Hessian-based Markov Chain Monte-Carlo Algorithms

Yuan Qi
Media Lab, MIT
Cambridge, MA, 02139
yuanqi@media.mit.edu

Thomas P. Minka
Statistics Department of CMU
Pittsburgh, PA, 15123
minka@stat.cmu.edu

Introduction

- Gibbs and Marginal Augmentation samplers for Probit Model
- Hessian-based Metropolis-Hastings (HMH) Algorithm
- Adaptive Multiple Importance-try (AMIT) Algorithm
- Experimental Results
- Conclusion and Future work

Generalized Linear Model

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 $\mathbf{w}$:

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

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:

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

Gibbs Sampler

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

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

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

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:

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

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

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

where $\mathbf{z} = [z_1, \ldots, z_n]^\top$ 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 .

From Fast EM to Efficient Sampler - Optimal Marginal Data-Augmentation Sampler

van Dyk and Meng adopt the EM criterion and design optimal marginal data augmentation (DA) sampler for probit regression:

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

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

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.

- The DA sampler converges much faster than the plain Gibbs sampler for probit regression.
- It is not easy to design an optimal marginal augmentation scheme for new problems.

Hessian-based Metropolis-Hastings algorithm (1)

Main Features:

- By locally approximating the target distribution by a Gaussian, HMH obtains an adaptive proposal distribution, 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.

Main idea:

- Jumping Distribution $J_t(\mathbf{w}^\star|\mathbf{w})$: a Gaussian approximation of $p(\mathbf{w}|\mathbf{y})$ as in the Newton optimization method:

$$\mathbf{g}^t = \frac{\mathrm{d}\log p(\mathbf{w}|\mathbf{y})}{\mathrm{d}\mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}^t} \tag{12}$$

$$\mathbf{H} = \frac{\mathrm{d}^2\log p(\mathbf{w}|\mathbf{y})}{\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}^\mathsf{T}} \tag{13}$$

$$\Sigma_t = -\mathbf{H}^{-1}|_{\mathbf{w}=\mathbf{w}^t} \tag{14}$$

$$\mathbf{m}^t = \mathbf{w}^t + \gamma\,\Sigma_t\,\mathbf{g}^t \tag{15}$$

$$J_t(\mathbf{w}^\star|\mathbf{w}^t) = \mathcal{N}(\mathbf{w}^\star \mid \mathbf{m}^t, \Sigma_t) \tag{16}$$

HMH (2)

- By using the Newton direction to update the mean of the jumping distribution, a new sample of HMH is more likely drawn from a high probable region.

- Note that the learning rate γ plays an important role in the performance of HMH.

  Using a small random learning rate helps Markov sampling chains leave a local region where the target distribution is far from a Gaussian. If the target distribution can not be well approximated by a Gaussian, a fixed big change of the proposed mean from $\mathbf{w}^t$ to $\mathbf{m}^\star$ will result in a low acceptance ratio and the Markov chain may be stuck in a local region for a long time.

- Another method to address the same problem is to use the model trust region method, proposed in the optimization community.

HMH for Generalized Linear Models

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

$$\mathbf{H} = -\mathbf{x}\mathbf{A}\mathbf{x}^\top = -\mathbf{x}\text{diag}\{\mathbf{a}\}\mathbf{x}^\top \tag{17}$$

where $\mathbf{a} = [a_1, \ldots, a_n]$.

- For probit models, we have

  $$a_i = \frac{\mathcal{N}(y_i \mathbf{x}_i^\top \mathbf{w}|0,1)}{\phi(y_i \mathbf{x}_i^\top \mathbf{w})} \left(\frac{\mathcal{N}(y_i \mathbf{x}_i^\top \mathbf{w}|0,1)}{\phi(y_i \mathbf{x}_i^\top \mathbf{w})} + y_i \mathbf{x}_i^\top \mathbf{w} \right) \tag{18}$$

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

- For logistic models, we have

  $$a_i = \sigma(\mathbf{x}_i^\top \mathbf{w})\big(1 - \sigma(\mathbf{x}_i^\top \mathbf{w})\big) \tag{19}$$

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

Cholesky Factor Approximation:

- At each iteration of HMH, the computation time: $\mathbf{R}_t$: $O(d^3)$.

- Approximate approximate $\mathbf{R}_t$ by k sequential Cholesky downdates based on pre-computed $\mathbf{R}_l$ where $l = 1, \ldots, k$ or previous $\mathbf{R}_t - 1$. The reduced computation time: $O(d^2k)$

Adaptive Multiple-Importance-Try Sampling

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. The AMIT algorithm explores more thoroughly in the neighboring region defined by its adaptive jumping distribution than HMH.

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 $\mathbf{R}_1^\top \mathbf{R}_1 = -\mathbf{H}_1$, 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 $j = 1, \ldots, h$. It costs $O(d^2h)$ computation. For each $\mathbf{w}_j^\star$, Compute the importance ratio

$$\alpha(\mathbf{w}_j^\star, \mathbf{w}^t) = \frac{p(\mathbf{w}_j^\star|D)}{J_t(\mathbf{w}_j^\star|\mathbf{w}^t)}. \tag{20}$$

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

$$\alpha(\mathbf{w}_j^\star, \mathbf{w}^t) = \frac{p(\mathbf{y}|\mathbf{w}_j^\star, \mathbf{x})p(\mathbf{w}_j^\star)}{J_t(\mathbf{w}_j^\star|\mathbf{w}^t)} \tag{21}$$

where $p(\mathbf{w}_j^\star)$ is the prior.

(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 (20); otherwise, assign the normalized weighted likelihood to the corresponding elements.

(c) Compute $\mathbf{m}^\star$ and $\mathbf{R}_\star$ of $-\mathbf{H}_\star$ for the backward jumping distribution $\mathcal{N}(\mathbf{m}^\star, \mathbf{\Sigma}_\star)$ where $\mathbf{\Sigma}_\star = -\mathbf{H}_\star$. We can compute the exact $\mathbf{R}_\star$, or use a few Cholesky donwdates to approximate $\mathbf{R}_\star$.

(d) Sample $\mathbf{q}_1, \ldots, \mathbf{q}_{h-1}$ from $N(\mathbf{w}^\star, \mathbf{\Sigma}_\star)$, and let $\mathbf{q}_h = \mathbf{w}^t$. Compute $\alpha(\mathbf{q}_j, \mathbf{w}^\star)$ for $\mathbf{q}_j, j = 1, \ldots, h$.

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

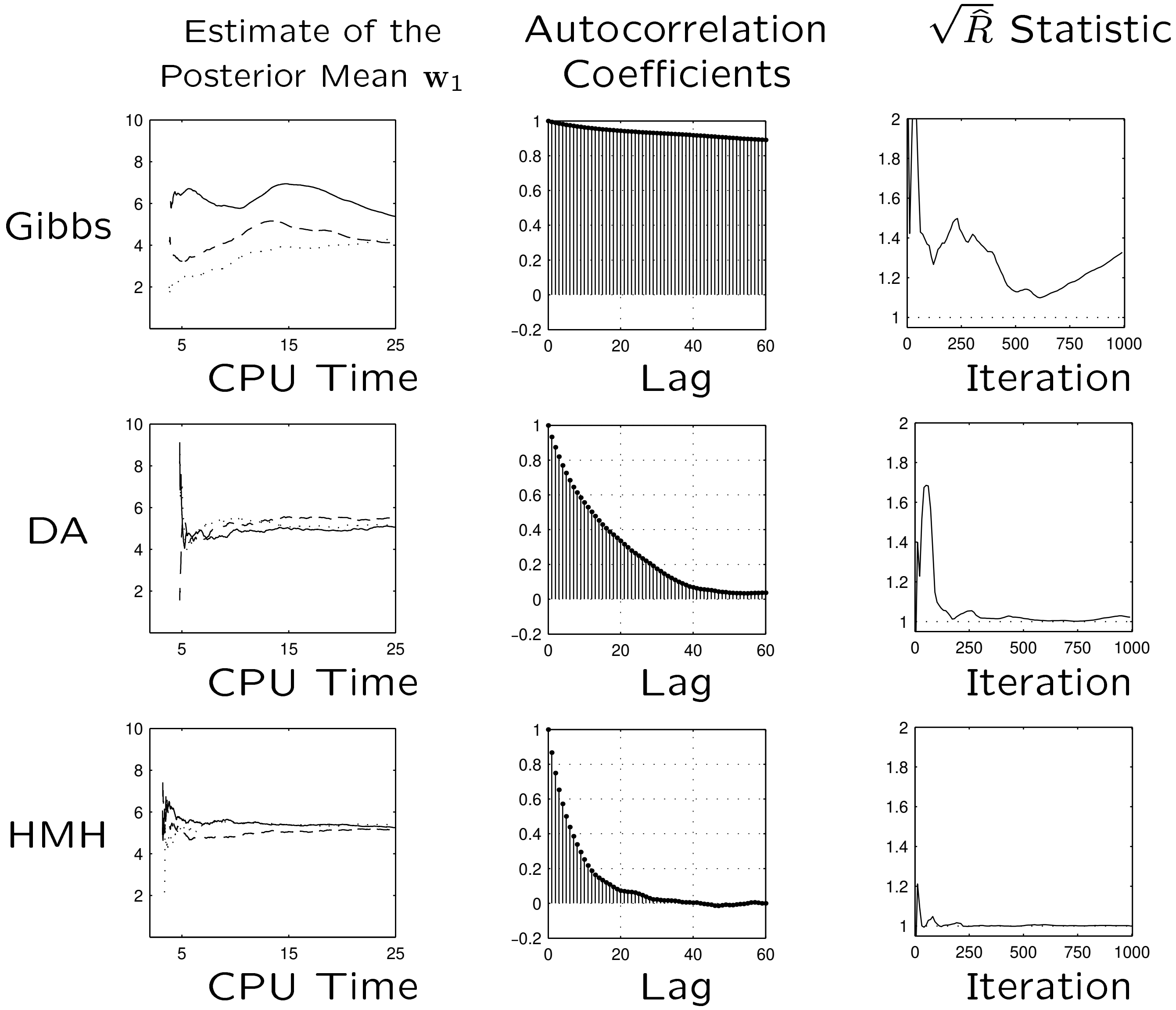
$$r_g = \frac{\sum_j \alpha(\mathbf{w}_j^\star, \mathbf{w}^t)}{\sum_j \alpha(\mathbf{q}_j, \mathbf{w}^\star)}. \tag{22}$$

(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 $\mathbf{R}_{t+1} = \mathbf{R}_t$.

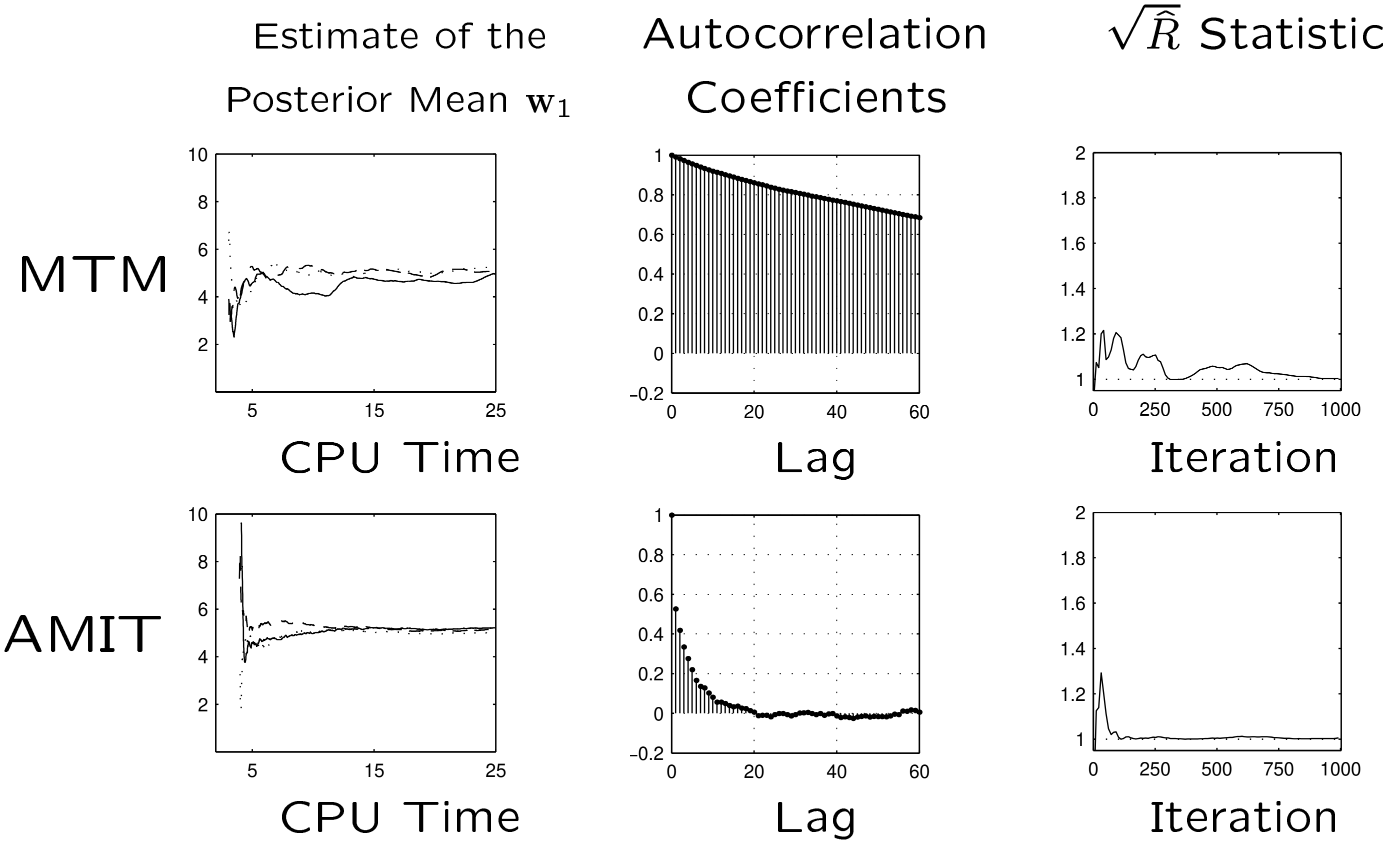
3. Finally, we estimate the integral $\int f(\mathbf{w})p(\mathbf{w}|D)d\mathbf{w}$ as the weighted mean of $f(\mathbf{w}^t)$s and $f(\mathbf{w}_j^\star)$s.

   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. 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 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.



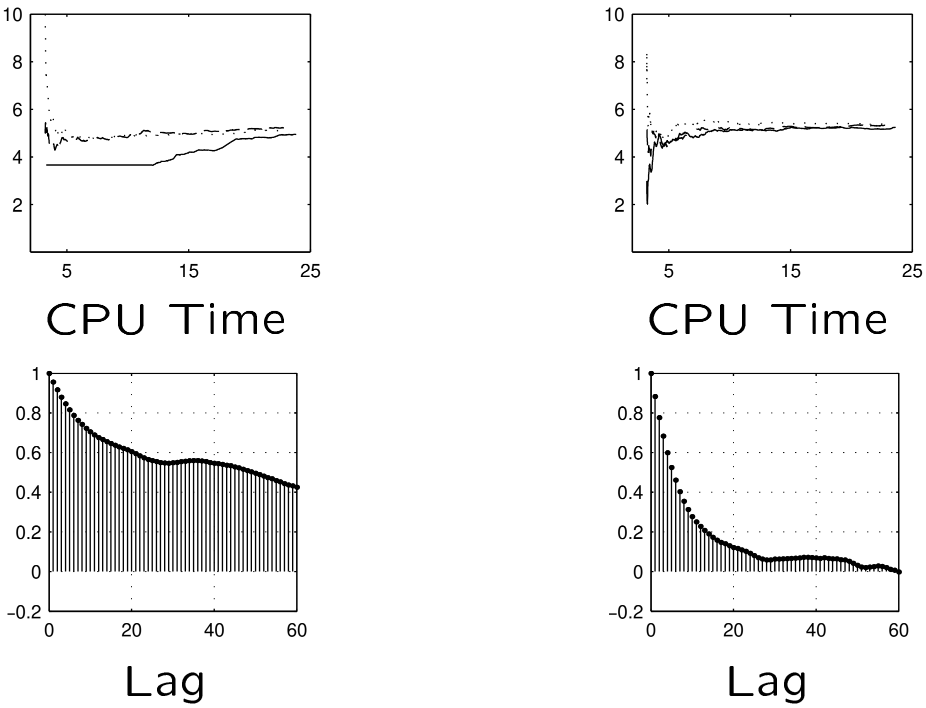
Comparison of Gibbs, DA, HMH, MTM, and AMIT Samplers for Fitting a Probit Regression Model with Two Covariates as well as an Intercept: Part 1. They are tested on a Kidney biopsy data set. The columns are with all summaries computed for the first covariate $\mathbf{w}_1$. Clearly, the HMH sampler results in smaller autocorrelation coefficients and reduces the $\sqrt{\widehat{R}}$ statistic faster than the Gibbs and DA samplers.



Comparison of Gibbs, DA, HMH, MTM, and AMIT Samplers for Fitting a Probit Regression Model with Two Covariates as well as an Intercept: Part 2. The MTM sampler uses a fixed Gaussian jumping distribution, $\mathcal{N}(\mathbf{w}, 10\mathbf{I})$. The AMIT sampler outperforms Gibbs, DA, HMH, and MTM samplers. It results in the smallest autocorrelation coefficients in column 2 compared to the other methods.

The Role of Learning Rate in HMH

HMH without using any learning rate | HMH with a random learning rate between 0 and 0.7



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.

Conclusion and Future Work

- Two new efficient Monte Carlo methods, the HMH and AMIT samplers.
- More experimental evaluations.
- Applying to different data models.

Reference:

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.