STATISTICALLY EFFICIENT THINNING OF A MARKOV CHAIN SAMPLER

By

Art B. Owen

Technical Report No. 2015-22
November 2015

Department of Statistics
STANFORD UNIVERSITY
Stanford, California 94305-4065

STATISTICALLY EFFICIENT THINNING OF A MARKOV CHAIN SAMPLER

By

Art B. Owen
Stanford University

Technical Report No. 2015-22
November 2015

This research was supported in part by National Science Foundation grants DMS 1407397 and DMS 1521145.

Department of Statistics
STANFORD UNIVERSITY
Stanford, California 94305-4065

http://statistics.stanford.edu

Statistically efficient thinning of a Markov chain sampler

Art B. Owen
Stanford University

October 2015

Abstract

It is common to subsample Markov chain samples to reduce the storage burden of the output. It is also well known that discarding $k-1$ out of every k observations will not improve statistical efficiency. It is less frequently remarked that subsampling a Markov chain allows one to omit some of the computation beyond that needed to simply advance the chain. When this reduced computation is accounted for, thinning the Markov chain by subsampling it can improve statistical efficiency. Given an autocorrelation parameter ρ and a cost ratio θ, this paper shows how to compute the most efficient subsampling frequency k. The optimal k grows rapidly as ρ increases towards 1. The resulting efficiency gain depends primarily on θ, not ρ. Taking $k = 1$ (no thinning) is optimal when $\rho \le 0$. For $\rho > 0$ it is optimal if and only if $\theta \le (1-\rho)^2/(2\rho)$. The efficiency gain never exceeds $1 + \theta$.

Keywords: Autoregression, Markov chain Monte Carlo, Subsampling

1 Introduction

It is common to thin a Markov chain sample, taking every k'th observation instead of all of them. Such subsampling is done to produce values that are more nearly independent. It also saves storage costs. It is well known that the average over a thinned sample set has greater variance than the plain average over all of the computed values (Geyer, 1992).

Most authors recommend against thinning, except where it is needed to reduce storage. MacEachern and Berliner (1994) go so far as to provide a 'justification for the ban against subsampling'. Link and Eaton (2011) write that "Thinning is often unnecessary and always inefficient".

One exception is Geyer (1991) who acknowledges that thinning can in fact increase statistical efficiency. Thinning makes each iteration cheaper which then makes it possible to run a thinned Markov chain longer than an unthinned one at the same computational cost. He gives some qualitative remarks about this

effect, but ultimately concludes that it is usually a negligible benefit because the autocorrelations in the Markov chain decay exponentially fast. Link and Eaton (2011) also acknowledge this possibility in their discussion as does Neal (1993, page 106).

This paper revisits the problem for an autoregressive correlation structure. That structure makes it possible to compute the exact optimal thinning factors. It is very often optimal to do no thinning. In other settings, optimality requires a very thin subsample. Sometimes that very thin subsample yields only a small efficiency, but at other times a very thin subsample yields a large improvement.

Section 2 defines asymptotic efficiency of thinning to every k'th observation when the samples have unit cost to generate, the function of interest costs $\theta > 0$ each time we compute it, and the observations have autocorrelation ρ^m at lag m. Section 3 presents some inequalities among the efficiency levels at different subsampling frequencies. Thinning never helps when the first order autocorrelation parameter ρ is negative. For $\rho > 0$ if any thinning level is to help, then taking every second sample must also help, and as a result we can get sharp expressions for the autocorrelation level at which thinning increases efficiency. In the limit $\rho \to \infty$ very large thinning factors become optimal but frequently much smaller factors are nearly as good. The efficiency gain does not exceed $1 + \theta$ for any ρ and k. Section 4 describes how to compute the optimal thinning factor k given the parameters θ and ρ. Section 5 has conclusions and discusses consequences of rejected proposals having essentially zero cost while accepted ones have a meaningfully large cost. An appendix has R code to compute the optimal k.

2 Asymptotic efficiency

To fix ideas, suppose that we generate a Markov chain x_t for $t \geq 1$. We have a starting value x_0 and then it costs one unit of computation to transition from x_{t-1} to x_t. Interest centers on the expected value of $y_t = f(x_t)$ for some function f. The cost to compute f is θ. It may be that $\theta \ll 1$ but it is also possible that θ is comparable to 1 or even larger. For instance it may be inexpensive to perform one update on a system of particles, but very expensive to find the new minimum distance among all those particles or some similar quantity of interest. When computation must pause to write $f(x_t)$ to storage then the cost of pausing is included in θ.

To focus on essentials, suppose that y_t are stationary with mean zero, unit variance and $\mathrm{cor}(y_t, y_{t+k}) = \rho^k$ for an autocorrelation $\rho \in (-1, 1)$. The most interesting cases have $\rho > 0$. We will sample $x_1, x_2, \ldots, x_N$ and retain every k'th observation $y_k = f(x_k)$, $y_{2k} = f(x_{2k})$ and so on. Our estimate of $\mu = \mathbb{E}(y_t)$ is

$$\hat{\mu}_k = \hat{\mu}_k(N) = \frac{1}{n}\sum_{i=1}^{n} f(x_{ki})$$

where $n = n_k = \lfloor N/k \rfloor$. We will get the same limits working with the not

necessarily integral $n_k = N/k$. Then

$$\lim_{N\to\infty} n_k \mathrm{var}(\hat{\mu}_k) = \frac{1+\rho^k}{1-\rho^k}.$$

Let the computing budget be B. Our cost is $N+n\theta = N(1+\theta/k)$. Therefore we can use $N_k = \lfloor B/(1+\theta/k) \rfloor$ which once again we replace by $B/(1+\theta/k)$. For this budget

$$n_k = \frac{N_k}{k} = \frac{B}{1+\theta/k}\frac{1}{k} = \frac{B}{k+\theta}.$$

The asymptotic efficiency of thinning at factor k relative to not thinning at all, that is using $k = 1$, is

$$\mathrm{eff}(k) = \mathrm{eff}(k;\rho,\theta) = \lim_{B\to\infty} \frac{\mathrm{var}(\hat{\mu}_1)}{\mathrm{var}(\hat{\mu}_k)} = \frac{1+\theta}{k+\theta}\frac{1+\rho}{1-\rho}\frac{1-\rho^k}{1+\rho^k}. \tag{1}$$

Table 1 shows $\arg\max_k \mathrm{eff}(k;\rho,\theta)$ for a range of correlations ρ and costs θ. As one would expect, the optimal thinning factor increases with both θ and ρ.

Perhaps surprisingly, the optimal thinning factor can be large even for $\theta \ll 1$, when the chain mixes slowly. For instance with $\theta = 0.01$ and $\rho = 0.9999$, the optimal thinning takes every 182'nd value. But Table 2 shows that in such cases only a small relative efficiency gain occurs. For $\theta = 0.01$ and $\rho = 0.9999$ the improvement is just under 1% and this gain may not be worth the trouble of using thinning.

When the cost θ is comparable to one, then thinning can bring a meaningful efficiency improvement for slow mixing chains. The efficiency gain approaches $\theta + 1$ in the limit as $\rho \to 1$. See equation (4) in Section 3.

A more efficient thinning rule allows the user to wait less time for an answer, or to attain a more accurate answer in the same amount of time. It may be a slight nuisance to incorporate thinning and when storage is not costly, we might even prefer to explore a larger set of sampled y values. Table 3 shows the least amount of thinning that we can do to get at least 95% efficiency relative to the most efficient value of k. That is, we find the smallest k with $\mathrm{eff}(k;\rho,\theta) \geq 0.95 \min_{\ell\geq 1} \mathrm{eff}(\ell;\rho,\theta)$. When 95% efficiency is adequate and θ is small then there is no need to thin. Theorem 2 below shows that there is no need to thin at any ρ, if efficiency $1/(1+\theta)$ is acceptable.

3 Some inequalities

Here we compare efficiencies for different choices of the thinning factor k. We find that thinning never helps when $\rho < 0$. In the limit as $\rho \to 1$ the optimal k diverges to infinity but we can attain nearly full efficiency by taking k to be a modest muliple of θ. When $\rho > 0$, the critical value of θ for which $\mathrm{eff}(k;\rho,\theta) > \mathrm{eff}(1;\rho,\theta)$ is an increasing function of $k \geq 2$. As a result we can determine when $k = 1$ is optimal. The following basic lemma underpins several of the results.

$\theta \setminus \rho$	0.1	0.5	0.9	0.99	0.999	0.9999	0.99999	0.999999
0.001	1	1	1	4	18	84	391	1817
0.01	1	1	2	8	39	182	843	3915
0.1	1	1	4	18	84	391	1817	8434
1	1	2	8	39	182	843	3915	18171
10	2	4	17	83	390	1816	8433	39148
100	3	7	32	172	833	3905	18161	84333
1000	4	10	51	327	1729	8337	39049	181612

Table 1: Optimal thinning factor k as a function of the relative cost θ of function evaluation and the first order autocorrelation ρ.

$\theta \setminus \rho$	0.1	0.5	0.9	0.99	0.999	0.9999	0.99999	0.999999
0.001	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
0.01	1.00	1.00	1.00	1.01	1.01	1.01	1.01	1.01
0.1	1.00	1.00	1.06	1.09	1.10	1.10	1.10	1.10
1	1.00	1.20	1.68	1.93	1.98	2.00	2.00	2.00
10	1.10	2.08	5.53	9.29	10.59	10.91	10.98	11.00
100	1.20	2.79	13.57	51.61	85.29	97.25	100.17	100.82
1000	1.22	2.97	17.93	139.29	512.38	845.38	963.79	992.79

Table 2: Asymptotic efficiency of the optimal thinning factor k from Table 1 as a function of θ and ρ. Values rounded to two places.

$\theta \setminus \rho$	0.1	0.5	0.9	0.99	0.999	0.9999	0.99999	0.999999
0.001	1	1	1	1	1	1	1	1
0.01	1	1	1	1	1	1	1	1
0.1	1	1	2	2	2	2	2	2
1	1	2	5	11	17	19	19	19
10	2	4	12	45	109	164	184	189
100	2	5	22	118	442	1085	1632	1835
1000	2	6	31	228	1182	4415	10846	16311

Table 3: Smallest k to give at least 95% of the efficiency of the most efficient k, as a function of θ and ρ.

Lemma 1. *Let $r > s \geq 1$ be integers, $\theta \geq 0$ and $-1 < \rho < 1$. Then* $\mathrm{eff}(r;\rho,\theta) > \mathrm{eff}(s;\rho,\theta)$ *if and only if*

$$2(\theta+s)(\rho^s-\rho^r) > (r-s)(1-\rho^s)(1+\rho^r). \tag{2}$$

Proof. Because $(1+\theta)(1+\rho)/(1-\rho) > 0$, the given inequality in efficiencies is equivalent to

$$(s+\theta)(1-\rho^r)(1+\rho^s) > (r+\theta)(1-\rho^s)(1+\rho^r).$$

Equation (2) follows by rearranging this inequality. □

It is obvious that thinning cannot improve efficiency when $\rho = 0$. Here we find that the same holds for $\rho < 0$.

Proposition 1. *If $\theta \geq 0$ and $-1 \leq \rho \leq 0$ then* $\mathrm{eff}(1;\rho,\theta) \geq \mathrm{eff}(k;\rho,\theta)$ *for all integers $k \geq 2$.*

Proof. Suppose to the contrary that $\mathrm{eff}(k) > \mathrm{eff}(1)$. Then Lemma 1 with $r = k$ and $s = 1$ yields

$$2(\theta+1)(\rho-\rho^k) > (k-1)(1-\rho)(1+\rho^k). \tag{3}$$

Because the right side of (3) is positive and the left side is not, we arrive at a contradiction, proving the result. □

With negative ρ there is an advantage to taking an odd integer k. For instance with Lemma 1 we find that $\mathrm{eff}(3;\rho,\theta) > \mathrm{eff}(2;\rho,\theta)$ when $\rho < 0$, but $k = 1$ remains the best odd integer. From here on we restrict attention to $\rho > 0$. Also it is obvious that $k = 1$ is best when $\theta = 0$ and so we assume $\theta > 0$.

Many applications have correlations very close to 1. Then

$$\mathrm{eff}(k;1,\theta) \equiv \lim_{\rho\to 1} \mathrm{eff}(k;\rho,\theta) = k\frac{1+\theta}{k+\theta}. \tag{4}$$

The optimal k grows without bound as $\rho \to 1$ and it has asymptotic efficiency $\theta + 1$. From Tables 1 and 2 we might anticipate that there are diminishing returns to very large k in this limit. If we do not insist on maximum efficiency we can use much smaller k. To obtain efficiency $1-\eta$ relative to the best k in the large ρ limit we impose

$$\mathrm{eff}(k;1,\theta) \geq (1+\theta)(1-\eta)$$

for $0 < \eta < 1$. Rearranging this inequality we obtain

$$k \geq \theta(1-\eta)/\eta.$$

For instance to attain 95% efficiency relative to the best k in the large ρ limit, we may take $\eta = 0.05$ and then $k = \lceil 19\theta \rceil$.

The next proposition introduces a critical cost $\theta_*(k)$ beyond which thinning by the factor k is more efficient than not thinning. That factor increases with k and the result is that we may then characterize when $k = 1$ (no thinning) is optimal.

Proposition 2. *Let $0 < \rho < 1$ and choose an integer $k \geq 2$. Then* $\mathrm{eff}(k;\rho,\theta) > \mathrm{eff}(1;\rho,\theta)$ *if and only if*

$$\theta > \theta_*(k,\rho) \equiv \frac{k-1}{2}\frac{(1-\rho)(1+\rho^k)}{\rho-\rho^k} - 1. \tag{5}$$

Proof. This follows from Lemma 1 using $r = k$ and $s = 1$. □

If k is very large and $0 < \rho < 1$ then ρ^k is negligible and

$$\theta_*(k,\rho) \doteq \frac{(k-1)(1-\rho)}{2\rho} - 1.$$

Proposition 3. *For $0 < \rho < 1$ the critical $\theta_*(k,\rho)$ from* (5) *is an increasing function of $k \geq 2$.*

Proof. Let $r = k - 1 \geq 1$ and put

$$\phi_*(r,\rho) = \frac{2(\theta_*+1)\rho}{1-\rho} = r\frac{1+\rho^{r+1}}{1-\rho^r}.$$

It suffices to show that ϕ_* is increasing in r. To this end,

$$\begin{aligned}\phi_*(r+1,\rho) - \phi_*(r,\rho) &= (r+1)\frac{1+\rho^{r+2}}{1-\rho^{r+1}} - r\frac{1+\rho^{r+1}}{1-\rho^r} \\ &= \frac{(r+1)(1+\rho^{r+2})(1-\rho^r) - r(1+\rho^{r+1})(1-\rho^{r+1})}{(1-\rho^{r+1})(1-\rho^r)}.\end{aligned} \tag{6}$$

The numerator in (6) is

$$\begin{aligned}&r[(1+\rho^{r+2})(1-\rho^r) - (1+\rho^{r+1})(1-\rho^{r+1})] + (1+\rho^{r+2})(1-\rho^r) \\ &= r[1-\rho^r+\rho^{r+2}] + (1+\rho^{r+2})(1-\rho^r)\end{aligned}$$

which is positive. Therefore $\phi_*(r+1,\rho) > \phi_*(r,\rho)$. □

Theorem 1. *For $0 < \rho < 1$, the choice $k = 1$ maximizes the efficiency* $\mathrm{eff}(k;\rho,\theta)$ *over integers $k \geq 1$ whenever*

$$\theta \leq \frac{(1-\rho)^2}{2\rho}. \tag{7}$$

For $\theta > 0$, the choice $k = 1$ maximizes efficiency if

$$\rho \leq 1 + \frac{\theta}{2} - \sqrt{\frac{\theta^2}{4} + \theta}. \tag{8}$$

Proof. From the monotonicity of θ_* in Proposition 3, if any $k > 1$ is better than $k = 1$ then $\mathrm{eff}(2;\rho,\theta) > \mathrm{eff}(1;\rho,\theta)$. Then $k = 1$ is most efficient if $\theta \leq \theta_*(2,\rho) = (1-\rho)(1+\rho^2)/(\rho-\rho^2) - 1 = (1-\rho)^2/(2\rho)$. The equation $\theta = (1-\rho)^2/(2\rho)$ has two roots ρ for fixed $\theta > 0$ and (7) holds for ρ between those roots. One of those roots is larger than 1 and the other is given in (8). □

Theorem 2. *For integer lag $k \geq 1$, cost $\theta >$ and autocorrelation $0 < \rho < 1$, the function* $\mathrm{eff}(k;\rho,\theta)$ *is nondecreasing in ρ, and so*

$$\mathrm{eff}(k;\rho,\theta) \leq \theta + 1.$$

Proof. The second conclusion follows from the first using the limit in (4).

The derivative of $\mathrm{eff}(k;\rho,\theta) \times (k+\theta)/(1+\theta)$ with respect to ρ is

$$\frac{(1-k\rho^{k-1}-(k+1)\rho^k)(1-\rho)(1+\rho^k)-(1+\rho)(1-\rho^k)(-1+k\rho^{k-1}-(k+1)\rho^k)}{(1-\rho)^2(1+\rho^k)^2}. \tag{9}$$

It suffices to show that the numerator in (9) is non-negative for $0 < \rho < 1$. That numerator simplifies to

$$2k\rho^{k+1} - 2k\rho^{k-1} - 2\rho^{2k} + 2.$$

It is zero at $\rho = 1$ and so it now suffices to show that the derivative of the numerator is negative for $0 < \rho < 1$. That derivative is

$$2k(k+1)\rho^k - 2k(k-1)\rho^{k-2} - 4k\rho^{2k-1} = 2k\rho^{k-2}(\rho-1)[(k+1)\rho + k - 1]$$

which is indeed negative as required. □

Next we show that $\mathrm{eff}(k;\rho,\theta)$ is unimodal in k for fixed $\rho \in (0,1)$ and $\theta > 0$.

Proposition 4. *Let $0 < \rho < 1$ and $k \in \mathbb{N}$. Then there is a critical value $\tilde{\theta}_*(k)$ such that* $\mathrm{eff}(k+1;\rho,\theta) > \mathrm{eff}(k;\rho,\theta)$ *if and only if $\theta > \tilde{\theta}_*$, and this critical value is an increasing function of k.*

Proof. From Lemma 1, using $r = k+1$ and $s = k$, we find that the critical value is

$$\tilde{\theta}_*(k) = \frac{(1-\rho^k)(1+\rho^{k+1})}{2\rho^k(1-\rho)} - k.$$

Now let $\eta_k = 2(1-\rho)\tilde{\theta}_k = [1 + (2k+1)(\rho^{k+1} - \rho^k) - \rho^{2k+1}]\rho^{-k}$ and write

$$\begin{aligned}
\rho^{k+1}(\eta_{k+1} - \eta_k) &= [1 + (2k+3)(\rho^{k+2} - \rho^{k+1}) - \rho^{2k+3}] \\
&\quad - \rho[1 + (2k+1)(\rho^{k+1} - \rho^k) - \rho^{2k+1}] \\
&= (1-\rho) + 2(\rho^{k+2} - \rho^{k+1}) + \rho^{2k+2} - \rho^{2k+3} \\
&= (1-\rho)(1-\rho^{k+1})^2.
\end{aligned}$$

Therefore $\eta_{k+1} > \eta_k$ and so $\tilde{\theta}_*(k+1) > \tilde{\theta}_*(k)$. □

From Proposition 4 we find that either $\{k \in \mathbb{N} \mid \tilde{\theta}_*(k) > \theta\}$ is empty or it takes the form $\{1, 2, \ldots, L\}$ for some $L \in \mathbb{N}$. In the first case, $k = 1$ is optimal and $\mathrm{eff}(k;\rho,\theta)$ is a nonincreasing function of k. In the second case, $\mathrm{eff}(k;\rho,\theta)$ is strictly increasing over $k = 1, 2, \ldots, k_* = L+1$ and nonincreasing thereafter. If we can find $r, s \in \mathbb{N}$ with $r > s$ and $\mathrm{eff}(r;\rho,\theta) < \mathrm{eff}(s;\rho,\theta)$ then we know that the optimal value k_* is in $\{1, 2, \ldots, r\}$.

4 Optimization

The most direct way to maximize $\mathrm{eff}(k;\rho,\theta)$ over $k \in \mathbb{N}$ is to compute $\mathrm{eff}(k;\rho,\theta)$ for all $k = 1,\dots,k_{\max}$ and then choose

$$k_* = k_*(\rho,\theta) = \arg\max_{1\le k\le k_{\max}} \mathrm{eff}(k;\rho,\theta).$$

It is necessary to find a value $k_{\max}$ that we can be sure is at least as large as k_*. In light of Proposition 4, we need only find a value $k_{\max}$ where $\mathrm{eff}(k_{\max};\rho,\theta) < \mathrm{eff}(k';\rho,\theta)$ holds for some $k' < k_{\max}$. We do this by repeatedly doubling k until we encounter a decreased efficiency.

For moderately large values of θ and $1/(1-\rho)$ it is numerically very stable to compute $\mathrm{eff}(k;\rho,\theta)$. But for more extreme cases it is better to work with

$$\mathrm{leff}(k) \equiv \log(\mathrm{eff}(k;\rho,\theta)) = c(\rho,\theta) - \log(k+\theta) + \log(1-\rho^k) - \log(1+\rho^k),$$

where $c(\rho,\theta) = \log[(1+\theta)(1+\rho)/(1-\rho)]$ does not depend on k. Many computing environments contain a special function $\mathrm{log1p}(x)$ that is a numerically more precise way to compute $\log(1+x)$ for small $|x|$. Ignoring c we then work with

$$\mathrm{leff}'(k) \equiv -\log(k+\theta) + \mathrm{log1p}(-\rho^k) - \mathrm{log1p}(\rho^k).$$

Now, to find $k_{\max}$ we set $m = 1$ and then while $\mathrm{leff}'(2m) > \mathrm{leff}'(m)$ set $m = 2m$. At convergence take $k_{\max} = 2m$. R Code to implement this optimization is given in the Appendix. Only in extreme circumstances will $k_{\max}$ be larger than one million, and so the enumerative approach will ordinarily have a trivial cost and so it will not be necessary to use more sophisticated searches. It takes about 1/6 of a second for this search to produce the values in Tables 1 and 2 on a MacBook Air. Relaxing k to noninteger values in $[1,\infty)$ one can see that $\log \mathrm{eff}(e^x;\rho,\theta)$ appears to be a concave function of $x \in [0,\infty)$.

5 Discussion

Thinning a Markov chain sample can improve statistical efficiency. The optimal subsampling rate grows rapidly as ρ increases towards 1 becoming unbounded in the limit. Sometimes those large subsampling rates correspond to only modest efficiency improvements. The magnitude of the improvement depends greatly on the ratio θ of the cost of function evaluation to the cost of updating the Markov chain. When θ is of order 1 or higher, a meaningful efficiency improvement can be attained by thinning the Markov chain.

In some problems, the cost θ may have an important dependence on k. In an MCMC, it is common to have $x_{t+1} = x_t$ because a proposal was rejected. In such cases $f(x_{t+1}) = f(x_t)$ need not be recomputed. Then an appropriate cost measure for θ would be the CPU time taken to evaluate f, normalized by the time to generate a proposal, and then multiplied by the acceptance rate. Larger values of k increase the chance that a proposal has been accepted and hence the

average cost of computing f. For instance, Gelman et al. (1996) find that an acceptance rate of $\alpha = 0.234$ is most efficient in high dimensional Metropolis random walk sampling. Then when thinning by factor k, the appropriate cost is $\theta(1-\alpha^k)$ where θ is the cost of an accepted proposal and the efficiency becomes

$$\frac{1+\theta(1-\alpha)}{k+\theta(1-\alpha^k)}\frac{1+\rho}{1-\rho}\frac{1-\rho^k}{1+\rho^k}.$$

Optimizing this problem is more difficult because the autocorrelation ρ is a function of the acceptance rate α. At any level of thinning, the optimal α may depend on θ.

Acknowledgments

This work was supported by the NSF under grants DMS-1407397 and DMS-1521145. I thank Hera He and Christian Robert for helpful comments.

References

Gelman, A., Roberts, G., and Gilks, W. (1996). Efficient metropolis jumping hules. In Bernardo, J. M., Berger, J. O., and Smith, A. F. M., editors, *Bayesian statistics 5*, pages 599–608.

Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood. In Keramides, E. M., editor, *Proceedings of the 23rd Symposium on the Interface*. Interface Foundation of North America.

Geyer, C. J. (1992). Practical Markov chain Monte Carlo. *Statistical Science*, 7(2):473–483.

Link, W. A. and Eaton, M. J. (2011). On thinning of chains in MCMC. *Methods in ecology and evolution*, 3(1):112–115.

MacEachern, S. N. and Berliner, L. M. (1994). Subsampling the Gibbs sampler. *The American Statistician*, 48(3):188–190.

Neal, R. M. (1993). Probabilistic inference using Markov chain Monte Carlo methods. Technical report, University of Toronto.

Appendix: R code

```
# Code to find the optimal amount of thinning for a Markov sample.
# It costs 1 unit to advance the chain, and theta units to evaluate
# the function. The autocorrelation is rho.

effk = function(k,theta,rho){
# Asymptotic efficiency of thinning factor k vs using k=1
# Compute and exponentiate log( effk )
# NB: log1p( x ) = log( 1+x )
t1 = log1p(theta) - log(k+theta)
t2 = log1p(rho) - log1p(-rho)
t3 = log1p(-rho^k) - log1p(rho^k)

exp( t1 + t2 + t3 )
}

leffkprime = function(k,theta,rho){
# Log of asymptotic efficiency at thinning factor k.
# It ignores terms that do not depend on k.

if( any( rho!=0 ) & any( abs(rho^k) == 0) ){
# Basic detection of underflow while still allowing rho=0
  badk = min( k[abs(rho^k)==0] )
  msg = paste("Underflow for k >=",badk,sep=" ")
  stop(msg)
}
- log(k+theta) + log1p(-rho^k) - log1p(rho^k)
}

getkmax = function(theta,rho){
# Find an upper bound for the optimal thinning fraction k
if( theta<0 )stop("Negative theta")
if( rho<0 )stop("Negative rho")
if( rho >=1 )stop("rho too close to one")

m=1
while( leffkprime(2*m,theta,rho) > leffkprime(m,theta,rho) )
  m = m*2

2*m
}
```

```
kopt = function(theta,rho,klimit=10^7){
# Find optimal k for the given theta and rho.
# Stop if kmax is too large. That usually
# means that theta is very large or rho is very nearly one

kmax = getkmax(theta,rho)
if( kmax > klimit ){
  msg = paste("Optimal k too expensive. It requires checking",kmax,"values.")
  stop(msg)
}
leffvals = leffkprime( 1:kmax,theta,rho )
best = which.max(leffvals)

best
}

kok = function(theta,rho,klimit=10^7,eta=.05){
# Find near optimal k for the given theta and rho.
# This is the smallest k with efficiency >= 1-eta times best.
# NB: computations in kopt are repeated rather than
# saved. This is inefficient but the effect is minor.

best = kopt(theta,rho,klimit)
leffvals = leffkprime( 1:best,theta,rho )
ok = min( which(leffvals >= leffvals[best]  + log1p(-eta) ) )
ok
}

kopttable = function( thvals = 10^c(-3:3), rhovals = c(.1,.5,1-10^-c(1:6)),eta=.05){

# Prepare tables of optimal k, its efficiency, and smallest
# k with at least 1-eta efficiency

T = length(thvals)
R = length(rhovals)

bestk = matrix(0,T,R)
row.names(bestk) = thvals
colnames(bestk) = rhovals
effbk = bestk
okk = bestk

for( i in 1:T )
for( j in 1:R ){
  theta = thvals[i]
  rho   = rhovals[j]
```

```
  bestk[i,j] = kopt(theta,rho)
  effbk[i,j] = leffkprime(bestk[i,j],theta,rho)-leffkprime(1,theta,rho)
  effbk[i,j] = exp(effbk[i,j])
  okk[i,j]   = kok(theta,rho,eta=eta)
}

list( bestk=bestk, effbk=effbk, okk=okk )
}
```