Hessian-based Markov Chain Monte-Carlo Algorithms

Yuan Qi and Thomas P. Minka[†]
Media Lab, MIT, Cambridge, MA, 02139
yuanqi@media.mit.edu
Department of Statistics of CMU, Pittsburgh, PA, 15123[†]
minka@stat.cmu.edu[†]

September 12, 2002

Abstract

In this paper, we propose two efficient Markov chain Monte-Carlo sampling methods, namely, the Hessian-based Metropolis-Hastings (HMH) and adaptive multiple importance-try (AMIT) algorithms. HMH utilizes Newton's optimization method to generate transition distributions in a Metropolis-Hastings sampling scheme. By introducing learning rates into Newton's method, HMH reduces the risk of having samples stuck in a local region where it is not good to approximate the target distribution by a Gaussian. We also provide a method to efficiently implement HMH for high-dimensional data. For more effective exploration of the adaptive proposal distributions obtained by HMH, AMIT combines HMH with the multiple-try Metropolis (MTM) algorithm, proposed by Liu et al [5]. We compare HMH and AMIT samplers with Gibbs and optimal marginal data augmentation (DA) samplers, proposed by van Dyk and Meng [7], on a probit regression problem. In the experiment, the AMIT sampler outperforms the other samplers. Though only tested on a probit model, these new sampling methods can be easily applied to any generalized linear model and other models for which we can efficiently compute or approximate Hessian matrices.

1 Introduction

Markov chain Monte-Carlo methods allows easy implementation of Bayesian inference systems and helps Bayesian techniques play a more and more important role in real world applications.

In this paper, we propose two efficient Markov chain Monte-Carlo sampling methods, namely, the Hessian-based Metropolis-Hastings (HMH) and adaptive multiple importance-try (AMIT) algorithms. HMH utilizes Newton's optimization method to generate transition distributions in a Metropolis-Hastings sampling scheme. By introducing learning rates into Newton's method, HMH reduces the risk of having samples stuck in a local region where it is not good to approximate the target distribution by a Gaussian. We also provide a method to efficiently implement HMH for high-dimensional data. Recently[3], Geweke and Tanizaki propose a sampling method that is similar to HMH. But their sampler does not have any learning rate and only deals with one dimensional data.

For more effective exploration of the adaptive proposal distributions obtained by HMH, AMIT combines HMH with the multiple-try Metropolis (MTM) algorithm, proposed by Liu et al [5]. We compare HMH and AMIT samplers with Gibbs and optimal marginal data augmentation (DA) samplers, proposed by van Dyk and Meng [7], on a probit regression problem. In the experiment, the AMIT sampler outperforms the other samplers. Though only tested on a probit model, these new sampling methods can be easily applied to any generalized linear model and other models for which we can efficiently compute or approximate Hessian matrices.

The paper is organized as follows. First, section 2 describes generalized linear models and reviews Gibbs and DA samplers for probit regression. Then section 3 presents HMH and its efficient implementation for generalized linear models, followed by section 4 that presents AMIT samplers. Section 5 compares HMH and AMIT samplers with Gibbs, DA, and MTM samplers on a probit regression problem, demonstrating the efficiency of the new samplers. Finally, section 6 concludes the paper and points out the directions for future work.

2 Generalized linear Model, Gibbs and DA Samplers

Given a data set

$$D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}, \tag{1}$$

$$\mathbf{y} = [y_1, \ldots, y_n]_{1\times n} \tag{2}$$

$$\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]_{d\times n} \tag{3}$$

where $y_i \in \{-1, 1\}$, the likelihood of a generalized linear model for $\mathbf{w}$ can be written as

$$p(\mathbf{y}|\mathbf{w}, \mathbf{x}) = \prod_i p(y_i|\mathbf{x}_i, \mathbf{w}) = \prod_i \psi(y_i \mathbf{x}_i^{\mathrm{T}} \mathbf{w}) \tag{4}$$

where $\psi(\cdot)$ is a linear or nonlinear link function. For the probit model,the link function $\psi(z)$ is the standard Gaussian cumulative function. The posterior distribution can be computed as follows

$$p(\mathbf{w}|\mathbf{y}, \mathbf{x}) \propto p(\mathbf{y}|\mathbf{w}, \mathbf{x})p(\mathbf{w}),$$

where a non-informative prior distribution is assigned to $\mathbf{w}$ such that $p(\mathbf{w}) \propto 1$.

To construct a Gibbs sampler for probit regression, we first augment $\mathbf{y}$ to $\mathbf{y}_{mboxaug} = \{(y_1, z_1), \dots, (y_n, z_n)\}$. It follows that

$$y_i = \text{sign}(z_i) \tag{5}$$

$$z_i|\mathbf{w} \sim \mathcal{N}(\mathbf{x}_i^{\mathrm{T}}\mathbf{w}, 1) \tag{6}$$

where $\text{sign}(z_i) = 1$ if $z_i \geq 0$, and $\text{sign}(z_i) = -1$ otherwise. Then the Gibbs sampler draws samples as follows [1]:

$$\tilde{\mathbf{w}} = (\mathbf{x}^{\mathrm{T}}\mathbf{x})^{-1}\mathbf{x}^{\mathrm{T}}\mathbf{z} \tag{7}$$

$$\mathbf{w}|\mathbf{y}_{aug}, \mathbf{z} \sim \mathcal{N}(\tilde{\mathbf{w}}, (\mathbf{x}^{\mathrm{T}}\mathbf{x})^{-1}) \tag{8}$$

$$z_i|\mathbf{w}, y_i \sim \mathcal{TN}(\mathbf{x}_i^{\mathrm{T}}\mathbf{w}, 1, y_i), \tag{9}$$

where $\mathbf{z} = [z_1, \dots, z_n]^{\mathrm{T}}$ and $\mathcal{TN}(\mu, \sigma, y_i)$ is a truncated Gaussian distribution with mean μ and variance σ, truncated to be positive if $y_i = 1$ and negative if $y_i = -1$.

Though easy to implement, the Gibbs sampler for probit regression suffers from its slow convergence speed as demonstrated in section 5. The Gibbs sampler actually defines a stochastic mapping for the parameter $\mathbf{w}$ based on the latent variable z_i, similar to the EM algorithm. It is well known that the EM algorithm for probit regression is not efficient, which in turn corresponds to the slow convergence speed of the Gibbs sampler. To increase the EM convergence rate, Liu, Rubin, and Wu introduce variables that are identifiable under the augmented model but not under the observed data model and design a fast EM algorithm for probit regression [4]. Also, Meng and van Dyk propose the working parameter approach to speed up EM-type algorithms [6]. For the purpose of sampling, van Dyk and Meng adopt the EM criterion and design !!!! data-augmentation (DA) samplers for different data models [7]. Among them, the DA sampler for probit regression is given as follows:

$$C = \sum_{i}^{n} (z_i - \mathbf{x}_i^{\mathrm{T}}\tilde{\mathbf{w}})^2 \tag{10}$$

$$\mathbf{w} \sim \mathcal{N}(\sqrt{\frac{\chi_n^2}{C}}\tilde{\mathbf{w}}, (\mathbf{x}^{\mathrm{T}}\mathbf{x})^{-1}) \tag{11}$$

where z_i and $\tilde{\mathbf{w}}$ are defined in equations (9) and (7), and χ_n^2 represents a sample drawn from the Chi-square distribution with the degree of freedom n. As shown by van Dyk and Meng,the DA sampler converges much faster than the plain Gibbs sampler for probit regression, just like a PX-EM algorithm usually converges much quicker than a plain EM algorithm.

While DA samplers enjoy fast convergence speeds, it may be hard to design new efficient DA samplers for new data models though some work has been done to facilitate the design [7].

3 Hessian-based Metropolis-Hastings algorithm

In this section, we propose a new MCMC sampling algorithm, namely, Hessian-based Metropolis-Hastings (HMH) algorithm. By locally approximating the posterior distribution by Gaussians, HMH obtains adaptive proposal distributions, which greatly improve the convergence speed. HMH can be applied to various data models for which we can efficiently compute or approximate their Hessians, such as generalized linear models.

First consider a general Metropolis-Hastings sampler. It samples a new point $\mathbf{w}^\star$ from a jumping distribution $J_t(\mathbf{w}^\star|\mathbf{w}^t)$ at the t^{th} iteration, and then computes the ratio of importance ratios:

$$r = \frac{p(\mathbf{w}^\star|\mathbf{y})/J_t(\mathbf{w}^\star|\mathbf{w}^t)}{p(\mathbf{w}^t|\mathbf{y})/J_t(\mathbf{w}^t|\mathbf{w}^\star)}. \tag{12}$$

With probability $\min(r,1)$, it sets $\mathbf{w}^{t+1} = \mathbf{w}^\star$; otherwise, it keeps the parameter unchanged, i.e., $\mathbf{w}^{t+1} = \mathbf{w}^t$.

Though ideally the jumping distribution $J_t(\mathbf{w}^\star|\mathbf{w}^t)$ would be the target distribution $p(\mathbf{w}^\star|\mathbf{y})$ for all $\mathbf{w}$, such a direct sampling scheme is not possible in most problems. Instead, we can use a local approximation of $p(\mathbf{w}^\star|\mathbf{y})$ that is also easy for sampling to served as the jumping distribution $J_t(\mathbf{w}^\star|\mathbf{w}^t)$. To this end, we borrow the approximation scheme from Newton's optimization method.

To maximize the logarithm of a posterior $p(\mathbf{w}|\mathbf{y})$ over parameter $\mathbf{w}$, Newton's method iteratively approximates $\log p(\mathbf{w}|\mathbf{y})$ by a quadratic function, then maximizes the quadratic approximation to get new parameter $\mathbf{w}^\star$:

$$\mathbf{g} = \frac{\mathrm{d}\log p(\mathbf{w}|\mathbf{y})}{\mathrm{d}\mathbf{w}} \tag{13}$$

$$\mathbf{H} = \frac{\mathrm{d}^2 \log p(\mathbf{w}|\mathbf{y})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^{\mathrm{T}}} \tag{14}$$

$$\mathbf{w}^\star = \mathbf{w} - \mathbf{H}^{-1}\mathbf{g} \tag{15}$$

where $\mathbf{H}$ is the Hessian matrix, and $\mathbf{H}^{-1}\mathbf{g}$ is known as the Newton direction. For the non-informative prior $p(\mathbf{w}) \propto 1$, $\mathbf{H} = \frac{\mathrm{d}^2 \log p(\mathbf{y}|\mathbf{w})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^{\mathrm{T}}}$. If the posterior $p(\mathbf{w}|\mathbf{y})$ is log-concave, $\mathbf{w}^\star$ will converge to the maximum *a posteriori* (MAP) solution by iteratively applying the Newton's updates.

Now instead of maximizing over $\mathbf{w}$, we first construct a Gaussian approximation of $p(\mathbf{w}|\mathbf{y})$ as in the Newton method, and then draw samples by taking the Gaussian approximation as the jumping distribution $J_t(\mathbf{w}^\star|\mathbf{w})$:

$$\mathbf{g}^t = \frac{\mathrm{d}\log p(\mathbf{w}|\mathbf{y})}{\mathrm{d}\mathbf{w}}|_{\mathbf{w}=\mathbf{w}^t} \tag{16}$$

$$\Sigma_t = -\mathbf{H}^{-1}|_{\mathbf{w}=\mathbf{w}^t} \tag{17}$$

$$\mathbf{m}^t = \mathbf{w}^t + \gamma\, \Sigma_t\, \mathbf{g}^t \tag{18}$$

$$J_t(\mathbf{w}^\star|\mathbf{w}^t) = \mathcal{N}(\mathbf{w}^\star \mid \mathbf{m}^t, \Sigma_t) \tag{19}$$

where γ is a learning rate that controls how much $\mathbf{m}^\star$ will change from $\mathbf{w}^t$ along the Newton's direction. By using the Newton direction to update the mean of the jumping distribution, the new sample is more likely drawn from a high probable region.

Note that the parameter γ is used as a learning rate in equation (18). For HMH, if the target distribution can not be well approximated by a Gaussian, a big change of the proposed mean from $\mathbf{w}^t$ to $\mathbf{m}^\star$ will result in a very low acceptance ratio and the Markov chain may be stuck in a local region for a long time. Thus, the learning rate γ is used to avoid this problem. A simple choice of the learning rate is to a number between 0 and 1. This number can be randomly generated. And the algorithm will still be a valid MCMC, since we have zero probability to come back to the same data point in a continuous probability space of $\mathbf{w}$. From another perspective, the random generation of learning rates simply means that the jumping distributions are chosen on the fly rather than in advance. Using a small learning rate helps Markov sampling chains leave a local region where the target distribution is far from a Gaussian.

A more sophisticated method [2] to address the same problem will be introducing a scaling factor into the Hessian so that the new Hessian equals

$$\mathbf{H} + \lambda \mathbf{I}$$

where $\mathbf{H}$ is defined in equation (14), and $\mathbf{I}$ is an identity matrix. We scale λ based on the quality of the quadratic approximation. If the quadratic approximation is not good, we use a large λ; otherwise, we make λ small. This technique is called a model trust region method in the optimization community. In addition, if $\mathbf{H}$ is not positive definite or near zeros, we can make the Hessian positive definite in a reasonable range by imposing a suitable λ. But for generalized linear models that we will discuss later, the Hessian is usually positive definite. In this paper, we only use the learning rate method in our experiments.

Though we have two jumping distributions at each iteration, we only need to compute the variance parameter for the backward jumping distribution except the first iteration. The variance of the next forward jumping distribution is either the variance of the current forward jumping distribution, or that of the current backward distribution, depending on if the new sample is accepted or not.

In sum, the HMH sampling algorithm is described as follows:

1. Draw a starting point $\mathbf{w}^1$, for which $p(\mathbf{w}^1|y) > 0$, from a starting distribution $p_1(\mathbf{w})$. Compute the mean $\mathbf{m}^1$ and the Cholesky factor $\mathbf{R}_1$ of the negative Hessian $\mathbf{H}_1$, for which $\mathbf{R}_1^{\mathrm{T}}\mathbf{R}_1 = -\mathbf{H}_1$, for the first forward jumping distribution.

2. For $t = 2, 3, \ldots$:

   (a) Sample a new point $\mathbf{w}^\star$ from the forward jumping distribution

$J_t(\mathbf{w}^\star|\mathbf{w}^t) = \mathcal{N}(\mathbf{w}^\star \mid \mathbf{m}^t, \Sigma_t)$:

$$\mathbf{c} \sim \mathcal{N}_d(\mathbf{0}, \mathbf{I}) \tag{20}$$

$$\mathbf{w}^\star = \mathbf{w}^t + \mathbf{R}_t^{-1}\mathbf{c}. \tag{21}$$

We can use the variable elimination method to compute $\mathbf{R}_t^{-1}\mathbf{c}$ without inverting $\mathbf{R}_t^{-1}$ directly. This reduces the computation complexity from $O(d^3)$ to $O(d^2)$.

(b) Compute the mean $\mathbf{m}^\star$ and the Hessian $\mathbf{H}_\star$ for the backward jumping distribution $J_t(\mathbf{w}^t|\mathbf{w}^\star) = \mathcal{N}(\mathbf{w}^t \mid \mathbf{m}^\star, \Sigma_\star)$ by switching the time indexes $(\star)$ to (t) in the equations (18) and (17). Then compute the Cholesky factor $\mathbf{R}_\star$ of $-\mathbf{H}_\star$.

(c) Compute the ratio of importance ratios:

$$r = \frac{p(\mathbf{w}^\star|\mathbf{y})/\mathcal{N}(\mathbf{w}^\star \mid \mathbf{m}^t, \Sigma_t)}{p(\mathbf{w}^t|\mathbf{y})/\mathcal{N}(\mathbf{w}^t \mid \mathbf{m}^\star, \Sigma_\star)}. \tag{22}$$

Clearly, given $\mathbf{H}_\star$ and $\mathbf{R}_\star$, we can easily compute the ratio r without any matrix inversion.

(d) With probability $\min(r, 1)$, we set $\mathbf{w}^{t+1} = \mathbf{w}^\star$, $\mathbf{m}^{t+1} = \mathbf{m}^\star$, and $\mathbf{R}_{t+1} = \mathbf{R}_\star$; Otherwise, we set $\mathbf{w}^{t+1} = \mathbf{w}^t$, $\mathbf{m}^{t+1} = \mathbf{m}^t$, and $\mathbf{R}_{t+1} = \mathbf{R}_t$.

Recently [3], Geweke and Tanizaki propose a sampler that is similar to HMH. But their sampler only deals with one dimensional data and does not have any learning rate. As discussed before, without the use of learning rate, HMH has the risk of being stuck in a local region where a quadratic approximation is not good.

In next section, we propose a method to considerably reduce the computation complexity at each sampling iteration for high-dimensional problems.

3.1 HMH for Generalized Linear Models

For generalized linear models, a Hessian matrix has the following form

$$\mathbf{H} = -\mathbf{x}\mathbf{A}\mathbf{x}^{\mathrm{T}} = -\mathbf{x}\mathrm{diag}\{\mathbf{a}\}\mathbf{x}^{\mathrm{T}} \tag{23}$$

where $\mathbf{a} = [a_1, \ldots, a_n]$. For probit models, the diagonal element a_i can be calculated as follows:

$$a_i = \frac{\mathcal{N}(y_i\mathbf{x}_i^{\mathrm{T}}\mathbf{w}|0,1)}{\phi(y_i\mathbf{x}_i^{\mathrm{T}}\mathbf{w})}\left(\frac{\mathcal{N}(y_i\mathbf{x}_i^{\mathrm{T}}\mathbf{w}|0,1)}{\phi(y_i\mathbf{x}_i^{\mathrm{T}}\mathbf{w})} + y_i\mathbf{x}_i^{\mathrm{T}}\mathbf{w}\right) \tag{24}$$

where $\mathcal{N}(\cdot|0,1)$ and $\phi(\cdot)$ are the standard Gaussian probability and cumulative density functions. For logistic models, we have

$$a_i = \sigma(\mathbf{x}_i^{\mathrm{T}}\mathbf{w})\big(1 - \sigma(\mathbf{x}_i^{\mathrm{T}}\mathbf{w})\big) \tag{25}$$

where $\sigma(z) = \frac{1}{1+\exp(-z)}$.

At each iteration of HMH, we need to compute the Cholesky factor $\mathbf{R}_t$ of $\mathbf{H}_t$, which requires $O(d^3)$ computation. For high-dimensional data, this computation is expensive. Therefore, instead of computing the exact Cholesky factor $\mathbf{R}_t$, we approximate it using k sequential Cholesky downdates. The approximation details are given in the following paragraphs.

First, generate several points in the parameter space as the reference points. The reference points could be chosen as the data points near the posterior mode, or sampled from the exact or approximate target distribution. Compute the diagonal elements of the Hessian matrix. Also, calculate Cholesky factors for the reference points.

Now given a new data point at $\mathbf{w}$, compute the diagonal vector $\mathbf{a}$ of the Hessian matrix and find the nearest vector, denoted by $\mathbf{a}^\dagger$, to $\mathbf{a}$ among the diagonal vectors of the reference points. Denote by $\mathbf{R}_\dagger$ the Cholesky factor of the negative Hessian matrix $\mathbf{H}_\dagger$ that corresponds to $\mathbf{a}^\dagger$. Then we have

$$\mathbf{H}_\dagger = -\mathbf{x}\mathbf{A}_\dagger\mathbf{x}^\mathrm{T} \qquad \mathbf{H} = -\mathbf{x}\mathbf{A}\mathbf{x}^\mathrm{T} = \mathbf{H}_\dagger - \mathbf{x}\mathbf{A}_s\mathbf{x}^\mathrm{T} \tag{26}$$

where $\mathbf{A}_s = \mathrm{diag}\{\mathbf{a}_s\} = \mathrm{diag}\{\mathbf{a} - \mathbf{a}^\dagger\}$.

To reduce the computation, we prune $\mathbf{a}_s$ into a shorter vector $\mathbf{b}_s$ by removing all the elements in $\mathbf{a}_s$ whose absolute values are smaller than a predefined threshold δ, such that the length k of $\mathbf{b}_s$ is much smaller than the length d of $\mathbf{a}_s$. Thus, we can approximate $\mathbf{H}$ by a low-rank update

$$\mathbf{H} \approx \mathbf{H}_\dagger - \tilde{\mathbf{x}}\mathbf{B}_s\tilde{\mathbf{x}}^\mathrm{T}$$

where $\mathbf{B}_s = \mathrm{diag}\{\mathbf{b}_s\}$, and $\tilde{\mathbf{x}}$ consists of the columns in $\mathbf{x}$ corresponding to the elements left in $\mathbf{b}_s$. Then, we can apply k sequential Cholesky rank-1 donwdates to the reference Cholesky factor $\mathbf{R}_\dagger$, in order to get the new factor $\mathbf{R}$. The k sequential Cholesky rank-1 donwdates require $O(d^2k)$ time, instead of $O(d^3)$ as the direct computation of $\mathbf{R}$. For high-dimensional data, this approximation could save a lot of time at each sampling iteration.

4 Adaptive Multiple-Importance-Try Sampling

In this section, we incorporate adaptive jumping distributions in HMH with the multiple-try Metropolis (MTM) algorithm [5]. The new algorithm, resulted from this combination, is called the Adaptive Multiple-Importance-Try (AMIT) algorithm. The basic idea of the AMIT algorithm is to draw multiple samples from a Hessian-based adaptive jumping distribution at each iteration, stochastically pick one by preferring those samples with large importance ratios, and weight all the rest samples.

Similar to MTMs, the AMIT algorithm explores more thoroughly in the neighboring region defined by its adaptive jumping distribution than HMH. This is useful when the data dimension is high and therefore the parameters of an adaptive jumping distribution is relatively expensive to obtain. In contrast to

MTMs, the AMIT algorithm can efficiently employ the information from all the samples instead of from single one selected from multiple trials at each iteration as MTMs. This is valuable when the number of trials at each iteration is large.

In sum, the AMIT algorithm estimates $\int f(\mathbf{w})p(\mathbf{w}|D)d\mathbf{w}$ as follows:

1. Draw a starting point $\mathbf{w}^1$, for which $p(\mathbf{w}^0|y) > 0$, from a starting distribution $p_1(\mathbf{w})$. Compute the mean $\mathbf{m}^1$ and the Cholesky factor $\mathbf{R}_1$ of $-\mathbf{H}_1$, for which $\mathbf{R}_1^{\mathrm{T}}\mathbf{R}_1 = -\mathbf{H}_1$, for the first forward jumping distribution.

2. For $t = 1, 2, \ldots$:

   (a) Sample h independent trials, $\mathbf{w}_1^\star, \ldots, \mathbf{w}_h^\star$ from the forward jumping distribution $J_t(\mathbf{w}_j^\star|\mathbf{w}^t) = \mathcal{N}(\mathbf{w}_j^\star|\mathbf{w}^t, \Sigma_t)$ for $j = 1, \ldots, h$ as in equations (20) and (21). It costs $O(d^2 h)$ computation. For each $\mathbf{w}_j^\star$, Compute the importance ratio

   $$\alpha(\mathbf{w}_j^\star, \mathbf{w}^t) = \frac{p(\mathbf{w}_j^\star|D)}{J_t(\mathbf{w}_j^\star|\mathbf{w}^t)}. \tag{27}$$

   If the posterior $p(\mathbf{w}_j^\star|D)$ is hard to obtain, we compute the following weighted likelihood

   $$\alpha(\mathbf{w}_j^\star, \mathbf{w}^t) = \frac{p(\mathbf{y}|\mathbf{w}_j^\star, \mathbf{x})p(\mathbf{w}_j^\star)}{J_t(\mathbf{w}_j^\star|\mathbf{w}^t)} \tag{28}$$

   where $p(\mathbf{w}_j^\star)$ is the prior. Here we actually choose a special form of the λ parameter in MTM [5].

   (b) Sample $\mathbf{w}^\star$ from the trial set $\{\mathbf{w}_1^\star, \ldots, \mathbf{w}_h^\star\}$ with probability proportional to its importance ratio or its weighted likelihood $\alpha_j, j = 1, \ldots, h$.

   First, assign weight 1 to $\mathbf{w}^\star$. Then denote by W_t the trial set without $\mathbf{w}^\star$. Assign $\alpha(\mathbf{w}_j^\star, \mathbf{w}^t)$ to the elements in W_t if we choose $\mathbf{w}^\star$ based on its importance ratio (27); otherwise, assign the normalized weighted likelihood to the corresponding elements.

   (c) Compute the mean and the Cholesky factor $\mathbf{R}_\star$ of $-\mathbf{H}_\star$ for the backward jumping distribution $\mathcal{N}(\mathbf{m}^\star, \Sigma_\star)$ where $\Sigma_\star = -\mathbf{H}_\star$. We can compute the exact $\mathbf{R}_\star$, or, as in HMH, use a few sequential Cholesky donwdates to approximate $\mathbf{R}_\star$ for high-dimensional data.

   (d) Sample $\mathbf{q}_1, \ldots, \mathbf{q}_{h-1}$ from the backward jumping distribution $N(\mathbf{w}^\star, \Sigma_\star)$, and let $\mathbf{q}_h = \mathbf{w}^t$. Compute $\alpha(\mathbf{q}_j, \mathbf{w}^\star)$ for $\mathbf{q}_j, j = 1, \ldots, h$ as in equation (27) or (28).

   (e) Compute the ratio of the sum of multiple importance ratios or weighted likelihoods:

   $$r_g = \frac{\sum_j \alpha(\mathbf{w}_j^\star, \mathbf{w}^t)}{\sum_j \alpha(\mathbf{q}_j, \mathbf{w}^\star)}. \tag{29}$$

(f) With probability $\min(r_g, 1)$, we set $\mathbf{w}^{t+1} = \mathbf{w}^\star$, $\mathbf{m}^{t+1} = \mathbf{m}^\star$, and $\mathbf{R}_{t+1} = \mathbf{R}_\star$; Otherwise, we set $\mathbf{w}^{t+1} = \mathbf{w}^t$, $\mathbf{m}^{t+1} = \mathbf{m}^t$, and $\mathbf{R}_{t+1} = \mathbf{R}_t$.

3. Finally, we estimate the integral $\int f(\mathbf{w})p(\mathbf{w}|D)d\mathbf{w}$ as the weighted mean of $f(\mathbf{w}^t)$s and $f(\mathbf{w}_j^\star)$s.

   Note that $\mathbf{w}^t$s are in the Markov chain and sampled from the target distribution after convergence since the detailed balance is satisfied. Its proof is same as in Liu et al's MTM paper [5]. On the other hand, $\mathbf{w}_j^\star$s are off the chain and not sampled from the target distribution. Thus, we need to weight $\mathbf{w}_j^\star$s by the importance ratios or the weighted likelihoods as in step 2.(b).

   If we use the importance ratios to weight $f(\mathbf{w}_j^\star)$s, the estimate of $\int f(\mathbf{w})p(\mathbf{w}|D)d\mathbf{w}$ is clearly unbiased. On the other hand, if we use the weighted likelihoods to weight $f(\mathbf{w}_j^\star)$s, the estimate is biased. In the latter case, in order to have an unbiased estimate of the integral, we can throw away $\mathbf{w}_j^\star$s and only keep $\mathbf{w}^t$s. Using $\mathbf{w}_j^\star$s or not is a trade-off between variance and bias.

5 Experimental Performance

In this section, we first empirically compare Gibbs, data augemntation (DA), normal multiple-try metropolis (MTM), HMH, and AMIT samplers on a probit regression problem. Then we illustrate by an example the important role of the learning rate in HMH and the risk of directly adopting Newton's method for sampling without using any learning rate.

First, we use a Kidney biopsy data set with the presence of Lupus as the response and the difference between IgG3 and IgG4 and Clq as two predictors. In our probit regression model, we use an intercept with both covariates. The comparison results are summarized in figures 1 and 2. Figure 1 shows the results of Gibbs, DA, and HMH samplers, and figure 2 shows the results of MTM and AMIT samplers. The MTM sampler has a fixed Gaussian jumping distribution $\mathcal{N}(\mathbf{w}, 10\mathbf{I})$.

As shown in these figures, the HMH sampler results in smaller autocorelation coefficients, and its $\sqrt{\hat{R}}$ statistic, a convergence criterion, deceases faster than the Gibbs, DA, and MTM samplers. By combining Hessian-based adaptive transition distributions with the local exploration capability of the multiple-try approach, the AMIT sampler outperforms all the other samplers as demonstrated in these figures.

In addition, we compare two HMH samplers to illustrate the effect of using the learning rate in equation (18). One sampler uses a learning rate, randomly drawn from a uniform distribution between 0 and 0.7, while the other one does not use any learning rate, i.e., γ equals 1 all the time. Since a poor Gaussian approximation of the target distribution often happens in some tail regions of the target distribution, we deliberately choose three random points far from the

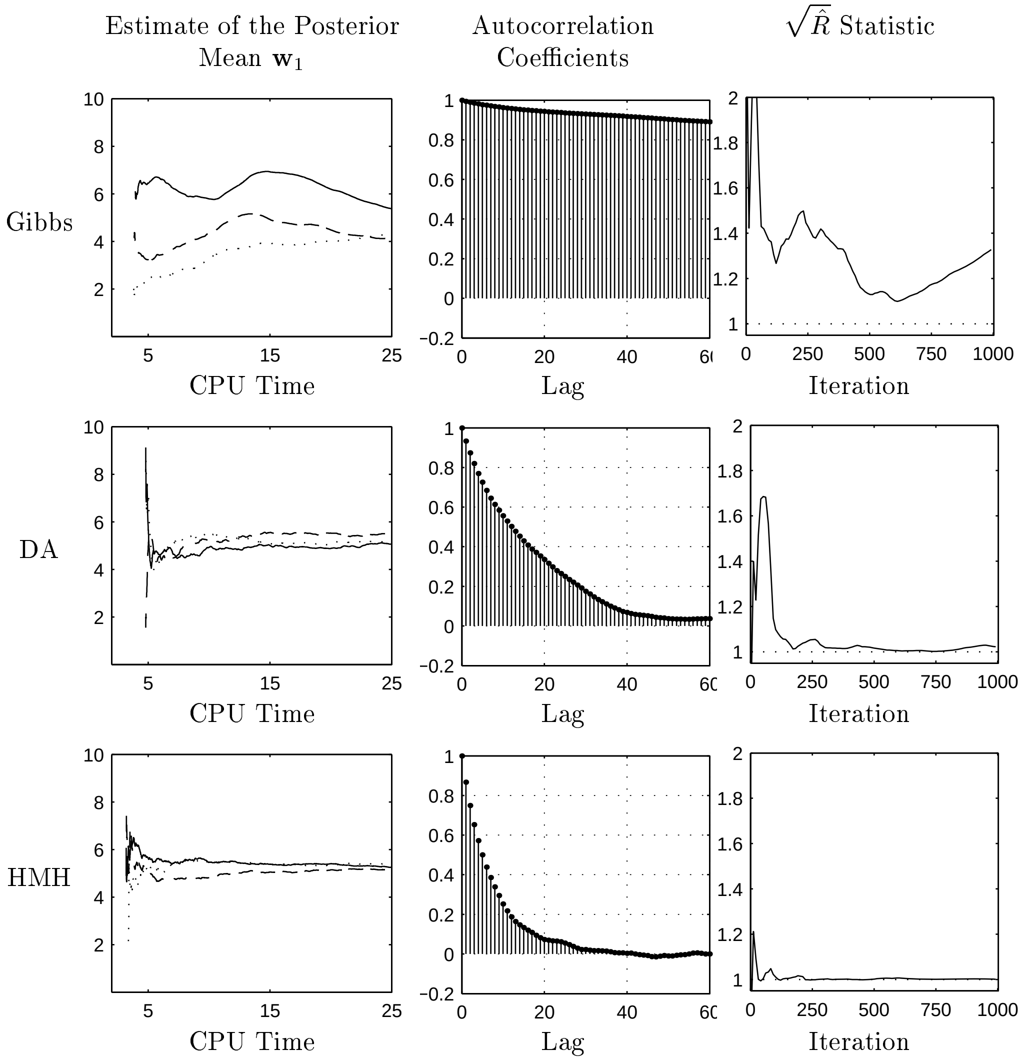

Figure 1: Comparison of Gibbs, DA, HMH, MTM, and AMIT Samplers for Fitting a Probit Regressoin Model with Two Covariates as well as an Intercept: Part 1. The rows of this figure correspond to the Gibbs, DA, and HMH samplers respectively. The columns are with all summaries computed for the first covariate $\mathbf{w}_1$. Clearly, the HMH sampler results in smaller autocorelation coefficients and reduces the $\sqrt{\hat{R}}$ statistic faster than the Gibbs and DA samplers. Note that the X axis in the first column is indexed by the CPU time instead of by the number of iterations.

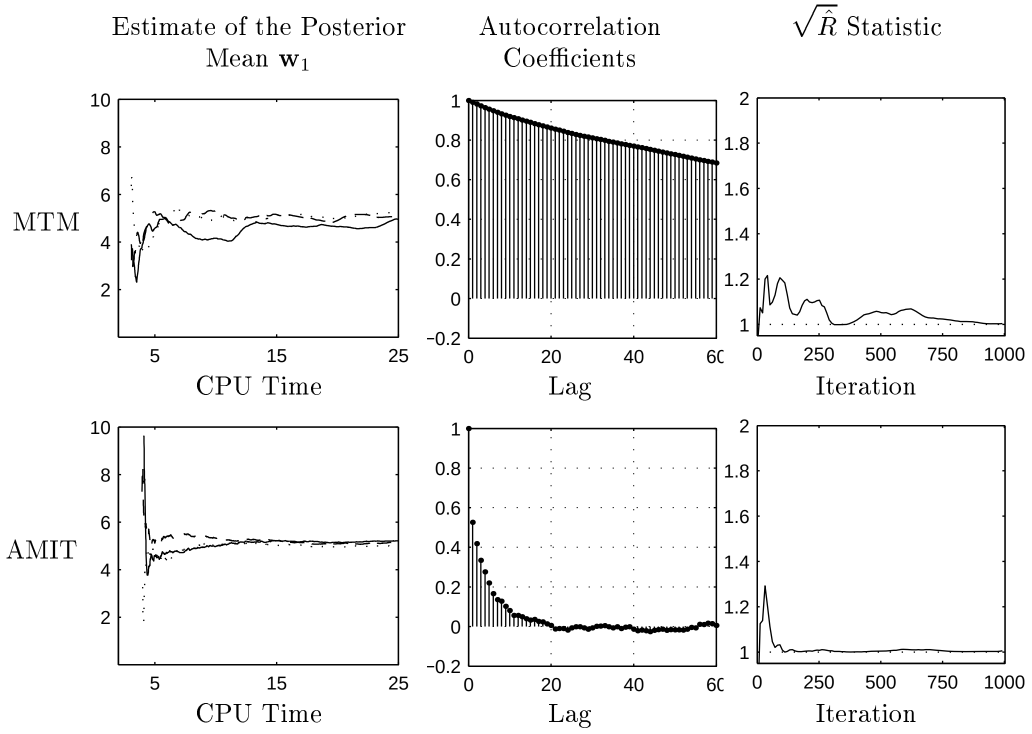

Figure 2: Comparison of Gibbs, DA, HMH, MTM, and AMIT Samplers for Fitting a Probit Regressoin Model with Two Covariates as well as an Intercept: Part 2. The rows of this figure correspond to MTM, and AMIT samplers respectively. The MTM sampler is a normal Multiple-try metropolis sampler with a fixed Gaussian jumping distribution, $\mathcal{N}(\mathbf{w}, 10\mathbf{I})$. As in figure 1, the columns of figure are with all summaries computed for the first covariate $\mathbf{w}_1$. By combining Hessian-based adaptive transition distributions with the multiple trial approach, the AMIT sampler outperforms Gibbs, DA, HMH, and MTM samplers. The AMIT sampler computes the exaxt Hessians and select samples from multiple trials. It results in the smallest autocorelation coefficients in column 2 compared to the other methods.

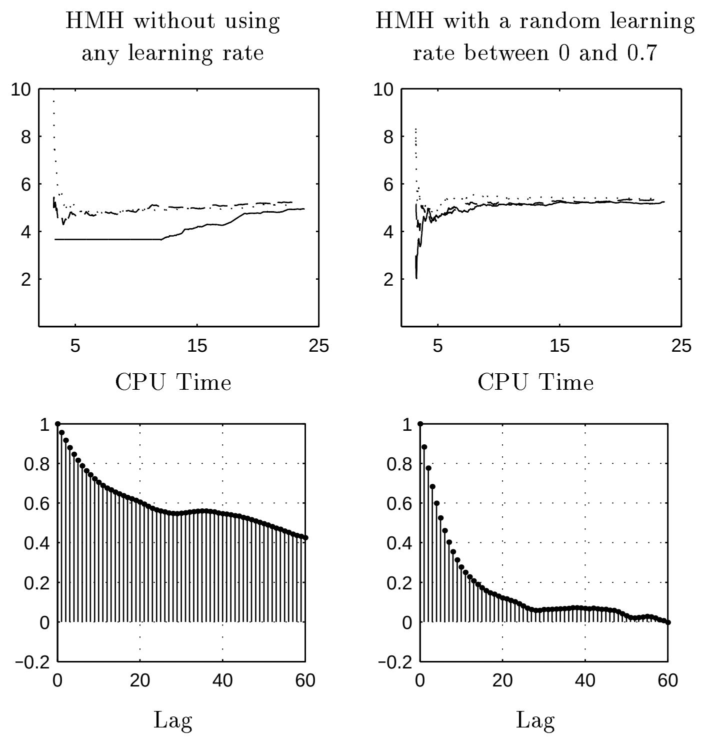

Figure 3: The Role of Learning Rate in HMH. The HMH chains in the left and right plots start from the same three points, which are far from the mean and mode of the target distribution. Without using any learning rate, one of the chains in the left plot is stuck in the local region for a long time due to a poor Gaussian approximation of the target distribution. This leads to large autocorrelation coefficients of that chain, and therefore slows down the convergence speed. By contrast, the HMH chains in the right plot do not suffer from this problem. In addition, we observe in our experiments that a fixed small number, which is less than 1, can be well served as the learning learning rate in HMH too.

mode as the initial sampling points to test these two HMHs. From figure 3, we can see that the use of the learning rate in HMH removes the risk of being stuck at a local region due to a unreasonable jumping distribution. Using a learning rate, the HMH sampler works pretty robust, so that they have never been observed to be stuck in any local region in our experiments.

6 Conclusion and Future Work

In this paper, we have proposed two new MCMC samplers, HMH and AMIT samplers. The AMIT sampler enjoys both HMH and MTM's advantages and greatly outperforms the other samplers on a probit regression problem.

As to the future work, we plan to apply the new sampling methods to different data models and have more experimental evaluations.

References

[1] J. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88:669–679, 1993.

[2] C. Bishop. *Neural networks for pattern recognition.* Oxford university press, 1995.

[3] J. Geweke and H. Tanizaki. Note on the sampling distribution for the metroplis-hastings algorithm. http://ht.econ.kobe-u.ac.jp/~tanizaki/cv/working/mh.pdf, 2002.

[4] C. Liu, D. B. Rubin, and Y. N. Wu. parameter expansion to accelerate em: the px-em algorithm. *Biometrika*, 85:755–770, 1998.

[5] J. S. Liu, F. Liang, and W. H. Wong. The use of multiple-try method and local optimization in metropolis sampling. *Journal of American Statistical Association.*, 95:121–134, 2000.

[6] X. L. Meng and D. A. van Dyk. The em algorithm – an old folk song sung to a fast new tune. In *Journal of the Royal Statistical Society*, B., pages 511–567. 1997.

[7] D. A. van Dyk and X. L.Meng. The art of data augmentation (with discussion). *Journal of Computational and Graphical Statistics*, 10(1):1–111, March 2001.