ANTONIETTA MIRA (*)

On Metropolis-Hastings algorithms with delayed rejection

CONTENTS: 1. Introduction. — 2. The Delaying Rejection Algorithm. — 3. The Symmetric Delaying Rejection Algorithm. — 4. Conclusions. Acknowledgments. References. Summary. Riassunto. Key words.

1. INTRODUCTION

We study the delaying rejection mechanism, a strategy that improves the Metropolis-Hastings algorithm (Tierney, 1994) in the Peskun sense (Peskun, 1973), in that the resulting estimates have, uniformly, a smaller asymptotic variance on a sweep by sweep basis. In Section 2, *The delaying rejection algorithm*, after a description of the basic strategy, we work out an iterative formula for the acceptance probability at the i-th iteration of the delaying rejection process. A special case is discussed in detail in Section 3, *The symmetric delaying rejection*, where the proposal distribution is symmetric and only allowed to depend on the last rejected candidate. We conclude with some final remarks (Section 4).

The key idea behind the Peskun ordering is that when a Markov chain retains the same position over subsequent time, the estimates obtained by averaging along the chain trajectory become less efficient. This has an intuitive explanation: when staying put we fail to explore the state space and increase the autocorrelation along the realized path and thus the variance of the estimates.

For a Metropolis-Hastings algorithm this happens when a candidate generated from the proposal is rejected. Therefore we will be able to improve the Metropolis-Hastings algorithm in the Peskun sense

(*) Università dell'Insubria, Facoltà di Economia, Via Ravasi 2, 21100 - Varese, Italy

by reducing the number of rejected proposals. A way to achieve this goal is proposed in Tierney and Mira, 2001: whenever a candidate is rejected, instead of retaining the same position and advancing time, as in the regular Metropolis-Hastings algorithm, propose a new candidate. The acceptance probability of the new candidate has to be computed in order to preserve the stationary distribution.

If the candidate at the second stage is also rejected we could either retain the starting position or move on to a third stage and so on. As we will show in Section 3, a combination of the two previous strategies can also be considered: when a candidate is rejected toss a p-coin. If the outcome is heads (that is with probability p), propose a new candidate, otherwise stay where you are. To terminate the delaying process it is sufficient to set $p = 0$ at some stage.

An interesting feature of the delaying rejection algorithm (DRA) is that the proposal at later stages is allowed to depend on the rejected values at earlier stages. In order to take full advantage of this strategy it is important to realize that, as the simulation proceeds, the values of the target distribution at points previously rejected become available (these values are computed when evaluating the acceptance probabilities). In a regular Metropolis-Hastings algorithm this information cannot be used in later iterations because the Markovian property would be destroyed. The DRA allows to use information acquired at different stages within the same iteration still retaining the Markovian property for the whole sampler. This permits to perform temporary local adjustments of the proposal distribution.

The natural question that arises is how to take full advantage of the newly acquired information by adjusting the proposal in an efficient way. Some suggestions are given in Mira, 1998, Tierney and Mira, 1999 and Green and Mira, 2001.

2. The Delaying Rejection Algorithm

We begin by giving a more detailed description of how the DRA works. Suppose the position of the chain at time t is $X_t = x$. A candidate y_1 is generated from $q_1(x, dy_1)$ and accepted with probability

$$\alpha_1(x, y_1) = 1 \wedge \frac{\pi(y_1)q_1(y_1, x)}{\pi(x)q_1(x, y_1)} = 1 \wedge \frac{N_1}{D_1} \tag{1}$$

as in a standard Metropolis-Hastings algorithm. The same letter will

be used to indicate both the target distribution, $\pi(dx)$ and the corresponding density $\pi(x)$. The rejection suggests a local bad fit of the current proposal and a better one, $q_2(x, y_1, dy_2)$, should be constructed in light of this. In order to maintain the same stationary distribution the acceptance probability of the new candidate, y_2, has to be properly computed. A possible (but not necessary) way to reach this goal is to impose detailed balance separately at each stage and derive the acceptance probability that preserves it. This is what is done in Tierney and Mira, 1999, to obtain:

$$\begin{aligned}\alpha_2(x, y_1, y_2) &= 1 \wedge \frac{\pi(y_2)q_1(y_2, y_1)q_2(y_2, y_1, x)[1 - \alpha_1(y_2, y_1)]}{\pi(x)q_1(x, y_1)q_2(x, y_1, y_2)[1 - \alpha_1(x, y_1)]} \\ &= 1 \wedge \frac{N_2}{D_2}\,.\end{aligned} \tag{2}$$

If the second stage is reached, it means that $N_1 < D_1$ and we can therefore replace $\alpha_1(x, y_1)$ with N_1/D_1 in D_2 and obtain:

$$\begin{aligned}\alpha_2(x, y_1, y_2) &= 1 \wedge \frac{N_2}{q_2(x, y_1, y_2)[\pi(x)q_1(x, y_1) - \pi(y_1)q_1(y_1, x)]} \\ &= 1 \wedge \frac{N_2}{q_2(x, y_1, y_2)[D_1 - N_1]}\,.\end{aligned} \tag{3}$$

The general i-th stage of the delaying rejection algorithm works as follows. If the candidate y_{i-1} proposed at the previous stage is rejected, generate y_i from $q_i(x, y_1, \cdots, dy_i)$ and accept it with probability

$$\begin{aligned}&\alpha_i(x, y_1, \cdots, y_i) \\ &= 1 \wedge \left\{ \frac{\pi(y_i)q_1(y_i, y_{i-1})q_2(y_i, y_{i-1}, y_{i-2}) \cdots q_i(y_i, y_{i-1}, \cdots, x)}{\pi(x)q_1(x, y_1)q_2(x, y_1, y_2) \cdots q_i(x, y_1, \cdots, y_i)} \right. \\ &\left. \frac{[1-\alpha_1(y_i, y_{i-1})][1-\alpha_2(y_i, y_{i-1}, y_{i-2})] \cdots [1-\alpha_{i-1}(y_i, \cdots, y_1)]}{[1-\alpha_1(x, y_1)][1-\alpha_2(x, y_1, y_2)] \cdots [1-\alpha_{i-1}(x, y_1, \cdots, y_{i-1})]} \right\} \\ &= 1 \wedge \frac{N_i}{D_i}\,.\end{aligned} \tag{4}$$

Again, if the i-th stage is reached, it means that $N_j < D_j$ for $j = 1, \cdots, i-1$, therefore $\alpha_j(x, y_1, \cdots, y_j)$ can be rewritten as N_j/D_j, $j = 1, \cdots, i-1$ and we obtain the recursive formula

$$D_i = q_i(x, \cdots, y_i)(D_{i-1} - N_{i-1})$$

which leads to

$$D_i = q_i(x, \cdots, y_i)[q_{i-1}(x, \cdots, y_{i-1})[q_{i-2}(x, \cdots, y_{i-2}) \cdots [q_2(x, y_1, y_2)[q_1(x, y_1)\pi(x) - N_1] - N_2] - N_3] \cdots - N_{i-1}] . \quad (5)$$

Expression (4) has been worked out by again imposing detailed balance separately at each stage and by forcing the backward path from y_i to x to follow the forward path from x to y_i with time reversed: propose y_{i-1} from y_i, reject it, then propose y_{i-2}, reject again, and so on until finally x is accepted. But from y_i we could instead propose and reject a different state since all the rejected variables are integrated out in the detailed balance equation. For an alternative second (and higher) stage acceptance probability that takes advantage of this observation refer to Green and Mira, 2001.

The described procedure gives rise to a Markov chain which is reversible with invariant distribution π. The average, along a sample path of the chain, of a function f is thus an asymptotically unbiased estimate of $\int f(x)\pi(dx)$.

One of the advantages of having the return path go through y_1 (besides the fact that we do not have to invent a new candidate move and evaluate the target at this new point), is that, since y_1 has been rejected in the forward path, it means that $\alpha_1(x, y_1)$ is likely small. There are good reasons, then, to believe that also $\alpha_1(y_2, y_1)$ will be small. Since the term $1 - \alpha_1(y_2, y_1)$ appears in the numerator of the second stage acceptance probability, this leads to a high acceptance probability of the second stage candidate which is ultimately good in order to explore the whole state space.

3. The Symmetric Delaying Rejection Algorithm

Consider now the special case of a symmetric proposal that only depends on the last rejected candidate value:

$$q_i(x, y_1, \cdots, y_{i-1}, dy_i) = q(y_{i-1}, dy_i) = q(y_i, dy_{i-1}) .$$

In this setting

$$\alpha_1(x, y_1) = 1 \wedge \frac{\pi(y_1)}{\pi(x)} .$$

that is: if $\pi(y_1) \geq \pi(x)$ accept y_1 and set $X_{t+1} = y_1$, otherwise accept y_1 with probability $\pi(y_1)/\pi(x)$. Notice that this acceptance

probability is the same as in the Metropolis-Hastings algorithm when a symmetric proposal is used. If y_1 is rejected, generate y_2 from $q(y_1, dy_2)$ and accept it with probability

$$\alpha_2(x, y_1, y_2) = 1 \wedge \frac{\pi(y_2)\left[1 - 1 \wedge \frac{\pi(y_1)}{\pi(y_2)}\right]}{\pi(x) - \pi(y_1)}. \tag{6}$$

Three cases can occur at this point:

1. if $\pi(y_2) \geq \pi(x)$ then $\alpha_2(x, y_1, y_2) = 1$, thus accept y_2 and set $X_{t+1} = y_2$;
2. if $\pi(y_2) < \pi(y_1)$ then $\alpha_2(x, y_1, y_2) = 0$, thus reject y_2 and move to the next stage (or toss a p-coin);
3. if $\pi(x) > \pi(y_2) \geq \pi(y_1)$ accept y_2 with probability $\alpha_2(x, y_1, y_2) = \frac{\pi(y_2) - \pi(y_1)}{\pi(x) - \pi(y_1)}$.

We can rewrite (6) as:

$$\alpha_2(x, y_1, y_2) = 1 \wedge \frac{0 \vee [\pi(y_2) - \pi(y_1)]}{\pi(x) - \pi(y_1)} = F\left(\frac{\pi(y_2) - \pi(y_1)}{\pi(x) - \pi(y_1)}\right)$$

where F is the cumulative distribution function of a uniform random variable on the interval $(0, 1)$.

If y_2 is rejected, move on to the next stage and draw a new candidate from $q(y_2, dy_3)$. The acceptance probability for this new candidate is:

$$\alpha_3(x, y_1, y_2, y_3) = 1 \wedge \frac{\pi(y_3)}{\pi(x)} \frac{[1 - \alpha_1(y_3, y_2)]\,[1 - \alpha_2(y_3, y_2, y_1)]}{\left[1 - \frac{\pi(y_1)}{\pi(x)}\right]\,[1 - \alpha_2(x, y_1, y_2)]}. \tag{7}$$

Again, if the third stage is reached $\pi(y_1) < \pi(x)$ and $\pi(y_2) < \pi(x)$. Equation (7) can be rewritten as:

$$\alpha_3(x, y_1, y_2, y_3) = 1 \wedge \frac{0 \vee \{\pi(y_3) - [\pi(y_1) \vee \pi(y_2)]\}}{\pi(x) - [\pi(y_1) \vee \pi(y_2)]}.$$

Three cases can occur at this point:

1. if $\pi(y_3) \geq \pi(x)$ then $\alpha_3(x, y_1, y_2, y_3) = 1$, thus accept y_3 and set $X_{t+1} = y_3$;

2. if $\pi(y_3) < [\pi(y_1) \vee \pi(y_2)]$, then $\alpha_3(x, y_1, y_2, y_3) = 0$, thus reject y_3 and move to the next stage;
3. otherwise accept y_3 with probability

$$\frac{\pi(y_3) - [\pi(y_1) \vee \pi(y_2)]}{\pi(x) - [\pi(y_1) \vee \pi(y_2)]}.$$

The i-th stage of the delaying process works as follows. If y_{i-1} is rejected generate y_i from $q(y_{i-1}, dy_i)$. Note that if the i-th stage is reached then $\pi(y_j) < \pi(x)$ for all $j < i$. Let $y^* = \text{argmax}_{j<i}\pi(y_j)$. The acceptance probability for y_i is given by:

$$\alpha_i(x, y_1, \cdots, y_i) = 1 \wedge \frac{0 \vee [\pi(y_i) - \pi(y^*)]}{\pi(x) - \pi(y^*)} \tag{8}$$

Again (8) can be interpreted this way:

1. if $\pi(y_i) \geq \pi(x)$ accept y_i;
2. if $\pi(y_i) < \pi(y^*)$ reject y_j and move to the next stage;
3. otherwise accept y_j with probability

$$P_{x,i} = \frac{\pi(y_i) - \pi(y^*)}{\pi(x) - \pi(y^*)}. \tag{9}$$

The process described above is represented in Figure 1: on the horizontal axis the values of the target distribution at the starting point x and the successive proposed points, $y_1, y_2, \cdots, y_i$ are reported with the various possible positions they can have relative to x and the previously rejected candidates.

The acceptance probability in (9) is considerably simpler than (4) but is obtained under the strong assumption that the proposal can be improved only on the basis of the last rejected candidate. In some sense this assumption weakens the power of the delaying rejection approach which comes from adjusting the proposal in light of all the previously rejected candidates.

An interesting question is the following. Can we show that $P_{x,i}$, or, more generally, that $\alpha_i(x, y_1, \cdots, y_i)$, increases as i tends to infinity? That is, can we show that eventually a candidate will be accepted? Not in the general setting presented because nothing guarantees that the proposal distribution will eventually propose candidates y_i such that $\pi(y_i) > \pi(y^*)$, that is, candidates that have some chances of being

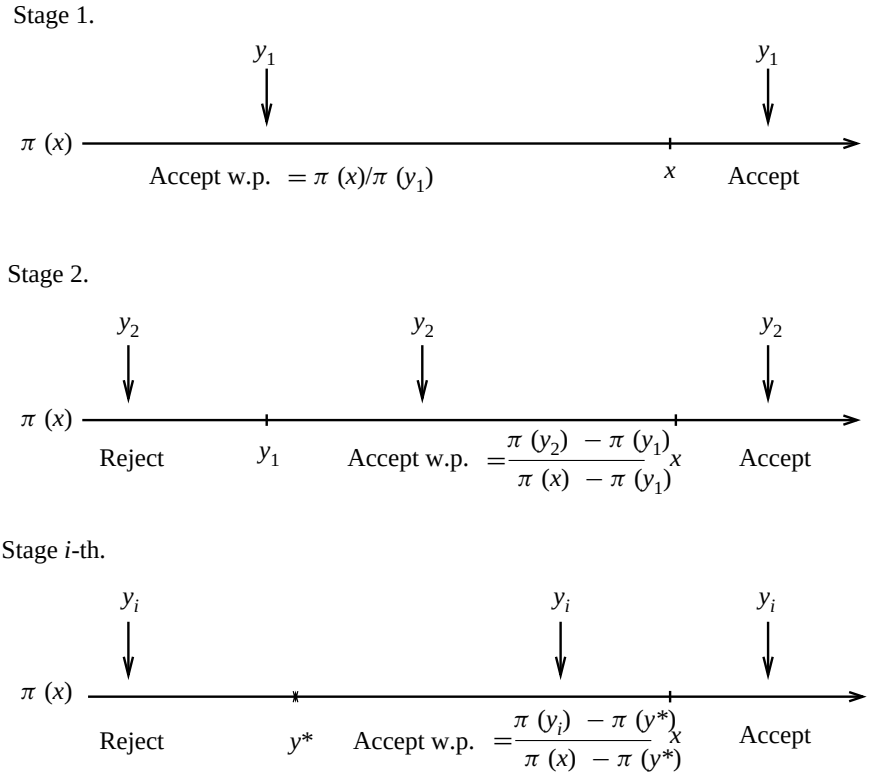

Fig. 1. Symmetric delaying rejection algorithm.

accepted. Only a careful design of the way the proposal is automatically modified at higher stages in view of the rejected candidates can guarantee that this happens. This can be achieved using both some common sense and good knowledge of the problem at hand. For example in the symmetric DRA (but this is a strategy that could be also adopted in a more general setting) good common sense would suggest to use proposals with variance decreasing at higher stages: $\sigma_i = \sigma/i$ or $\sigma_i = \sigma/2^i$ for an overdispersed starting value of σ. This guarantees that, if there is some mass of the target distribution far away from the current position, the chain has some chances of moving there (multiple modes). If this is not the case, by decreasing the dispersion of the proposal we will eventually generate candidates with good probability of being accepted.

Still, in light of the fact that nothing guarantees an automatic increase in the acceptance probability, we would suggest that every time a candidate is rejected we move to the next stage and propose a new candidate with some probability, say p, while with probability $1-p$ the current state is retained and the delaying process ended. This modification of the sampler does not affect the acceptance probabilities computed in Section 2 as the following reasoning shows. The transition kernel of the two stage process, for moving from x to $y_2 \neq x$, provided we insert this additional p-coin step, is given by:

$$q_1(x,y_2)\alpha_1(x,y_2)+p\int q_1(x,y_1)q_2(x,y_1,y_2)[1-\alpha_1(x,y_1)]\alpha_2(x,y_1,y_2)dy_1.$$

A sufficient condition for detailed balance to hold is:

$$\begin{aligned} &p\pi(x)q_1(x,y_1)q_2(x,y_1,y_2)[1-\alpha_1(x,y_1)]\alpha_2(x,y_1,y_2) = \\ &\quad p\pi(y_2)q_1(y_2,y_1)q_2(y_2,y_1,x)[1-\alpha_1(y_2,y_1)]\alpha_2(y,y_1,x) \end{aligned} \tag{10}$$

for all x, y_1 and y_2. Since p cancels we get the same acceptance probability as in Section 2. Similar reasoning can be repeated for subsequent stages, possibly with different values of p at different stages.

4. Conclusions

A little bit of care is needed when we say that the delaying rejection algorithm "performs better" than the Metropolis-Hastings algorithm.

Up to now we have only compared the two schemes in terms of the Peskun ordering, that is, the asymptotic variance of MCMC estimates obtained using the delaying rejection algorithm is smaller, on a sweep by sweep basis, than the one obtained using the Metropolis-Hastings scheme (regardless of the function f whose expectation with respect to the stationary distribution of the Markov chain we are interested in evaluating).

But it is important to realize that one sweep in the delaying rejection scheme, that is a move from X_t to X_{t+1} may take considerably more CPU time than one sweep in the Metropolis-Hastings algorithm. This is due to the fact that, when using the delaying rejection strategy, we have to generate the sequence $y_1, y_2, \ldots$ and possibly evaluate more than one acceptance probability.

A more reasonable comparison should be made by taking into account this lack of symmetry between the two schemes. One possibility is to compare the asymptotic variance of the two competing schemes given a fixed number of evaluations of the target distribution.

Let τ_{DR} be the mean CPU time needed for the delaying rejection algorithm to go from X_t to X_{t+1}. Let τ_{MH} be the mean CPU time needed for the Metropolis-Hastings algorithm to move from X_t to the first $X_{t+j} \neq X_t$, $j = 1, 2, \ldots$ Consider implementing a delaying rejection algorithm where the proposal distribution is not adjusted to take advantage of the newly acquired information. In other words, take the proposal distribution used for the delaying rejection algorithm at any stage to be the same as the one used for the Metropolis-Hastings algorithm. Then τ_{DR} and τ_{MH} should be comparable. But if we construct the proposal distribution in the delaying rejection algorithm in a clever way we should be able to make $\tau_{\mathrm{DR}} < \tau_{\mathrm{MH}}$. This means that we should be able to construct delaying rejection algorithms that outperform Metropolis-Hastings algorithms not only because they have smaller asymptotic variance of MCMC estimates but also because they have smaller asymptotic variance given a fixed amount of CPU time. For accurate simulation studies in this direction refer to Tierney and Mira, 1999 and Green and Mira, 2001.

Finally the performance of a MCMC sampler can be evaluated not only by the asymptotic variance of the resulting estimates but also in terms of the speed of convergence to stationarity (measured by total variation distance). As observed by Besag and Green, 1993, the two goals, asymptotic variance and speed of convergence, lead to different notions of optimality, since the former depends on the eigenvalues and the latter on the absolute values of the eigenvalues. Thus, while the optimal sampler will be different, we could get good performance with respect to both criteria simultaneously.

Consider the special case of a transition kernel with positive eigenvalues (finite state space) or positive spectrum (general state space) as the independence Metropolis-Hastings sampler, for example. Then the introduction of the delaying rejection mechanism, besides being beneficial in terms of asymptotic variance, could improve the speed of convergence and certainly does not slow down the original sampler. In general, however, it is not clear what the effect of the DRA is, in terms of the speed of convergence, therefore some care and further research are needed in this respect.

We would suggest using a sampler with good convergence properties for the first part of the simulation (until convergence is presumably achieved) such as the Metropolis Adjusted Langevin Algorithm (MALA, Besag, 1994) and then switching to a sampler with better properties in terms of asymptotic variance by adding, for example, a delaying rejection move to the original sampler.

A combination that might be worth studying is a sampler which uses a first stage regular Metropolis proposal and a second stage Metropolis adjusted Langevin proposal. The rationale behind this is that the second stage proposal requires the computation of the first derivative of the (log) target distribution. This proposal has good properties (is "self-targeting") but might be expensive to compute and thus we want to use it only when a more standard proposal fails.

ACKNOWLEDGMENTS

I would like to thank Luke Tierney: this paper is part of my dissertation completed under his precious and careful guidance at School of Statistics, University of Minnesota.
I gratefully acknowledge the Graduate School of the University of Minnesota for supporting my research with a Doctoral Dissertation Fellowship.
Finally, the discussions with Peter Green on the delaying rejection strategy have been very valuable and gave rise to an extension of the algorithm in a varying dimensional setting.

REFERENCES

BESAG, J.E (1994) Comments on "Representations of knowledge in complex systems" by U. GRENANDER and M. I. MILLER, *Journal of the Royal Statistical Society, Series B*, 56, 569–603.

BESAG, J. and GREEN, P.J. (1993) Spatial Statistics and Bayesian computation, *Journal of the Royal Statistical Society, Series B*, 55, 25-37.

GREEN, P.J. and MIRA, A. (2001) Delaying rejection in reversible jump Metropolis-Hastings, *Biometrika*, 88, 3, to appear.

MIRA, A. (1998) *Ordering, Slicing and Splitting Monte Carlo Markov Chains*, Ph.D. thesis, School of Statistics, University of Minnesota.

PESKUN, P.H. (1973) Optimum Monte Carlo sampling using Markov chains, *Biometrika*, 60, 607–612.

TIERNEY, L. (1994) Markov chains for exploring posterior distributions, *Annals of Statistics*, 22, 1701-1762.

TIERNEY, L. and MIRA, A. (1999) Some adaptive Monte Carlo methods for Bayesian inference, *Statistics in Medicine*, 8, 2507-2515.

On Metropolis-Hastings algorithms with delayed rejection

SUMMARY

The class of Metropolis-Hastings algorithms can be modified by delaying the rejection of proposed moves. The new samplers are proved to perform better than the original ones in terms of asymptotic variance of the estimates on a sweep by sweep basis. The delaying rejection algorithms also allow some space for local adaptation of the proposal distribution. We give an iterative formula for the acceptance probability at the i-th iteration of the delaying process. A special case is discussed in detail: the delaying rejection algorithm with symmetric proposal distribution.

Algoritmi di Metropolis-Hastings con rifiuto ritardato

RIASSUNTO

La classe degli algoritmi di Metropolis-Hastings può essere modificata attraverso il meccanismo del rifiuto ritardato delle mosse proposte. Si dimostra che i nuovi algoritmi così ottenuti hanno una migliore performance di quelli originari in termini di minor varianza asintotica degli stimatori ottenuti (a parità di numero di iterazioni). Gli algoritmi con rifiuto ritardato permettono inoltre di adattare localmente la distribuzione usata per proporre nuovi candidati. In questo lavoro diamo una formula iterativa per calcolare la probabilità di accettazione all'i-esima iterazione del processo di ritardo del rifiuto. Infine analizziamo in dettaglio un caso particolare: l'algoritmo di riufiuto ritardato con distribuzione proponente simmetrica.

KEY WORDS

Markov chain Monte Carlo Methods; Metropolis-Hastings algorithm; Asymptotic variance; Peskun ordering.

[Manuscript received June 2000; final version received February 2001.]