BILINGUAL PUBLISHING CO.
Pioneer of Global Academics Since 1984

Review of Educational Theory
https://ojs.bilpublishing.com/index.php/ret

ARTICLE

Metropolis-Hastings Algorithm with Delayed Acceptance and Rejection

Yulin Hu Yayong Tang*

College of Mathematics, Sichuan University, Chengdu, Sichuan, 610064, China

ARTICLE INFO

Article history
Received: 25 March 2019
Revised: 1 April 2019
Accepted: 23 April 2019
Published Online: 30 April 2019

Keywords:
Metropolis-Hastings algorithm
Delayed acceptance
Delayed rejection

ABSTRACT

Metropolis-Hastings algorithms are slowed down by the computation of complex target distributions. To solve this problem, one can use the delayed acceptance Metropolis-Hastings algorithm (MHDA) of Christen and Fox (2005). However, the acceptance rate of a proposed value will always be less than in the standard Metropolis-Hastings. We can fix this problem by using the Metropolis-Hastings algorithm with delayed rejection (MHDR) proposed by Tierney and Mira (1999). In this paper, we combine the ideas of MHDA and MHDR to propose a new MH algorithm, named the Metropolis-Hastings algorithm with delayed acceptance and rejection (MHDAR). The new algorithm reduces the computational cost by division of the prior or likelihood functions and increase the acceptance probability by delay rejection of the second stage. We illustrate those accelerating features by a realistic example.

1. Introduction

MH algorithm (Hastings, 1970; Metropolis, et al, 1953) has solved integral calculation in the complex posterior.[1,2] However, one of the most important and challenging issues is to compute the rate of acceptance. The acceptance probability of the standard MH algorithm is expressed as

$$a\left(x,y\right)=\min\left\{1,\frac{\pi(y)q(x\mid y)}{\pi(x)q(y\mid x)}\right\} \tag{1.1}$$

Where represents the target density function (that is, the posterior distribution), represents the proposal distribution. In the acceptance and rejection of MH algorithm, we always need to calculate and thus need to calculate rations of . But it is a complex task with large amount of computation, e.g., the posterior distribution involves another integral.

To reduce the computational cost, the delayed acceptance MH algorithm of Christen and Fox (2005) is a two-stage Metropolis-Hastings algorithm in which, typically, proposed parameter values are accepted or rejected at the first stage based on a computationally cheap surrogate for the likelihood.[3] Detailed balance with respect to the true posterior is ensured by a second accept-reject step, based on the computationally expensive likelihood, for those parameter values which are accepted in the first stage. Delayed acceptance algorithms thus provide draws from the posterior distribution of interest whilst potentially limiting the number of evaluations of the expensive likelihood. Although the amount of computation is reduced, the acceptance rate is also reduced compared to the standard MH algorithm. For the purpose of improving the acceptance of MH algorithm, we can use Metropolis-Hastings algorithm

**Corresponding Author:*
Yayong Tang,
College of Mathematics, Sichuan University, No. 24, South Section, First Ring Road, Chengdu, Sichuan, 610064, China;
E-mail: yayongtang@scu.edu.cn.

DOI: https://doi.org/10.30564/ret.v2i2.682

with delayed rejection (MHDA) as defined by Tierney and Mira (1999).[4] In this paper we combine the ideas of MHDA with MHDR and propose a new MH method is the Metropolis-Hastings algorithm with delayed acceptance and rejection (MHDAR).

The new algorithm, involving two stages, reduces the computational cost by division the prior or likelihood function (that is, the first stage) and increase acceptance ratio by the second stage. We illustrate those accelerating features by a realistic example of two-dimensional distribution.

2. Metropolis-Hastings Algorithm with Delayed Acceptance and Rejection

In this section, we recall the Metropolis-Hastings algorithm with delayed rejection and the Metropolis-Hastings algorithm with delayed acceptance, which are useful for our later discussions. And then we propose our new algorithm, the Metropolis-Hastings algorithm with delayed acceptance and rejection.

2.1 Metropolis-Hastings Algorithm with Delayed Rejection

The Metropolis-Hastings algorithm with delayed rejection (MHDR), proposed by Tierney and Mira (1999), achieve this goal: when making a rejection decision, we use a different proposal distribution to generate a second candidate state instead of obtaining a duplicate sample and accepting or rejecting it based on the probability of an appropriate calculation. [4] Therefore, the update process of the MHDR algorithm is as follows:

Algorithm 2.1 Metropolis-Hastings Algorithm with Delayed Rejection

(1) Generate $y \sim q_1(\cdot|x_0)$

(2) Compute

$$a_1(x_0, y) = \min\left\{1, \frac{\pi(y)q_1(x_0|y)}{\pi(x_0)q_1(y|x_0)}\right\}$$

(3)

$$x_1 = \begin{cases} y, & with\ probability \quad a_1(x_0,y); \\ x_0, & with\ probability \quad 1-a_1(x_0,y). \end{cases}$$

If $x_1 \neq y$ go to step 4, otherwise stop and output $x_1=y$

(4) Generate $y_1 \sim q_1(\cdot|x_0,y)$

(5) Compute

$$a_2(x_0,y,y_1) = \min\left\{1, \frac{\pi(y_1)q_1(y|y_1)\mathrm{q}_2(\mathrm{x}_0|\mathrm{y},\mathrm{y}_1)(1\text{-}\mathrm{a}_1(\mathrm{y}_1,\mathrm{y}))}{\pi(x_0)q_1(y|x_0)q_2(y_1|x_0,y)(1-a_1(x_0,y))}\right\}$$

(6)

$$x_1 = \begin{cases} y, & with\ probability \quad a_2(x_0,y,y_1); \\ x_0, & with\ probability \quad 1-a_2(x_0,y,y_1). \end{cases}$$

2.2 Metropolis-Hastings Algorithm with Delayed Acceptance

The MH algorithm with delayed acceptance is aimed at reduce the amount of computation. The delayed acceptance MCMC algorithm of Christen and Fox (2005) is a two-stage Metropolis-Hastings algorithm in which, typically, proposed parameter values are accepted or rejected at the first stage based on a computationally cheap surrogate for the likelihood.[3]

In the section, we reduce the computational cost by division the prior or likelihood function in the first stage and increase acceptance ratio by the second stage. It is similar to Banterle et al. (2015). The update process of the MHDA algorithm is as follows:

Algorithm 2.2 Metropolis-Hastings Algorithm with Delayed Acceptance

(1) Generate $y \sim q_1(\cdot|x_0)$

(2) Compute

$$a_1(x_0, y) = \min\left\{1, \frac{\pi_1(y)q(x_0 \mid y)}{\pi_1(x_0)q(y \mid x_0)}\right\}$$

(3) Take

$$x_1 = \begin{cases} y, & with\ probability \quad a_1(x_0,y); \\ x_0, & with\ probability \quad 1-a_1(x_0,y). \end{cases}$$

If $x_1=y$ go to step 4, otherwise stop and output $x_1=x_0$

(4) Compute

$$a_2(x_0, y) = \min\left\{1, \frac{\pi(y_1)\pi_1(x_0)}{\pi(x_0)\pi_1(y)}\right\}$$

(5) Take

$$x_1 = \begin{cases} y, & with\ probability \quad a_2(x_0,y); \\ x_0, & with\ probability \quad 1-a_2(x_0,y,y_1). \end{cases}$$

So acceptance probability of MHDA is $a(x_0,y)=a_1(x_0,y)\,a_2(x_0,y)$.

2.3 Metropolis-Hastings Algorithm with Delayed Acceptance and Rejection

The Metropolis-Hastings algorithm with delayed acceptance and rejection (MHDAR) combines the ideas of MHDA and MHDR. It reduces the computing costs by a decomposition of the target function, increases rate of acceptance by the second stage of delayed rejection. The

DOI: https://doi.org/10.30564/ret.v2i2.682

update process of the MHDAR algorithm is as follows:

Algorithm 2.3 Metropolis-Hastings Algorithm with Delayed Acceptance and Rejection

(1) Generate $y \sim q_1(\cdot|x_0)$

(2) Compute

$$a_1\left(x_0,y\right)=\min\left\{1,\frac{\pi_1(y)q(x_0\mid y)}{\pi_1(x_0)q(y\mid x_0)}\right\}$$

(3) Take

$$x_1=\begin{cases} y, & with\ \ probability\ \ \ a_1(x_0,y); \\ x_0, & with\ \ probability\ \ \ 1-a_1(x_0,y). \end{cases}$$

If $x_1=y$ go to step 4, otherwise stop and output $x_1=x_0$

(4) Compute

$$a_2\left(x_0,y\right)=\min\left\{1,\frac{\pi(y_1)\pi_1(x_0)}{\pi(x_0)\pi_1(y)}\right\}$$

(5) Take

$$x_1=\begin{cases} y, & with\ \ probability\ \ \ a_2(x_0,y); \\ x_0, & with\ \ probability\ \ \ 1-a_2(x_0,y,y_1). \end{cases}$$

If $x_1 \neq y$ go to step 6, otherwise stop and out put $x_1=y$

(6) Generate $y_1 \sim q_1(\cdot|x_0,\ \ y)$

(7) Compute

$$a_3(x_0,y,y_1)=\min\left\{1,\frac{\pi(y)q_2(x_0\mid y_1,y)q_1(y\mid y_1)(1\text{-}a_2(y_1,y))a_1(y_1,y)}{\pi(x_0)q_2(y_1\mid x_0,y)q_1(y\mid x_0)(1-a_2(x_0,y))a_1(x_0,y)}\right\}$$

(8) Take

$$x_1=\begin{cases} y, & with\ \ probability\ \ \ a_3(x_0,y,y_1); \\ x_0, & with\ \ probability\ \ \ 1-a_3(x_0,y,y_1). \end{cases}$$

We assume that the target distribution π and the proposal density $q(\cdot|x)$ all admit densities with respect to the Lebesgue or counting measures. We also denote the target density by π.[5]

Let $(x_n)_{n\geq 1}$ be a Markov chain evolving on X with MHDAR algorithm Markov transition kernel P associated wit q hand π .i.e. for $A \in B(X)$, where $B(X)$ is the Boreal σ-field, on X.[6]

$$P(x_0,A)=P_1\ (x_0,A)+\int_X(1-a_2\ (x_0,x_1\ )\ )a_1\ (x_0,x_1\ )\ P_2\ (x_0,y,A)$$

Theorem 2.1 The transition kernel of the MHDAR update satisfies the reversibility with respect to target distribution:[7]

$$\pi(x_0\ )P(x_0,dx_1\ )=\pi(x_1\ )P(x_1,dx_0\ ) \qquad (2.1)$$

To prove Theorem 2.1, we have the following two lemmas.

Lemma 2.1 If $x_0,\ \ x_1 \in X$ then

$$\pi(x_0)\ P_1\ (x_0,dx_1)=\pi(x_1)\ P_1\ (x_1,dx_0) \qquad (2.2)$$

Proof: The form of Markov chain with transition kernel P_1

$$P_1\ (x_0,dx_1)=\int_A q(x_0,x_1)\ a_1\ (x_0,x_1),dx_1\ \ +(\text{-}\int_X q(x_0,x_1)a_1\ (x_0,x_1),dx_1)I_A(x).$$

From stage one of the MHDAR algorithm we know that the detailed balance condition

$$\pi(x_0)\ q_1\ (x_0,x_1)\ a_1\ (x_0,x_1)=\pi(x_1)\ q_1\ (x_1,x_0)\ a_1\ (x_1,x_0)$$

(2.3)

(1) It is obviously (2.2) established when $x_0=x_1$

(2) When $x_0 \neq x_1$

$$a_1\left(x_0,x_1\right)=\min\left\{1,\frac{\pi_1(x_1)q_1(x_0\mid x_1)}{\pi(x_0)q_1(x_1\mid x_0)}\right\} \qquad (2.4)$$

$$a_2\left(x_0,x_1\right)=\min\left\{1,\frac{\pi(x_1)\pi_1(x_0)}{\pi(x_0)\pi_1(x_1)}\right\} \qquad (2.5)$$

Then, (2.3) using the identity $b\min\left\{1,\frac{a}{b}\right\}=a\min\left\{1,\frac{b}{a}\right\}$, which is valid for any two positive numbers a and b, in (2.2) we have

$\pi(x_0\ )\ P_1\ (x_0,dx_1)$

$=\pi(x_0)\ q1\ (x_0,x_1)\ \alpha_1\ (x_0,x_1)\ \alpha_2\ (x_0,x_1)$

$=\pi(x_1)P_1\ (x_1,dx_0).$

(2.6)

Lemma 2.2 Keeping in mind theorem 2.1 and lemma 2.1, it remain to show that for any $x_0,\ \ x_1 \in X$ and $x_0 \neq x_1$ then

$$\pi(x0)\int_X(1-a_2\ \ (x_0,x_1))a_1\ \ (x_0,x_1)P_2\ \ (x_0,y,x_1)dx_1$$

$$=\pi(x_1)\int_X(1\text{-}a_2\ \ (x_1,y))a_1\ \ (x_1,y)P_2\ \ (x_1,y,x_0)dx_0$$

Where

$$P_2\ \ (x_0,y,x_1)=q_2\ \ (x_1\ \ |x_0,y)a_3\ \ (x_0,y,x_1) \qquad (2.7)$$

$$P_2\ \ (x_1,y,x_0)=q_2\ \ (x_0\ \ |\ y,x_1)a_3\ \ (x_1,y,x_0) \qquad (2.8)$$

Proof:

(1)$y \neq x_0$, $y \neq x_1$

In this case we have :

$$\pi(x_0)\int_X(1-a_2\ \ (x_0,x_1))a_1\ \ (x_0,x_1)P_2\ \ (x_0,y,x_1)dx_1$$

$$=\pi(x_0)\int_X(1-a_2\ \ (x_0,x_1))a_1\ \ (x_0,x1)q_2\ \ (x_1\ \ |x_0,y)a_3\ \ (x_0,y,x_1)$$

$$=\pi(x_1)\int_X(1-a_2(x_1,y))a_1\ \ (x_1,y)P_2(x_1,y,x_0)dx_0$$

DOI: https://doi.org/10.30564/ret.v2i2.682

As desired.

(2) $y=x_0$

In this case, the left-hand side of Lemma 2.2 is zero, since

$P_2(x_0,x_0,x_1)$. The last equality holds because we have assumed that $y \neq x_1$. Let's now analyze the righthand side of Lemma 2.2.

$$\pi(x_1)\int_X (1-a_2(x_1,y))a_1\ (x_1,y)P_2(x_1,y,x_0)dx_0$$

$$=\pi(x_1)\int_X (1-a_2\ (x_1,y))a_1\ (x_1,y)q_2\ (x_0\ \mid x_0,x_1)a_3\ (x_1,x_0,x_0)=0$$

(3) $y=x_1$

This case only needs to exchange x_0 and x_1 .Just like the second one, you can prove it in the same way.

Combining Lemmas 2.1 and 2.2, we prove Theorem 2.1.

2.4 Expected Square Jumping Distance

In this section, when considering efficiency for MHDA, MHDR and MHDAR, we need to consider the execution time of the algorithm. So it measured efficiency through Eff, defined by Banterle et al.(2015).

Sherlock and Roberts (2009) focus on unimodal elliptically symmetric targets and show that a proxy for the ACT in finite dimensions is the Expected Square Jumping Distance (ESJD), defined as

$$\mathbb{E}[\| X' - X \|_{\beta}^{2}] = \mathbb{E}[\sum_{i=1}^{d} \frac{1}{\beta_i^2}(X_i' - X)^2]$$

where X and X' are two successive points in the chain and $||\cdot||$ represent the norm on the principal axes of the ellipse rescaled by the coefficients β_i so that every direction contributes equally. [5,8]

We measured efficiency through following formula, it defined by Banterle et al. (2015).

Eff=ESJD/cost per iteration

3 Examples

In this section, to demonstrate the advantage of the MHDAR algorithm, we apply an example with target distribution $f(x,y) \propto exp(-10(x^2-y)^2-(y-1/4)^4)$, $f_1(y)=exp(-(y-1/4)^4)$ proposal distribution is a normal distribution with $\mu=0,\sigma^2=0.75$ and the number of iterations is 50000.

Figure 1. shows that MHDA, MHDR and MHDAR fitting of the target distribution and Figure 2 shows autocorrelation plot of the MHDA, MHDR and MHDAR.

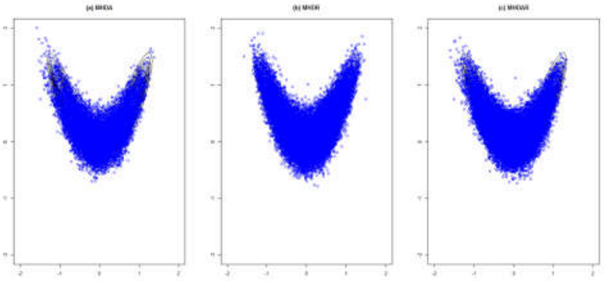

Figure 1. Fitting of target distribution

We generate samplers from target distribution using the three algorithms and calculate the acceptance rate of three algorithms in the following table. From the Table 1 and the Figure 1, we can obtain that acceptance rate of the MHDAR algorithm outperforms the MHDA algorithm. Difference acceptance rate between MHDAR and MHDR is 0.00944. But computation time of MHDAR is just one-third of MHDR.

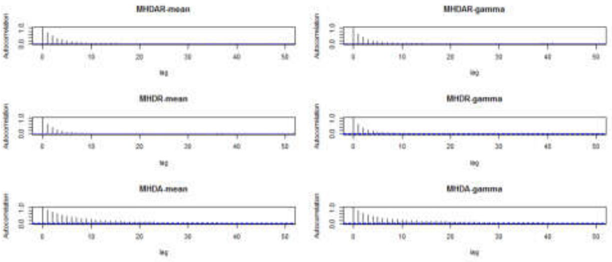

Figure 2. Autocorrelation for each parameter

Table 1. Comparison between MHDA, MHDR and MHDAR in four aspects

Algorithms	a (aver.)	Time (aver.)	ESJD (aver.)	Eff
MHDA	0.215	8.424	0.43504	2582.14
MHDR	0.397	53.491	0.80076	748.50
MHDAR	0.388	18.376	0.72644	196.60

ESJD the expected square jumping distance, a is the acceptance rate, time is the overall computation time.

3.1 Convergence Diagnostics

We use the Geweke(1992) and the Heidelberger-Welch (1983) combining with the coda R package to test chain that generated by the MHDAR algorithm.[9,10] The geweke statistic value of each parameter (as shown in Table 2) and the Z-score scatter plot of each parameter are obtained (As shown in Figure 3). The absolute value of Z-Score of each parameter is less than 1.96, P value is greater than 0.05. So the Markov chain of generated by MHDAR is convergence. It can be seen from Table 3 that the Heidelberger-Welch stability is all passed and the interval half-

DOI: https://doi.org/10.30564/ret.v2i2.682

width test of the parameter is failed (where the halfwidth test is failed that indicates the corresponding confidence interval does not satisfy the accuracy), which shows that the Markov chain generated by sampling is stable.

Table 2. Geweke's statistic for each parameter and associate P-value

	μ	σ^2
Z-score	-0.0412	0.639
P-value	0.9671	0.5228

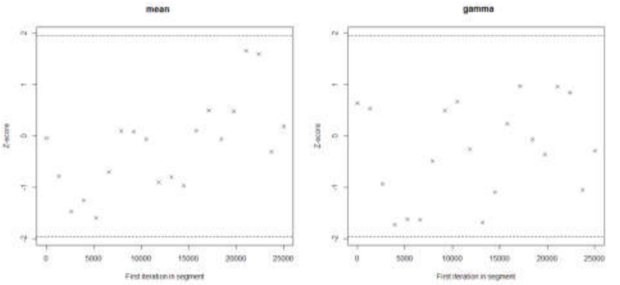

Figure 3. Geweke's scatter diagram for each parameter

Table 3. Heidelberger-Welch stability and the interval half-width test

parameter	Stationarity test	p-value	Halfwidth test	Mean	Halfwidth
μ	passed	0.462	failed	0.00475	0.0111
σ^2	passed	0.862	passed	0.247	0.0066

4. Conclusion

We propose a new MH algorithm combining the ideas of MHDA and MHDR, called MHDAR. The new algorithm, with two stages, reduces the computational cost by division decomposes the prior or likelihood function (that is, the first stage) and increase acceptance ratio by the second stage. We illustrate those accelerating features by a realistic example of two-dimensional distribution. The simulation validates the theoretic results.

References

[1] Hastings, W.K. (1970). Monte Carlo sampling methods using Markov chains and their applications. Biometrika, 57:97-109.

[2] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, and Augusta H. Teller. (1953). Equation of State Calculations by Fast Computing Machines. J. Chemical Physics, 21:1087-1092.

[3] Christen, J. and Fox, C. (2005). Markov chain Monte Carlo using an approximation. Journal of Computational and Graphical Statistics, 14:795-810.

[4] Tierney, L. and Mira, A. (1999). Some adaptive Monte Carlo methods for Bayesian inference. Statistics in Medicine, 18:2507-2515.

[5] Tierney, L. and Mira, A. (1999). Some adaptive Monte Carlo methods for Bayesian inference. Statistics in Medicine, 18:2507-2515.

[6] Banterle, M., Grazian, C., Lee, A., and Robert, C. P. (2015). Accelerating Metropolis-Hastings algorithms by delayed acceptance. URL http://arxiv.org/abs/1406.2660.

[7] Robert, C. and Casella, G. (2004). Monte Carlo Statistical Methods. (2nd ed). Springer Verlag, New York.

[8] Sherlock, C. and Roberts, G. (2009). Optimal scaling of the random walk Metropolis on elliptically symmetric unimodal targets. Bernoulli, 15:774-798. URL http://dx.doi.org/10.3150/08-BEJ176.

[9] Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to calculating posterior moments, Bayesian Statistics 4 (J. Bernardo, J. Berger, A. Dawid and A. Smith, eds.), Oxford: Oxford University Press, 69-194.

[10] Heidelberger P and Welch PD. (1983). Simulation run length control in the presence of an initial transient. Opns Res, 31:1109-44.

DOI: https://doi.org/10.30564/ret.v2i2.682