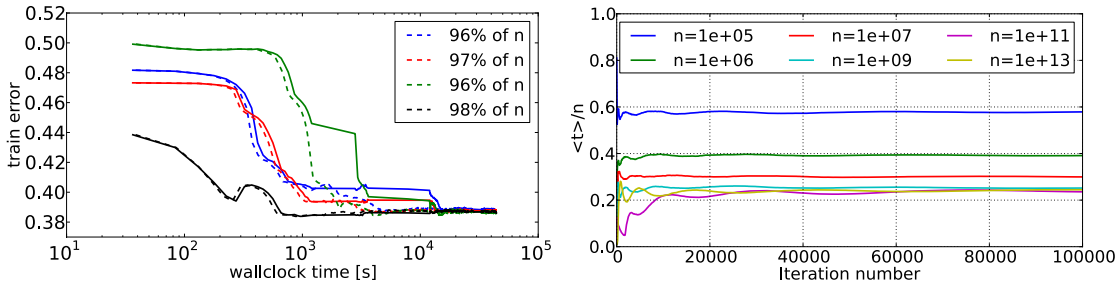
Figure 5. Results of 4 independent runs of MH (plain lines) and MHSUBLHD (dashed lines) for the 2 first attributes of the *covtype* dataset. The legend indicates the average number of samples required as a proportion of n.



(a) train error vs. time (b) test error vs. time

(c) 30% quantile of the chain vs. time (d) mean of the chain vs. time

Figure 6. Results of 4 independent runs of MH (plain lines) and MHSUBLHD (dashed lines) for the synthetic dataset described in Section 4.2.2. The legend indicates the average number of samples required as a proportion of n. On Figures 6(a) and 6(b), a dash-dotted line indicates the error obtained by the Bayes classifier.



(a) train error vs. time (b)

Figure 7. (a) Results of 4 independent runs of MH (plain lines) and MHSUBLHD (dashed lines) for the 10 quantitative attributes of the *covtype* dataset. The legend indicates the average number of samples required as a proportion of n. (b) Running proportion of samples needed vs. iteration number for different values of n, for the Gaussian experiment of Section 4.3.

sient phase for all approximate chains.

4.3. A Gaussian example

To further investigate when gains are made at equilibrium, we now consider inferring the mean θ of a $\mathcal{N}(\theta, 1)$ model, using a sample $\mathcal{X} \sim \mathcal{N}(\frac{1}{2}, \sigma_{\mathcal{X}}^2)$ of size n. Although simple, this setting allows us analytic considerations. The log likelihood ratio is

$$\log \frac{p(x|\theta')}{p(x|\theta)} = (\theta' - \theta)(x - \frac{\theta + \theta'}{2})$$

so that we can set

$$C_{\theta,\theta'} = 2|\theta' - \theta| \left(\max_{1 \leq i \leq n} |x_i| + \frac{|\theta + \theta'|}{2} \right).$$

We also remark that

$$\sqrt{\mathbb{V}_x \log \frac{p(x|\theta')}{p(x|\theta)}} = |\theta - \theta'| \sigma_{\mathcal{X}}. \tag{20}$$

Under the equilibrium assumptions of Section 3.2.2, $|\theta - \theta'|$ is of order $n^{-1/2}$, so that the leading term $t^{-1/2}\hat{\sigma}_t$ of the concentration inequality (9) is of order $\sigma_{\mathcal{X}} n^{-1/2} t^{-1/2}$. Thus, to break out of the *while* loop in Line 19 in Figure 3, we need $t \propto \sigma_{\mathcal{X}}^2 n$. In a nutshell, larger gains are thus to be expected when data are clustered in terms of log likelihood.

To illustrate this phenomenon, we set $\sigma_{\mathcal{X}} = 0.1$. To investigate the behavior at equilibrium, all runs were started at the mode of a subsampled likelihood, using a proposal covariance matrix proportional to the covariance of the target. In Figure 7(b), we show the running average number of samples needed for 6 runs of MHSUBLHD, with n ranging from 10^5 to 10^{15}. With increasing n, the number of samples needed progressively drops to 25% of the total n. This is satisfying, as the number of samples required at equilibrium should be less than 50% to actually improve on usual MH, since a careful implementation of MH in Figure 1 only requires to evaluate one single full likelihood per iteration, while methods based on subsampling require two.

5. Conclusion

We have presented an approximate MH algorithm to perform Bayesian inference for large datasets. This is a robust alternative to the technique in (Korattikara et al., 2014), and this robustness comes at an increased computational price. We have obtained theoretical guarantees on the resulting chain, including a user-controlled error in total variation, and we have demonstrated the methodology on several applications. Experimentally, the resulting approximate chains achieve fast burn-in, requiring on average only a fraction of the full dataset. At equilibrium, the performance of the method is strongly problem-dependent. Loosely speaking, if the expectation w.r.t. $\pi(\theta)q(\theta'|\theta)$ of the variance of $\log p(x|\theta')/p(x|\theta)$ w.r.t. to the empirical distribution of the observations is low, then one can expect significant gains. If this expectation is high, then the algorithm is of limited interest as the Monte Carlo estimate $\Lambda_t^*(\theta, \theta')$ requires many samples t to reach a reasonable variance. It would be desirable to use variance reduction techniques but this is highly challenging in this context.

Finally, the algorithm and analysis provided here can be straightforwardly extended to scenarios where π is such that $\log \pi(\theta')/\pi(\theta)$ is intractable, as long as a concentration inequality for a Monte Carlo estimator of this log ratio is available. For models with an intractable likelihood, it is often possible to obtain such an estimator with low variance, so the methodology discussed here may prove useful.

ACKNOWLEDGMENTS

Rémi Bardenet acknowledges his research fellowship through the 2020 Science programme, funded by EPSRC grant number EP/I017909/1. Chris Holmes is supported by an EPSRC programme grant and a Medical Research Council Programme Leader's award.

References

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

Andrieu, C. and Thoms, J. A tutorial on adaptive MCMC. *Statistics and Computing*, 18:343–373, 2008.

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

Bardenet, R. and Maillard, O.-A. Concentrations inequalities for sampling without replacement. preprint, 2013. URL arxiv.org/abs/1309.4029.

Bulgak, A. A. and Sanders, J. L. 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*, 1988.

Cesa-Bianchi, N. and Lugosi, G. Combinatorial bandits. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

Collobert, R., Bengio, S., and Bengio, Y. A parallel mixture of SVMs for very large scale problems. *Neural Computation*, 14(5):1105–1114, 2002.

Domingo, C. and Watanabe, O. MadaBoost: a modification of AdaBoost. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory (COLT)*, 2000.

Ferré, D., L., Hervé, and Ledoux, J. Regular perturbation of V-geometrically ergodic Markov chains. *Journal of Applied Probability*, 50(1):184–194, 2013.

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

Haario, H., Saksman, E., and Tamminen, J. An adaptive Metropolis algorithm. *Bernoulli*, 7:223–242, 2001.

Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

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

Maron, O. and Moore, A. Hoeffding races: Accelerating model selection search for classification and function approximation. In *Advances in Neural Information Processing Systems*. The MIT Press, 1993.

Mnih, V. Efficient stopping rules. Master's thesis, University of Alberta, 2008.

Mnih, V., Szepesvári, Cs., and Audibert, J.-Y. Empirical Bernstein stopping. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 2008.

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

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

Serfling, R. J. Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, 2 (1):39–48, 1974.

Singh, S., Wick, M., and McCallum, A. 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*, 2012.

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

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