Statistically efficient thinning of a Markov chain sampler

Art B. Owen*
Department of Statistics
Stanford University

This version: April 2017

Abstract

It is common to subsample Markov chain output to reduce the storage burden. Geyer (1992) shows that discarding $k-1$ out of every k observations will not improve statistical efficiency, as quantified through variance in a given computational budget. That observation is often taken to mean that thinning MCMC output cannot improve statistical efficiency. Here we suppose that it costs one unit of time to advance a Markov chain and then $\theta > 0$ units of time to compute a sampled quantity of interest. For a thinned process, that cost θ is incurred less often, so it can be advanced through more stages. Here we provide examples to show that thinning will improve statistical efficiency if θ is large and the sample autocorrelations decay slowly enough. If the lag $\ell \geqslant 1$ autocorrelations of a scalar measurement satisfy $\rho_\ell \geqslant \rho_{\ell+1} \geqslant 0$, then there is always a $\theta < \infty$ at which thinning becomes more efficient for averages of that scalar. Many sample autocorrelation functions resemble first order AR(1) processes with $\rho_\ell = \rho^{|\ell|}$ for some $-1 < \rho < 1$. For an AR(1) process it is possible 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 $\rho \leqslant 0$. For $\rho > 0$ it is optimal if and only if $\theta \leqslant (1-\rho)^2/(2\rho)$. This efficiency gain never exceeds $1 + \theta$. This paper also gives efficiency bounds for autocorrelations bounded between those of two AR(1) processes.

Keywords: Autoregression, Markov chain Monte Carlo, Subsampling

*This work was supported by NSF grants *DMS-1407397 and DMS-1521145.*

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". In discussing thinning of the Gibbs sampler, Gamerman and Lopes (2006) say: "There is no gain in efficiency, however, by this approach and estimation is shown below to be always less precise than retaining all chain values."

One exception is Geyer (1991) who acknowledges that thinning can in fact increase statistical efficiency. Thinning reduces the average cost of iterations 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 thinning problem and shows that the usual advice against thinning can be misleading, by quantifying the argument of Geyer (1991) described above. The key variables are the cost of computing the quantity of interest (after advancing the Markov chain) and the speed at which correlations in the quantity of interest decay. When the cost is expensive and the decay is slow, then thinning can improve efficiency by a large factor.

We suppose that it costs one unit to advance the Markov chain and $\theta > 0$ units each time the quantity of interest is computed. If lag ℓ autocorrelations satisfy $\rho_1 \geqslant \rho_2 \geqslant \cdots > 0$, then there is always a θ for which thinning by a factor of k will improve efficiency.

For a first order autoregressive autocorrelation structure in the quantity of interest, very precise results are possible. Given the update costs and the autocorrelation parameter we can compute the optimal thinning factor as well as the efficiency improvement with that factor. The autoregressive assumption is very convenient because it reduces the dependence problem to just one scalar parameter. Also, real-world autocorrelations commonly resemble those of an AR(1) model. In the social sciences, the book by Jackman (2009) shows many sample autocorrelation functions that resemble AR(1). The physicists Newman and Barkema (1999) writing about the Ising model state that "the autocorrelation is expected to fall off exponentially at long times" (p 60). Geyer (1991) notes an exponential upper bound for autocorrelations when processes are ρ-mixing.

Sometimes thinning is built in to standard simulation practice. For instance an Ising model may be simulated as a sequence of 'passes' with each pixel being examined on average once per pass. The state of the Markov chain might only be inspected once per pass. That represents a substantial, though not necessarily optimal amount of thinning. It might really be better to sample several times per pass or just once every k passes.

An outline of this paper is as follows. 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. If the autocorrelations ρ_ℓ for $\ell \geqslant 1$ are nonnegative and nonincreasing and $\rho_k > 0$ then there is always some finite $\theta > 0$ for which thinning by a factor of k is more efficient than not thinning. Much sharper results can be obtained when the autocorrelations take the form $\rho_\ell = \rho^\ell$ at lag ℓ. In many cases the optimal thinning factor k is greater than one.

Section 3 presents some inequalities among the efficiency levels at different subsampling frequencies in the AR(1) case. Thinning never helps when $\rho \leqslant 0$. 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 $\rho \to 1$ 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 considers autocorrelations that are bounded between two autoregressive forms $\underline{\rho}^\ell \leqslant \rho_\ell \leqslant \bar{\rho}^\ell$. The range of optimal thinning factors widens, but it is often possible to find meaningful efficiency improvements from thinning. Section 5 describes how to compute the optimal thinning factor k given the parameters θ and an autoregression parameter ρ. Section 6 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.

We close with some practical remarks. When thinning benefits, it does not appear to be critical to find the optimal factor k. Instead there are many near optimal thinning factors. If the autocorrelations decay slowly and the cost θ is large then a suggestion of Hans Anderson is to thin in such a way that about half of the cost is spent advancing the Markov chain and about half is spent computing the quantity of interest. That should be nearly as efficient as using the optimal k.

2 Asymptotic efficiency

To fix ideas, suppose that we generate a Markov chain x_t for $t \geqslant 1$. We have a starting value x_0 and then it costs one unit of computation to transition from x_{t-1} to x_t. The state space for x_t can be completely general in the analysis below.

Interest centers on the expected value of $y_t = f(x_t)$ for some real-valued function f. There is ordinarily more than one such function, but here we focus on a single one. The cost to compute f is θ. Often $\theta \ll 1$ 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. Or, it may be very inexpensive to flip one or more edges in a simulated network but expensive to compute a connectivity property of the resulting network. Finally, when computation must pause to store $f(x_t)$, then the cost of pausing is included in θ.

The efficiency of thinning derived here depends on the cost of computing y_t from x_t, the cost of transition from x_t to x_{t+1}, and the autocovariances of the series y_t. We assume that y_t is stationary: any necessary warmup has taken place.

The variance of $\sqrt{n}\hat{\mu} \equiv (1/\sqrt{n})\sum_{i=1}^n f(x_i)$ is asymptotically $\sigma^2(1 + 2\sum_{\ell=1}^\infty \rho_\ell)$ where

$\rho_\ell = \mathrm{cor}(y_i, y_{i+\ell})$ and $\sigma^2 = \mathrm{var}(y_i)$. We assume that $0 < \sigma^2 < \infty$. Now suppose that we thin the chain as follows. We compute $y_i = f(x_i)$ only for every k'th observation. The number of function values we get will depend on k. If we take n_k of them then we estimate μ by

$$\hat{\mu}_k = \frac{1}{n_k}\sum_{i=1}^{n_k} f(x_{ik}).$$

To compute $\hat{\mu}_k$ we must advance the chain kn_k times and evaluate f at each of n_k points for a total cost of $n_k(k+\theta)$. When our computational budget is a cost of $B > 0$, then we will use the largest n_k with $n_k(k+\theta) \leqslant B$. That is $n_k = \lfloor B/(k+\theta)\rfloor$.

The relative efficiency of thinning by a factor k compared to not thinning at all is

$$\mathrm{eff}_B(k) = \frac{(\sigma^2/n_1)(1+2\sum_{\ell=1}^\infty \rho_\ell)}{(\sigma^2/n_k)(1+2\sum_{\ell=1}^\infty \rho_{k\ell})} = \frac{\lfloor B/(k+\theta)\rfloor}{\lfloor B/(1+\theta)\rfloor}\frac{1+2\sum_{\ell=1}^\infty \rho_\ell}{1+2\sum_{\ell=1}^\infty \rho_{k\ell}}.$$

The dependence on B is minor and is a nuisance. We work instead with

$$\mathrm{eff}(k) = \frac{1+\theta}{k+\theta}\frac{1+2\sum_{\ell=1}^\infty \rho_\ell}{1+2\sum_{\ell=1}^\infty \rho_{k\ell}}, \tag{1}$$

which is also the limit of $\mathrm{eff}_B(k)$ as $B \to \infty$.

2.1 Generic autocorrelations

The efficiency of thinning depends on the autocorrelations ρ_ℓ only through certain sums of them. We can use this to get inequalities on autocorrelations that are equivalent to statements on the efficiency $\mathrm{eff}(k)$. Then under a monotonocity constraint on autocorrelations we can get a condition that ensures that thinning will help.

Lemma 1. *Let $R = \sum_{\ell=1}^\infty \rho_\ell$, and for a thinning factor $k \geqslant 1$, define $R_k = \sum_{\ell=1}^\infty \rho_{k\ell}$ and $R_{-k} = R - R_k$. Then $\mathrm{eff}(k) < 1$ if and only if*

$$R_{-k} < \frac{k-1}{\theta+1}\big(R_k + 1/2\big). \tag{2}$$

Proof. We rewrite (1) as

$$\mathrm{eff}(k) = \frac{1+\theta}{k+\theta}\frac{1+2R_k+2R_{-k}}{1+2R_k}.$$

Then $\mathrm{eff}(k) < 1$ if and only if $1+2R_k+2R_{-k} < (1+2R_k)(k+\theta)/(1+\theta)$, which can be rearranged into (2). □

Only one out of every k consecutive autocorrelations contributes to R_k while the other $k-1$ of them contribute to R_{-k}. If we let $\bar{R}_{-k} = R_{-k}/(k-1)$, then equation (2) becomes $\bar{R}_{-k} < (R_k + 1/2)/(\theta+1)$. For a Markov chain with slowly converging autocorrelations we will have $R_k \gg 1/2$. Then for thinning to be inefficient, the autocorrelations contributing to R_k have to be enough larger than the others to overcome the factor $\theta+1$. When θ is

large we would then need every k'th autocorrelation to be surprisingly large compared to the nearby ones, in order to make thinning inefficient.

Now suppose that the autocorrelations satisfy

$$\rho_1 \geqslant \rho_2 \geqslant \cdots \geqslant 0. \tag{3}$$

This quite mild sufficient condition rules out a setting where every k'th autocorrelation is unusually large compared to its $k-1$ predecessors.

Theorem 1. *If* (3) *holds then* $\mathrm{eff}(k) < 1$ *can only hold for* $\theta < 1/(2R_k)$.

Proof. From Lemma 1, $\mathrm{eff}(k) < 1$ implies that $R_{-k} < (R_k + 1/2)(k-1)/(\theta+1)$. If (3) holds then $(k-1)R_k \leqslant R_{-k}$. Therefore $(k-1)R_k < (R_k + 1/2)(k-1)/(\theta+1)$ which can be rearranged to complete the proof. □

If ρ_ℓ are large and slowly decreasing, then R_k will be quite large and $1/(2R_k)$ will be very small. Then even for mild costs θ, Theorem 1 ensures that some form of thinning will improve asymptotic efficiency. The converse does not hold: thinning might still help, even if $\theta < 1/(2R_k)$.

The condition (3) includes the case with $\rho_\ell = 0$ for all $\ell > 1$. This is a case where thinning cannot help. We also get $R_k = 0$ here, so Theorem 1 then places no constraint on θ, consistent with the fact that thinning cannot then help. If (3) holds, then all we need is $\rho_k > 0$ to get $R_k > 0$. Then there is a $\theta < \infty$ for which $\mathrm{eff}(k) > 1$ holds.

2.2 AR(1) autocorrelations

Here we consider a first order autoregressive model, $\rho_k = \rho^k$ for $\rho \in (-1, 1)$ and $k \in \mathbb{N}$. In this setting it is possible to find the most efficient values of k and to measure the efficiency gain from them. It is reasonable to expect qualitatively similar results from autocorrelations that have approximately the AR(1) form. Some steps in that direction are in Section 4.

Under an AR(1) model

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

We use $\mathrm{effar}(k)$ to denote an efficiency computed under the autoregressive assumption and $\mathrm{eff}(k)$ to denote a more general efficiency. The efficiency in (1) is a continuous function of the underlying ρ_ℓ inside of it, so small departures from the autoregressive assumption will make small changes in efficiency. When the peak of $\mathrm{eff}(k)$ is flat then small changes in ρ_ℓ may bring large changes in $\arg\max_k \mathrm{eff}(k)$.

Table 1 shows $\arg\max_k \mathrm{effar}(k; \rho, \theta)$ for a range of correlations ρ and costs θ. This k is computed via a search described in Section 5. 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 (7) 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 $\mathrm{effar}(k;\rho,\theta) \geqslant 0.95 \min_{\ell \geqslant 1} \mathrm{effar}(\ell;\rho,\theta)$. When 95% efficiency is adequate and θ is small then there is no need to thin. Theorem 3 below shows that in the AR(1) model, 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, under the autoregressive assumption $\rho_\ell = \rho^\ell$. 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 multiple of θ. When $\rho > 0$, the critical value of θ, meaning one large enough to make $\mathrm{effar}(k;\rho,\theta) > \mathrm{effar}(1;\rho,\theta)$, is an increasing function of $k \geqslant 2$. As a result we can determine when $k = 1$ is optimal. The following basic lemma underpins several of the results.

Lemma 2. *Let $r > s \geqslant 1$ be integers, $\theta \geqslant 0$ and $-1 < \rho < 1$. Then* $\mathrm{effar}(r;\rho,\theta) > \mathrm{effar}(s;\rho,\theta)$ *if and only if*

$$2(\theta + s)(\rho^s - \rho^r) > (r - s)(1 - \rho^s)(1 + \rho^r). \tag{5}$$

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 (5) 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 $\theta \geqslant 0$ and $-1 < \rho \leqslant 0$ then* $\mathrm{effar}(1;\rho,\theta) \geqslant \mathrm{effar}(k;\rho,\theta)$ *for all integers $k \geqslant 2$.*

Proof. Suppose to the contrary that $\mathrm{effar}(k) > \mathrm{effar}(1)$. Then Lemma 2 with $r = k$ and $s = 1$ yields

$$2(\theta + 1)(\rho - \rho^k) > (k - 1)(1 - \rho)(1 + \rho^k). \tag{6}$$

Because the right side of (6) is positive and the left side is not, we arrive at a contradiction, proving the result. □

$\theta \setminus \rho$	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 autoregressive parameter 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 the autoregression parameter ρ.

With negative ρ there is an advantage to taking an odd integer k compared to nearby even integers, stemming from the factor $(1+\rho^k)/(1-\rho^k)$ in effar(k). For instance with Lemma 2 we find that effar$(3;\rho,\theta) >$ effar$(2;\rho,\theta)$ 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

$$\text{effar}(k;1,\theta) \equiv \lim_{\rho\to 1} \text{effar}(k;\rho,\theta) = k\frac{1+\theta}{k+\theta}. \tag{7}$$

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 at least $1-\eta$ relative to the best k in the large ρ limit we impose

$$\text{effar}(k;1,\theta) \geqslant (1+\theta)(1-\eta)$$

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

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

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 threshold cost 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 $k \geqslant 2$. Then* effar$(k;\rho,\theta) >$ effar$(1;\rho,\theta)$ *if and only if*

$$\theta > \theta_*(k,\rho) \equiv \frac{k-1}{2}\frac{(1-\rho)(1+\rho^k)}{\rho-\rho^k} - 1. \tag{8}$$

Proof. This follows from Lemma 2 using $r = k$ and $s = 1$. □

For fixed $\rho \in (0,1)$ and very large k we find that

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

If ρ is near zero, then choosing large k is only efficient for very large θ. If ρ is close to 1 then the threshold for k's efficiency can be quite low.

Proposition 3. *For $0 < \rho < 1$ the critical $\theta_*(k,\rho)$ from* (8) *is an increasing function of $k \geqslant 2$.*

Proof. Let $r = k - 1 \geqslant 1$ 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 over $r \in [1, \infty)$. Differentiating,

$$\frac{\mathrm{d}\phi_*}{\mathrm{d}r} = \frac{(1 + \rho^{r+1} + r\rho^{r+1}\log(\rho))(1 - \rho^r) - r(1 + \rho^{r+1})(-\rho^r \log(\rho))}{(1 - \rho^r)^2}$$

and we need only show that the numerator

$$\eta(\rho, r) = 1 - \rho^r + \rho^{r+1} - \rho^{2r+1} + (\rho^r + \rho^{r+1})\log(\rho^r)$$

is positive. We will show that $\eta(\rho, r) \geqslant \eta(\rho, 1) \geqslant 0$.

First we show that $\eta(\rho) \equiv \eta(\rho, 1) \geqslant 0$. This function and its first three derivatives are

$$\begin{aligned}
\eta(\rho) &= 1 - \rho + \rho^2 - \rho^3 + (\rho + \rho^2)\log(\rho), \\
\eta'(\rho) &= 3\rho - 3\rho^2 + (1 + 2\rho)\log(\rho), \\
\eta''(\rho) &= 5 - 6\rho + \rho^{-1} + 2\log(\rho), \quad \text{and} \\
\eta'''(\rho) &= -6 - \rho^{-2} + 2\rho^{-1} \\
&= -5 - (1 - \rho^{-1})^2.
\end{aligned}$$

Because $\eta''' \leqslant 0$ and $\eta''(1-) = 0$ we find that $\eta''(\rho) \geqslant 0$. Similarly, $\eta'(1-) = 0$ and so $\eta'(\rho) \leqslant 0$. Finally, $\eta(1-) = 0$ so that $\eta(\rho) = \eta(\rho, 1) \geqslant 0$, completing the first step.

For the second step, treating $r \geqslant 1$ as a continuous variable,

$$\begin{aligned}
\frac{\partial}{\partial r}\eta(\rho, r) &= -\rho^r \log(\rho) + \rho^{r+1}\log(\rho) - 2\rho^{2r+1}\log(\rho) \\
&\quad + (\rho^r + \rho^{r+1})\log(\rho) + r(\rho^r + \rho^{r+1})\log^2(\rho) \\
&= 2(\rho^{r+1} - \rho^{2r+1})\log(\rho) + r(\rho^r + \rho^{r+1})\log^2(\rho) \\
&= \rho^r \log(\rho)\big(2\rho - 2\rho^{r+1} + r\log(\rho) + r\rho\log(\rho)\big).
\end{aligned}$$

We now show that this partial derivative is nonnegative. Because $\rho^r \log(\rho) \leqslant 0$, it suffices to show that the second factor $F(\rho, r) \leqslant 0$ where $F(\rho, r) \equiv 2\rho - 2\rho^{r+1} + r\log(\rho) + r\rho\log(\rho)$. Differentiating yields

$$\begin{aligned}
\frac{\partial}{\partial \rho}F(\rho, r) &= 2 - 2(r + 1)\rho^r + r\rho^{-1} + r\log(\rho) + r, \quad \text{and} \\
\frac{\partial^2}{\partial \rho^2}F(\rho, r) &= -2r(r + 1)\rho^{r-1} + r\rho^{-1}(1 - \rho^{-1}) \leqslant 0.
\end{aligned}$$

Proceeding as before, $(\partial/\partial\rho)F(1-, r) = 0$ so this first partial derivative is nonnegative. Then $F(1-, r) = 0$ so F is nonpositive as required. □

Theorem 2. *For $0 < \rho < 1$, the choice $k = 1$ maximizes efficiency* $\mathrm{effar}(k;\rho,\theta)$ *over integers $k \geqslant 1$ whenever*

$$\theta \leqslant \frac{(1-\rho)^2}{2\rho}. \tag{9}$$

For $\theta > 0$, the choice $k = 1$ maximizes efficiency $\mathrm{effar}(k;\rho,\theta)$ *if*

$$\rho \leqslant 1 + \theta - \sqrt{\theta^2 + 2\theta}. \tag{10}$$

Proof. From the monotonicity of θ_* in Proposition 3, if any $k > 1$ is better than $k = 1$ then $\mathrm{effar}(2;\rho,\theta) > \mathrm{effar}(1;\rho,\theta)$. Then $k = 1$ is most efficient if

$$\theta \leqslant \theta_*(2,\rho) = \frac{(1-\rho)(1+\rho^2)}{2(\rho-\rho^2)} - 1 = \frac{(1-\rho)^2}{2\rho},$$

establishing (9). The equation $\theta = (1-\rho)^2/(2\rho)$ has two roots ρ for fixed $\theta > 0$ and (9) holds for ρ outside the open interval formed by those two roots. One of those roots is larger than 1 and the other is given as the right hand side of (10). □

The upper limit in (10) is asymptotically $1/(2\theta)$ for large θ. That is $\lim_{\theta\to\infty} \theta(1+\theta-\sqrt{\theta^2+2\theta}) = 1/2$.

Theorem 3. *For integer lag $k \geqslant 1$, cost $\theta > 0$ and autocorrelation $0 < \rho < 1$, the function* $\mathrm{effar}(k;\rho,\theta)$ *is nondecreasing in ρ, and so*

$$\mathrm{effar}(k;\rho,\theta) \leqslant \theta + 1.$$

Proof. The second conclusion follows from the first using the limit in (7). The derivative of $\mathrm{effar}(k;\rho,\theta) \times (k+\theta)/(1+\theta)$ with respect to ρ is

$$\frac{(1-k\rho^{k-1}-(k+1)\rho^k)(1-\rho)(1+\rho^k) - (1+\rho)(1-\rho^k)(-1+k\rho^{k-1}-(k+1)\rho^k)}{(1-\rho)^2(1+\rho^k)^2}. \tag{11}$$

It suffices to show that the numerator in (11) is non-negative for $0 < \rho < 1$. The numerator simplifies to twice $N(\rho,k) = k\rho^{k+1} - k\rho^{k-1} - \rho^{2k} + 1$. Now

$$\frac{\partial}{\partial\rho} N(\rho,k) = k(k+1)\rho^k - k(k-1)\rho^{k-2} - 2k\rho^{2k-1} = k\rho^{k-2}F(\rho,k)$$

for a factor $F(\rho,k) = (k+1)\rho^2 - (k-1) - 2\rho^{k+2}$. Because $F(1-,k) = 0$ and $\partial F(\rho,k)/\partial\rho = 2(k+1)(\rho-\rho^k) \geqslant 0$ we have $F(\rho,k) \leqslant 0$. Therefore $\partial N(\rho,k)/\partial\rho \leqslant 0$ and because $N(1-,k) = 0$ we conclude that $N(\rho,k) \geqslant 0$ and so $\mathrm{effar}(k;\rho,\theta)$ is nondecreasing in ρ. □

Next we consider how to locate the maximizer over k of $\mathrm{effar}(k;\rho,\theta)$. The next result on relaxing $\log(k)$ to be a nonnegative real value helps.

Proposition 4. *For $\theta > 0$ and $\rho \in (0,1)$ the function* $\log(\mathrm{effar}(e^x;\rho,\theta))$ *is strictly concave in $x \geqslant 0$.*

Proof. We write $\log(\mathrm{effar}(e^x;\rho,\theta)) = g(x) + f(x)$ for

$$g(x) = \log\Big(\frac{1+\theta}{e^x+\theta}\Big) + \log\Big(\frac{1+\rho}{1-\rho}\Big), \quad \text{and} \quad f(x) = \log\Big(\frac{1-\rho^{e^x}}{1+\rho^{e^x}}\Big).$$

Now $g''(x) = -\theta e^x/(e^x+\theta)^2$, so g is strictly concave for $\theta > 0$. It now suffices to show that f is concave. Let $h = h(x) = \rho^{e^x}$. Then

$$f'(x) = \frac{-h'}{1-h} - \frac{h'}{1+h} = \frac{-2h'}{1-h^2} = \frac{-2h\log(h)}{1-h^2},$$

using $h' = h\log(h)$, and so

$$-\frac{1}{2}f''(x) = \frac{(h'\log(h)+h')(1-h^2) - h\log(h)(-2hh')}{(1-h^2)^2}. \tag{12}$$

The numerator in (12) is $h\log(h)\big(1-h^2+\log(h)+h^2\log(h)\big)$. We need only show that $t(z) = 1 - z^2 + \log(z) + z^2\log(z) \leqslant 0$ on $(0,1]$ which includes all relevant values of h. The result follows because $t(1) = t'(1) = t''(1) = 0$ and $t'''(z) = 2z^{-3} + 2z^{-1} \geqslant 0$. □

From Proposition 4, we know that if we relax k to real values in $[1,\infty)$, then $\mathrm{effar}(k;\rho,\theta)$ is either nonincreasing as k increases from 1, or it increases to a peak before descending, or it increases indefinitely as k increases. Because $\lim_{k\to\infty}\mathrm{effar}(k;\rho,\theta) = 0$ we can rule out the third possibility.

As a result, we know that if $\mathrm{effar}(k';\rho,\theta) < \mathrm{effar}(k;\rho,\theta)$ for $k' > k$, then the optimal value of k is in the set $\{1,2,\dots,k'-1\}$. Neither the value k' nor any larger value can be optimal because the function $\mathrm{effar}(k;\rho,\theta)$ is already in its descending region by the time we consider lags as large as k'.

4 Approximately autoregressive dependence

In this section we consider how efficiencies and optimal thinning factors behave when the autocorrelations are nearly but not exactly of autoregressive form. Recall that the efficiency of thinning to every k'th observation versus not thinning ($k = 1$) is given by $\mathrm{eff}(k)$ of (1). More generally, the efficiency of thinning by factor $r \in \mathbb{N}$ versus thinning by factor $s \in \mathbb{N}$ is

$$\mathrm{eff}(r,s) = \frac{\mathrm{eff}(r)}{\mathrm{eff}(s)} = \frac{s+\theta}{r+\theta}\frac{1+2\sum_{\ell=1}^{\infty}\rho_{s\ell}}{1+2\sum_{\ell=1}^{\infty}\rho_{r\ell}}.$$

Thinning should help for autocorrelations that are approximately autoregressive and decay very slowly. Then the numerator in $\mathrm{eff}(k,1)$ will be large. For there to be a meaningful efficiency gain, the denominator in $\mathrm{eff}(k,1)$ should not be too large. That is reasonable as we ordinarily expect that $\rho_{k\ell} < \rho_\ell$. We suppose now that the autocorrelations of y_i satisfy

$$\underline{\rho}^{\ell} \leqslant \rho_\ell \leqslant \bar{\rho}^{\ell}. \tag{13}$$

Then ρ_ℓ need not follow exactly the autoregressive form and indeed they need not be monotonically decreasing in ℓ.

Under condition (13) we can get upper and lower bounds on the summed autocorrelations and these yield

$$\frac{s+\theta}{r+\theta}\frac{1+\underline{\rho}^s}{1-\underline{\rho}^s}\Big/\frac{1+\bar{\rho}^r}{1-\bar{\rho}^r} \leqslant \mathrm{eff}(r,s) \leqslant \frac{s+\theta}{r+\theta}\frac{1+\bar{\rho}^s}{1-\bar{\rho}^s}\Big/\frac{1+\underline{\rho}^r}{1-\underline{\rho}^r} \equiv U_{rs}.$$

Any value r for which $U_{rs} < 1$ holds for some $s \neq r$ cannot be the optimal thinning factor.

There are two main ways that thinning can help. One is that the autocorrelations decay slowly. The other is that the cost θ to compute $y_i = f(x_i)$ is large. We consider one example of each type.

First, consider a slow but not extremely slow correlation decay, $\underline{\rho} = 0.98$ and $\bar{\rho} = 0.99$ with moderately large $\theta = 10$. If $U_{1k} < 1$, then thinning at factor k must be more efficient than not thinning for any autocorrelation satisfying (13). We find that this holds whenever $3 \leqslant k \leqslant 1078$. If $U_{1k} < 1/2$, then thinning at factor k must be at least twice as efficient as not thinning. This holds for $6 \leqslant k \leqslant 529$. If $28 \leqslant k \leqslant 195$ then thinning is at least four times as efficient as not thinning. In this not very extreme example, there are gains from thinning and they hold over a wide range of thinning factors $k > 1$. The thinning factors that are not dominated by some other thinning factor are given by $8 \leqslant k \leqslant 220$. Any other k cannot be optimal. The given values of $\underline{\rho}$, $\bar{\rho}$ and θ allow for a large set of possible optimal k, but they do not allow for $k = 1$ to be optimal. Instead, $k = 1$ is suboptimal by at least four-fold.

As a second example, consider a high cost $\theta = 100$ with moderately slow correlation decay given by $\underline{\rho} = 0.9$ and $\bar{\rho} = 0.95$. Then there is at least a 10-fold efficiency gain for any $34 \leqslant k \leqslant 87$ and the optimal k must satisfy $16 \leqslant k \leqslant 74$.

5 Optimization

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

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

It is necessary to find a value $k_{\max}$ that we can be sure is at least as large as k_*. In light of the discussion following the log concavity Proposition 4, we need only find a value $k_{\max}$ where $\mathrm{effar}(k_{\max};\rho,\theta) < \mathrm{effar}(k';\rho,\theta)$ 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 $\mathrm{effar}(k;\rho,\theta)$. But for more extreme cases it is better to work with

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

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

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

Now, to find $k_{\max}$ we set $m = 1$ and then while $\text{leffar}'(2m) > \text{leffar}'(m)$ 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 it will not then 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. If $k_{\max}$ is thought to be extraordinarily large then one could run a safeguarded Newton method to find $x_* = \arg\max_x \log(\text{effar}(e^x; \rho, \theta))$ and whichever of $k = \lceil e^{x_*} \rceil$ or $k = \lfloor e^{x_*} \rfloor$ maximizes $\text{effar}(k; \rho, \theta)$.

6 Discussion

Contrary to common recommendations, thinning a Markov chain sample can improve statistical efficiency. This phenomenon always holds for monotonically decreasing nonnegative autocorrelations if the cost of evaluating f is large enough. When the correlations follow an autoregressive model, 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 such a Markov chain. When the autocorrelations decay slowly but do not necessarily follow the exact autoregression pattern we may still find that thinning brings a large efficiency gain.

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

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

under an autoregressive assumption. Optimizing this case is outside the scope of this article. It is more difficult because the autocorrelation ρ depends on the acceptance rate α. At any level of thinning, the optimal α may depend on θ.

It is also common that one has multiple functions $f_1, \ldots, f_M$ to evaluate. They might each have different optimal thinning ratios. Optimizing the efficiency over such a collection raises issues that are outside the scope of this article. For instance, the cost of evaluating a subset of these functions may be subadditive in the costs of evaluating them individually due to shared computations. The importance of estimating those M different means may also be unequal. Finally, there may be greater statistical efficiency for comparisons of those corresponding means when the f_j are evaluated on common inputs.

Acknowledgments

This work was supported by the NSF under grants DMS-1407397 and DMS-1521145. I thank Hera He, Christian Robert, Hans Andersen, Michael Giles and some anonymous reviewers for helpful comments.

References

Gamerman, D. and Lopes, H. F. (2006). *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. CRC Press, Boca Raton,FL, 2nd edition.

Gelman, A., Roberts, G., and Gilks, W. (1996). Efficient Metropolis jumping rules. 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.

Jackman, S. (2009). *Bayesian analysis for the social sciences*. John Wiley & Sons, New York.

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.

Newman, M. E. J. and Barkema, G. T. (1999). *Monte Carlo Methods in Statistical Physics*. Oxford University Press, New York.

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 )
}
```