ADAPTIVE METROPOLIS-HASTINGS SAMPLING, OR MONTE CARLO KERNEL ESTIMATION

CHRISTOPHER A. SIMS

Abstract. A new algorithm for sampling from an arbitrary pdf.

1. Introduction

Consider the standard problem of sampling from a distribution for which the pdf p can be calculated, up to a factor of proportionality, at each point in the sample space, but for which no simple algorithm exists for generating a random sample. One approach, called importance sampling, is to find a good approximation q to p for which drawing a sample is easy, then to weight the observations drawn by the ratio p/q. The resulting weighted sample can then be used in many respects as if it were a random sample from the p distribution. In practice, though, especially in high-dimensional situations, it often turns out that there are regions in which p/q is extremely large. These regions are difficult to identify in advance, often occurring where p itself is small, but the ratio is large because q is even smaller. When the artificial sample turns out for this reason to have a few observations with very large weight that dominate it, the sampling scheme is very inefficient.

An alternative is Metropolis sampling. It generates a j'th draw by drawing z from a "jump" distribution with pdf $q(z|\theta_{j-1})$ satisfying $q(z-\theta|\theta) = q(\theta - z|\theta)$ for all z. It then sets $\theta_j = z$ with probability $\min(1, p(z)/p(\theta_{j-1}))$, $\theta_j = \theta_{j-1}$ otherwise. This scheme avoids handling a region of unexpectedly high p by simply weighting a single draw there heavily. Instead it adapts by lingering in the area of the unexpectedly high p. This adaptability comes at a price, however, as the Metropolis algorithm provides no route by which to use knowledge of the global shape of p.

The Metropolis-Hastings algorithm removes the requirement that the jump distribution in the Metropolis algorithm be symmetric. It allows sampling from a global approximation q to p at every step of the algorithm. Just as with importance sampling, in the limiting case $q = p$, the algorithm produces a simple random sample from the p distribution. With every draw taken from the same fixed q, however, It

Date: December 3, 1998.

Key words and phrases. markov chain Monte Carlo, Metropolis-Hastings, numerical integration.

I have benefited from comments by John Geweke and Tao Zha.

reacts to a region of unexpectedly high p/q much as does importance sampling, as it tends simply to repeat the same observation many times once it hits such a region.

This paper proposes a method that combines the Metropolis algorithm's adaptability with importance sampling's ability to exploit knowledge of a good global approximate q. The idea is this: at draw j, use draws $1, \dots, j-1$ to form a non-parametric kernel estimate of p, called q_{j-1}. Sample from this distribution, and use the Metropolis-Hastings rule to decide whether to accept the new draw z or repeat the previous draw. The result is not a Markov chain, but in the limit as $j \to \infty$ the usual proof that a Metropolis-Hastings sampling scheme has a fixed point at the true pdf goes through.

Proving that the algorithm converges to the true pdf from an arbitrary startup may turn out to be difficult. However in practice it often converges very rapidly, in terms of iteration count, quickly approaching the efficiency that would be obtained by sampling directly from p.

Drawing the new candidate sample value from the kernel estimate is computationally easy, requiring a single draw from a uniform distribution plus a single draw from the distribution defined by the kernel pdf . Computing the pdf values

$$q(z|\theta_j, \theta_{j-1}, \theta_{j-2}, \dots)\,, \quad q(\theta_j|z, \theta_{j-1}, \theta_{j-2}, \dots)\,, \tag{1.1}$$

which are required for applying the Metropolis-Hastings jump rule, is more computationally demanding, requiring j evaluations of the kernel, twice. When j gets to be very large, therefore, it may make sense to pick a fixed number J and use a random sample of J from the previous j draws[1] in forming the kernel estimate.

2. Details

We suppose we are given the true pdf p and an algorithm for evaluating it. We also suppose that we are given an initial, or base, pdf q that is the kind of slightly over-dispersed good approximation to p that we might use for importance sampling. It must be easy to draw from. In many applications q will be based on the Gaussian asymptotic approximation to p constructed from the second order Taylor expansion of $\log p$ about the maximum likelihood estimator $\hat{\theta}$. Usually the Gaussian approximation is replaced by a corresponding multivariate t distribution with a fairly low degrees of freedom parameter. We choose a kernel pdf, g. A natural choice is the q pdf scaled down. It is difficult to know by what factor to scale down q in choosing g. One guideline is that N ellipses of the size of the $\chi^2 = 1$ contour of the g distribution

[1] In a previous version of this paper it was suggested that the J most recent draws could be used. But the argument for a fixed point at the true distribution depends on vanishing dependence between the set of θ_k's other than θ_j used to form the kernel estimate and the θ_j, z_{j+1} pair. Thus use of any fixed set of J lags for this purpose results in bias in the limiting distribution, even though if J is large the bias will be correspondingly small. I am grateful to Tao Zha for uncovering the bad behavior of the algorithm when J is finite and the J most recent draws are used in forming the kernel.

should be capable of roughly filling the $\chi^2 = 1$ contour of the q distribution, where N is the number of points to be used in forming the kernel approximation over a substantial portion of the iterations. If all the past draws are being used, N might be some fraction of the number of draws expected, say one fifth or one third. If a fixed number J of past draws are redrawn at each iteration, $N = J$. This suggests a scale parameter for g of roughly $N^{-1/d}$, where d is the dimension of the θ vector. For example, if one guesses that about 15000 draws will be needed of a 20-element θ vector, one might scale the q distribution by $5000^{-1/20} = .65$ or $3000^{-1/20} = .67$. For 500 draws of a 2-element θ one would use a factor of about .1 or .08. Finally the algorithm requires that we choose a parameter n_b that determines the relative weight to put on q and the kernel estimates as the algorithm starts up.

At iteration $j+1$ of the algorithm we have previous draws $\{\theta_i, i = 1, \dots j\}$. We draw the random variable z_{j+1} from a distribution that mixes the base pdf q, with weight $n_b/(n_b + j)$, with j pdf's, each of which is centered at one of the previous θ_i's and has the shape of the g pdf. The overall pdf then is

$$h(z; \{\theta_i\}_{i=1}^{j}) = \frac{n_b \cdot q(z) + \sum_{i=1}^{j} g(\theta_i - z)}{n_b + j} \,. \tag{2.1}$$

To implement a draw from this distribution, we draw ν_1 from a uniform distribution on $(0, 1)$. If $(n_b + j)\nu_1 \leq n_b$, we draw z from the q distribution. Otherwise we find the smallest integer k exceeding $(n_b + j)\nu_1 - n_b$ and draw z from the $g(\theta_k - z)$ pdf.

We use the usual Metropolis-Hastings rule for deciding whether to use our draw of z as θ_j, or instead to discard it and set $\theta_j = \theta_{j-1}$. That is, we set

$$R = \left. \frac{p(z)}{h\left(z; \theta_j, \theta_{j-1}, \dots\right)} \middle/ \frac{p(\theta_j)}{h\left(\theta_j; z, \theta_{j-1}, \dots\right)} \right. \,, \tag{2.2}$$

and "jump" (use z) if $R > \nu_2$, where ν_2 is a second random draw from a uniform distribution on $(0, 1)$.

It is likely that we will want to store the $p(\theta_j)$ sequence in any case, and it is computationally useful to store also the sequence

$$h_{j+1} = h(\theta_{j+1}; \theta_j, \theta_{j-1}, \dots) \,. \tag{2.3}$$

If we do so, then the evaluation of the second h in (2.2), with reversed arguments, simplifies to

$$h\left(\theta_j; z, \theta_{j-1}, \dots\right) = \frac{(n_b + j - 2)h_{j-1} + g(z - \theta_j)}{j - 1} \,. \tag{2.4}$$

In the variant of the algorithm that uses a fixed number of J points randomly drawn from the set of previous draws in forming the kernel, there is of course an intermediate step of making these draws. Also, in this case it is not possible to obtain the simplification in (2.4), because the points playing the role of $\theta_{j-s}, s \geq 1$ are a different set for each j.

3. Properties of the Algorithm

The standard proof that, if θ_{j-1} was drawn from a target pdf π, the marginal distribution of θj is also π, applies to this algorithm just as it does to ordinary Metropolis-Hastings sampling. At each j, in fact, this algorithm is just a special case of Metropolis-Hastings. The innovation here is only the suggestion that the jump distribution be updated according to a systematic rule at every j. However, what the Metropolis-Hastings argument applied to this algorithm shows is that if the *conditional* distribution of $\theta_j \mid S_j$ is the target, where S_j is the set of previous θ's used in forming the distribution from which z_{j+1} is drawn, then the *conditional* distribution of $\theta_{j+1} \mid S_j$ will be the same. But at the next step of the algorithm, the conditioning set changes from S_j to S_{j+1} as θ_j becomes part of the history used in forming the kernel estimate. it is therefore not guaranteed that the conditional distribution of $\theta_{j+1} \mid S_{j+1}$ matches the target, which is what is needed for the fixed point argument. However, when the full set of past draws is used, or when a fixed number J of past draws, drawn from the full set of past draws, is used, the change in the approximating distribution from adding the most recent observation to the list of draws becomes negligible as j increases, so that the Metropolis-Hastings argument comes closer and closer to applying exactly and recursively for large j.

Of course it would be a good idea to develop a rigorous proof of this point. In the meantime, researchers who are uneasy about the properties of the algorithm have another way to minimize their concerns. Two parallel sequences A and B of iterations can be set up, A following the algorithm suggested above, B instead using the previous values of the A sequence's draws in forming the kernel estimates. Then the update of S for B is always taking place in the other chain. The standard Metropolis-Hastings argument shows that if the distribution of $\theta_j^B \mid \left\{\theta_k^A \mid k < j\right\}$ matches the target (and hence is independent of $\left\{\theta_k^A \mid k < j\right\}$), then the distribution of $\theta_{j+1}^B \mid \left\{\theta_k^A \mid k < j\right\}$ matches the target. Since by construction θ_j^A and θ_{j+1}^B are conditionally independent given $\left\{\theta_k^A \mid k < j\right\}$[2] the distribution of $\theta_{j+1}^B \mid \left\{\theta_k^A \mid k < j+1\right\}$ will match the target as well, completing the fixed point argument. This two-track version of the algorithm might be the best way to run it in general. Without the second track, standard methods of assessing convergence of the sequence apply, but they can never tell us whether the bias from dependence has become small yet. Comparing results from the two sequences provides a way to assess the size of the bias. If it is small, the results from the two sequences can be merged; if it is large, but sequence B appears converged, results from the A sequence can be discarded. It is an interesting question whether the B sequence is likely to be notably less efficient than the A sequence.[3]

[2] See Appendix A

[3] A version of the algorithm that more obviously has the correct fixed point would omit θ_{j-1}^B from the set of θ's used to form the pdf for the new candidate draw z_j. This makes the pdf for the new draw completely independent of the B sequence. However, it also means that when θ_{j-1}^B has a very

This algorithm is not Markovian with stationary transition rule, so that a proof that it must converge requires a new argument, which this version of this paper also does not supply. The algorithm has been applied successfully to a number of simple one-dimensional cases and to a rather well-behaved 20-parameter simultaneous equations model.

4. Assessing Convergence

This algorithm's convergence can be assessed by the same kind of measure used for Gibbs or Metropolis sampling. One can start several replicates of the algorithm, producing K sequences θ_{jk}, $k = 1, \dots, K$, $j = 1, \dots, M$. Or in the case of the two-sequence version, one can start several pairs of replicates. If all the subsequences were i.i.d. draws from the same finite-variance distribution, the ratio of the sample variance of one sequence to the sample variance of sequence sample means would be about M. This ratio is then a measure of effective sample size in a single one of the K sequences. As suggested by Gelman, Carlin, Stern, and Rubin (1995), this measure can differ for different functions of the θ vector, and can be computed for the specific functions of interest rather than just the elements of the θ vector itself. Random subsamples of the θ sequence of size up to about effective sample size should behave as if i.i.d. for purposes of assessing the Monte Carlo error in sample averages. Using larger subsamples, or the full sequence, will improve accuracy further, but not in proportion to the number of additional Monte Carlo observations used. When trying to evaluate $E[f(\theta)]$ for an f that is expensive to compute, therefore, it may be worthwhile to use a subsample considerably smaller than the full Monte Carlo sample.

Because the AMH algorithm does not generate simple serial correlation along the lines of that from the Metropolis algorithm, assessing the accuracy measure may be more difficult. At the start, each of the sequences is approximately simply sampling from q. If the q pdf is a fairly good approximation to p, it may be only after many draws that its problems start to appear and effective sample size ceases to increase in proportion to M. In this respect AMH is like importance sampling or Metropolis-Hastings with a fixed q, not centered at θ_{j-1}, as jump distribution. However, unlike these other two, the AMH algorithm has a "self-repairing" property, and may eventually start producing effective sample sizes close to M. It can do so if the kernel density g is concentrated enough, and M (or in the case of a bounded number of points used in forming the kernel, J) is large enough, so that a kernel estimate of the pdf based on the Monte Carlo sample is very close to p.

high importance ratio, the algorithm will tend to stick at that point much longer than it would with the algorithm described in the text.

Appendix A. Proof of the Fixed Point Property for the Two-Sequence Algorithm

We use the notation $S_j^A = \left\{\theta_j^A, \theta_{j-1}^A, \dots ,\right\}$. We use $p(x, y \mid z)$ to refer to the joint pdf of x and y conditional on z, with the same symbol p used for all pdf's.

The usual Metropolis-Hastings argument is available, with all distributions conditioned on S_{j-2} to show that, under the assumption that $p(\theta_{j-1}^B \mid S_{j-2}^A) = \pi(\theta_{j-1}^B)$,

$$p(\theta_j^B \mid S_{j-2}^A) = \pi(\theta_j^B)\ . \tag{A.1}$$

But to make the fixed point argument, we need the stronger conclusion that

$$p(\theta_j^B \mid S_j^A) = \pi(\theta_j^B)\ . \tag{A.2}$$

The distribution from which θ_j^B is drawn depends on previous draws in either sequence only via θ_{j-1}^B and S_{j-2}. In particular it does not depend on θ_k^A values for $k \geq j-1$. Thus the joint distribution of $\theta_j^B, \theta_{j-1}^B, \theta_j^A, \theta_{j-1}^A \mid S_{j-2}$ has a pdf of the form

$$\begin{aligned} p(\theta_j^B, \theta_{j-1}^B, \theta_j^A, \theta_{j-1}^A \mid S_{j-2}^A) \\ &= p(\theta_j^B \mid \theta_{j-1}^B, S_j^A) \cdot p(\theta_{j-1}^B \mid S_j^A) \cdot p(\theta_j^A \mid S_{j-1}^A) \cdot p(\theta_{j-1}^A \mid S_{j-2}) \\ &= p(\theta_j^B \mid \theta_{j-1}^B, S_{j-2}^A) \cdot p(\theta_{j-1}^B \mid S_j^A) \cdot p(\theta_j^A \mid S_{j-1}^A) \cdot p(\theta_{j-1} \mid S_{j-2}^A)\ . \end{aligned} \tag{A.3}$$

Now assume that $p(\theta_{j-1}^B \mid S_j^A) = \pi(\theta_{j-1}^B)$. That there is no dependence of θ_{j-1}^B's conditional distribution on θ_j^A follows by construction, but that this distribution is totally independent of the A sequence is an hypothesis that we impose on the way to showing that it leads to a fixed point. Under this assumption we can rewrite the first and last line of (A.3) as

$$\begin{aligned} p(\theta_j^B, \theta_{j-1}^B, \theta_j^A, \theta_{j-1}^A \mid S_{j-2}^A) \\ &= p(\theta_j^B \mid \theta_{j-1}^B, S_{j-2}^A) \cdot \pi(\theta_{j-1}^B) \cdot p(\theta_j^A \mid S_{j-1}^A) \cdot p(\theta_{j-1} \mid S_{j-2}^A)\ . \end{aligned} \tag{A.4}$$

But if we integrate this expression with respect to θ_{j-1}^B, the only two terms in it that depend on θ_{j-1}^B are exactly those that the standard Metropolis-Hastings argument tells us integrate to something proportional to $\pi(\theta_j^B)$. This delivers the conclusion (A.2) we were aiming at. □

References

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin (1995): *Bayesian Data Analysis.* Chapman and Hall, London.