2nd Symposium on Advances in Approximate Bayesian Inference, 2019 1–6

Non-reversibly updating a uniform $[0, 1]$ value for accept/reject decisions

Radford M. Neal RADFORD@CS.UTORONTO.CA
University of Toronto, Vector Institute Affiliate

Abstract

I show how it can be beneficial to express Metropolis accept/reject decisions in terms of comparison with a uniform $[0, 1]$ value, and to then update this uniform value non-reversibly, as part of the Markov chain state, rather than sampling it independently each iteration. This provides a small improvement for random walk Metropolis and Langevin updates in high dimensions. It produces a larger improvement when using Langevin updates with persistent momentum, giving performance comparable to that of Hamiltonian Monte Carlo (HMC) with long trajectories. This is of significance when some variables are updated by other methods, since if HMC is used, these updates can be done only between trajectories, whereas they can be done more often with Langevin updates. This is seen for a Bayesian neural network model, in which connection weights are updated by persistent Langevin or HMC, while hyperparameters are updated by Gibbs sampling.

A decision to accept or reject a Metropolis proposal to move from x to x^* can be done by checking whether $u < \pi(x^*)/\pi(x)$, where π is the density function for the distribution being sampled, and u is a uniform $[0, 1]$ random variable. Standard practice is to generate a value for u independently for each decision. I show here that it can be beneficial to instead update u each iteration without completely forgetting the previous value, using a non-reversible method.

Doing non-reversible updates to u will not change the fraction of proposals that are accepted, but can result in acceptances being clustered together (with rejections similarly clustered). This can be beneficial, for example, when rejections cause reversals of direction, as in Horowitz's (1991) variant of Langevin updates with persistent momentum.

1. The New Method for Accept/Reject Decisions

For any MCMC method, we can augment the variable of interest, x, with density $\pi(x)$, by a variable s, whose conditional distribution given x is uniform over $[0, \pi(x)]$. The resulting joint distribution for x and s is uniform over the region where $0 < s < \pi(x)$. This is the view underlying "slice sampling" (Neal 2003), in which s is introduced temporarily, by sampling uniformly from $[0, \pi(x)]$, and then forgotten once a new x has been chosen. Metropolis updates can also be viewed in this way, with the new x found by accepting or rejecting a proposal, x^*, according to whether $\pi(x^*) > s$, with s newly sampled from $[0, \pi(x)]$.

However, it is valid to instead retain s in the state between updates, utilizing its current value for accept/reject decisions, and updating this value when desired by any method that

© R.M. Neal.

leaves invariant the uniform distribution on $[0, \pi(x)]$ (since this is the conditional distribution for s given x). Equivalently, one can retain in the state a value u whose distribution is uniform over $[0, 1]$, independent of x, with u corresponding to $s/\pi(x)$. Accept/reject decisions are then made by checking whether $u < \pi(x^*)/\pi(x)$. Note, however, that if x^* is accepted, u must then be updated to $u\,\pi(x)/\pi(x^*)$, which corresponds to s not changing.

Here, I will consider non-reversible updates for u, which translate it by some fixed amount, δ, and perhaps add some noise, reflecting off the boundaries at 0 and 1. It is convenient to express such an update with reflection in terms of a variable v that is uniform over $[-1, +1]$, and define $u = |v|$. An update for v can then be done as follows:

```
$v \leftarrow v + \delta + \text{noise}$
while $v > +1$ do $v \leftarrow v - 2$
while $v < -1$ do $v \leftarrow v + 2$
```

For any δ and any distribution for the noise (not depending on the current value of v), this update leaves the uniform distribution over $[-1, +1]$ invariant.

The full state consists of x and v, with x having density $\pi(x)$ and, independently, v being uniform over $[-1, +1]$, which corresponds to the conditional distribution of $s = |v|\pi(x)$ given x being uniform over $[0, \pi(x)]$. If a proposed move from x to x^* is accepted we change v to $v\,\pi(x)/\pi(x^*)$, which leaves s unchanged (allowing the reverse move). Because of this change to v on acceptance, when $\pi(x)$ varies continuously, it may not be necessary to include noise in the update for v, but if $\pi(x)$ has only a finite number of possible values, adding noise may be necessary to ensure ergodicity.

The hope with these non-reversible updates is that u will move slowly (if δ and the noise amount are small) between values near 0, where acceptance is easy, and values near 1, where acceptance is harder. (But note that u may change in either direction when proposals are accepted, though s will not.) Non-reversibly updating u will not change the overall acceptance rate, but it is expected that acceptances and rejections will become more clustered — with an accepted proposal more likely to be followed by another acceptance, and a rejected proposal more likely to be followed by another rejection.

We might wish to intermingle Metropolis updates for x that use v to decide on acceptance with other sorts of updates for x — for example, Gibbs sampling updates, or Metropolis updates accepted in the standard fashion. We can do these updates while ignoring v, and then simply resume use of v afterwards, since v is independent of x. We could also include several independent v variables in the state, using different v values for different classes of updates, but this generalization is not explored here.

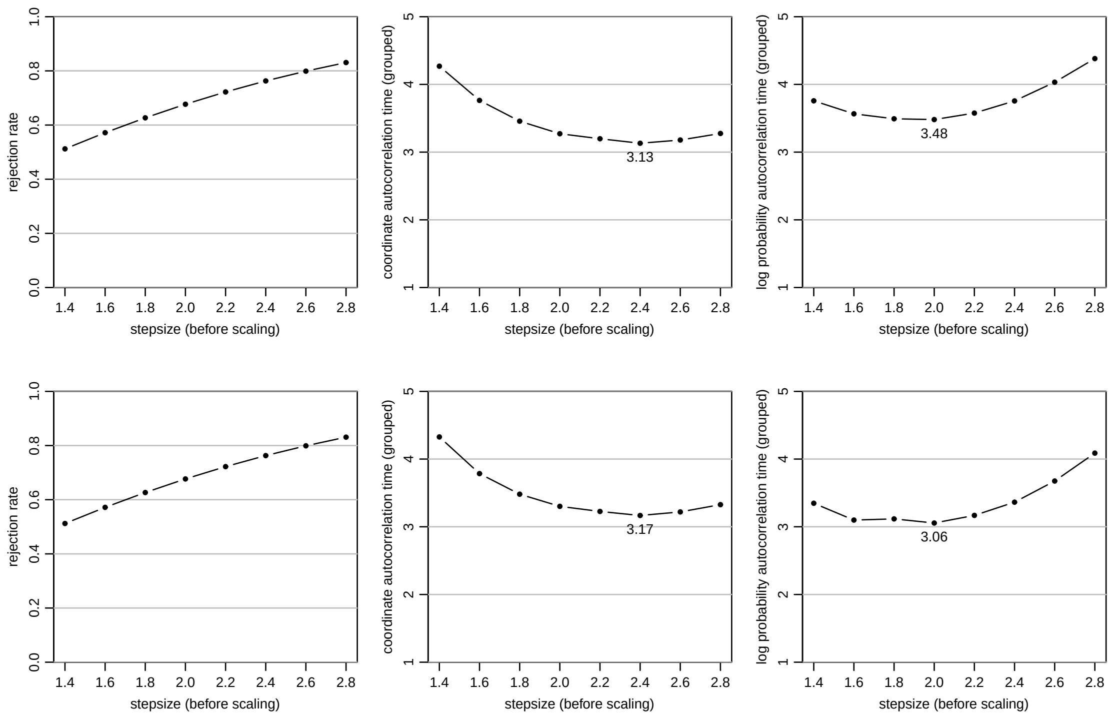
2. Results for random-walk Metropolis updates in high dimensions

A small benefit from non-reversible updating of u can be seen with simple random-walk Metropolis updates. Such updates operate as follows:

1) Propose $x^* \sim N(x, \sigma^2 I)$.

2) Accept $x' = x^*$ as next state if $u < \frac{\pi(x^*)}{\pi(x)}$,; otherwise let $x' = x$.

Here are results when sampling a 40-dimensional Gaussian distribution with identity covariance matrix:



Top: Standard Metropolis. Bottom: Non-reversible with $\delta = 0.2$, no noise.

The values for σ used were the stepsizes shown above scaled down by $40^{1/2}$. The autocorrelation times (one plus twice sum of autocorrelations) shown are for groups of 40 iterations (hence the single-iteration autocorrelation time is about 40 times larger).

When estimating the mean of a single coordinate, little difference is seen between the standard method and using a non-reversible update for u. But for estimating the mean of the log of the probability density, the non-reversible method is about 1.14 times better. A similar small benefit is seen for simple Langevin updates.

One possible explanation for the improvement is that, as noted by Caracciolo, et al (1994), the performance of Metropolis methods in high dimensions is limited by their ability to sample different values for the log of the density. In D dimensions, the log density typically varies over a range proportional to $D^{1/2}$. A Metropolis update will typically change the log density only by about one — larger decreases in the log density are unlikely to be accepted, and it follows from reversibility that increases in the log density of much more than one must also be rare (once equilibrium has been reached). Since standard Metropolis methods are reversible, these changes of order one will occur in a random walk, and so around D steps will be needed to traverse the range of log densities of about $D^{1/2}$, limiting the speed of mixing.

The gain seen from using non-reversible updates for u may come from helping with this problem. When u is small few proposals will be rejected, and the chain will tend to drift towards smaller values for the log density, with the opposite behaviour at times when u is near one. This could reduce the random walk nature of changes in the log density.

3. Results for Langevin updates with persistent momentum

I obtained more interesting results when applying the non-reversible acceptance method to the one-step, non-reversible version of Hamiltonian Monte Carlo (Duane, et al 1987) due to Horowitz (1991). This method is a "persistent" form of "Langevin" update. See the review by Neal (2011) for more discussion of these methods.

Hamiltonian Monte Carlo works in an extended state space with momentum variables, p, newly sampled each iteration. It proposes a new value for (x, p) by simulating Hamiltonian dynamics with some number of "leapfrog" steps (and then negating p, so the proposal is reversible). A leapfrog step has the form

$$\begin{aligned} p_{t+\eta/2} &= p_t \;-\; (\eta/2)\,\nabla U(x_t) \\ x_{t+\eta} &= x_t \;+\; \eta\, p_{t+\eta/2} \\ p_{t+\eta} &= p_{t+\eta/2} \;-\; (\eta/2)\,\nabla U(x_{t+\eta}) \end{aligned}$$

In the limit as the stepsize η goes to zero, the proposed point will always be accepted. If many leapfrog steps are used, the proposed point can be distant from the current point, avoiding the slowdown from doing a random walk with small steps.

In Horowitz's method, only one leapfrog step is done, but a trick is used so that these steps nevertheless usually keep going in the same direction, except on a rejection. These updates operate as follows:

1) Set $p' = \alpha p \;+\; \sqrt{1-\alpha^2}\, n$, where $n \sim N(0, I)$, and $x' = x$.
2) Find (x^*, p^*) from (x', p') with one "leapfrog" step (as in HMC), with stepsize η.
3) Accept $(x'', p'') = (x^*, -p^*)$ if $u < \frac{\pi(x^*, -p^*)}{\pi(x', p')}$; otherwise $(x'', p'') = (x', p')$.
4) Let $p''' = -p''$ and $x''' = x''$.

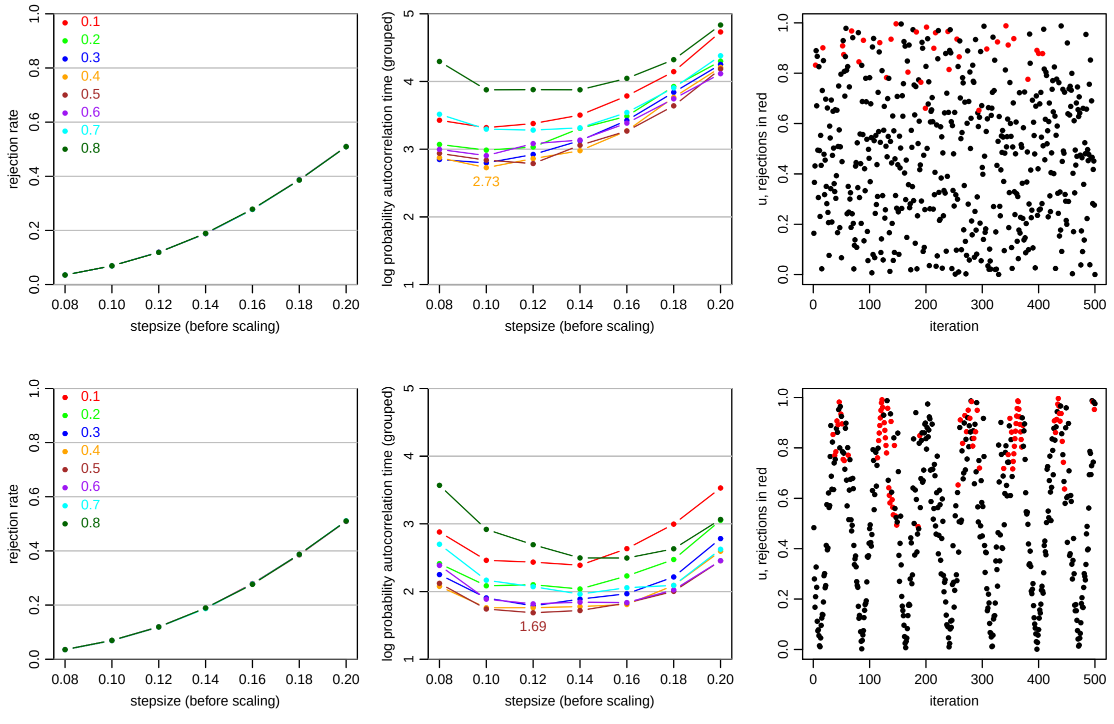
For α near 1, Step (1) only slightly changes p. If Step (3) accepts, the negation in the proposal is canceled by the negation in Step (3). But a rejection will reverse p, leading the chain to almost double back on itself.

Unfortunately, even with this non-reversibility trick, Horowitz's method is not as efficient as HMC with long trajectories. To reduce the rejection rate, and hence random-walk inducing reversals of direction, a small, inefficient step size (η) is needed.

But a higher rejection rate would be tolerable if rejections cluster in time, producing long runs of no rejections. For example, rejection-free runs of 20, 0, 20, 0, ... are better than rejection-free runs of 10, 10, 10, 10, ..., since N of the former runs will move via a random walk a distance proportional to $20(N/2)^{1/2} \approx 14\,N^{1/2}$, whereas N of the latter runs will move only a distance proportional to $10\,N^{1/2}$.

I tried sampling with the standard persistent Langevin method and with persistent Langeving updates with non-reversible updating of u, on a multivariate Gaussian distribution consisting of 16 independent pairs of variables having variances of 1 and correlations of 0.99. (Ie, a 32-dimensional Gaussian with block-diagonal covariance matrix).

The plots below show the results. The values for η used were the stepsizes shown scaled down by $32^{1/6}$. The curves in different colours are for $\alpha \in \{0.1^\eta, ..., 0.8^\eta\}$. The autocorrelation times shown are for groups of 31 iterations.



Top: Standard persistent Langevin. Bottom: Non-reversible with $\delta = 0.03$, no noise.

The non-reversible method is 1.62 times better estimating the mean log probability. The right plots show that rejections are indeed clustered when non-reversible updates are used, which reduces random walks, explaining the improvement.

4. Use for Bayesian neural networks

I tried using the persistent Langevin method with non-reversible updates for u to sample the posterior distribution of a Bayesian neural network model. Such models (Neal 1995) typically have hyperparameters controlling the variance of groups of weights in the network. It is convenient to use Gibbs sampling updates for these hyperparameters, alternating such updates with HMC updates for the network weights. However, when long trajectories are used for HMC, as is desirable to reduce random-walk behaviour, the Gibbs sampling updates for hyperparameters are done infrequently. Using persistent Langevin updates for weights would allow hyperparameters to be updated more frequently, perhaps speeding overall convergence. We hope to make this work better by non-reversibly updating u.

I tested this approach on a binary classification problem, with 5 inputs, and 300 training cases. A network with one hidden layer of 12 tanh hidden units was used. Only two of the inputs for this problem were relevant, with two more being slightly noisy versions of the two relevant inputs, and one input being independent noise. Five separate hyperparameters controlled the variance of weights out of each inputs.

For the best-tuned persistent Langevin method non-reversibly updating u, the average autocorrelation time for the four plausibly-relevant input hyperparameters was 1.25 times smaller than for the best-tuned HMC method. This is an encouraging preliminary result.

5. Bibliography

Caracciolo, S, Pelisseto, A, and Sokal, A. D. (1994) “A general limitation on Monte Carlo algorithms of Metropolis type”, *Physical Review Letters*, vol. 72, pp. 179-182. Also at `arxiv.org/abs/hep-lat/9307021`

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987) “Hybrid Monte Carlo”, *Physics Letters B*, vol. 195, pp. 216-222.

Horowitz, A. M. (1991) “A generalized guided Monte Carlo algorithm”, *Physics Letters B*, vol. 268, pp. 247-252.

Neal, R. M. (2003) “Slice sampling” (with discussion), *Annals of Statistics*, vol. 31, pp. 705-767.

Neal, R. M. (2011) “MCMC using Hamiltonian dynamics”, in the *Handbook of Markov Chain Monte Carlo*, S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng (editors), Chapman & Hall / CRC Press, pp. 113-162. Also at `arxiv.org/abs/1206.1901`

Neal, R. M. (1995) *Bayesian Learning for Neural Networks*, Ph.D. Thesis, Dept. of Computer Science, University of Toronto, 195 pages.