Stat 8054 Lecture Notes: Autocovariances in MCMC

Charles J. Geyer

December 22, 2023

1 License

2 R

- The version of R used to make this document is 4.3.2.
- The version of the `rmarkdown` package used to make this document is 2.25.
- The version of the `mcmc` package used to make this document is 0.9.8.

3 Functionals of a Markov Chain

If X_1, X_2, … is a Markov chain and g is a real-valued function on the state space, then $g(X_1)$, $g(X_2)$, … is called a *functional* of the Markov chain (Geyer, 2011, Introduction to MCMC, in *Handbook of Markov Chain Monte Carlo*, hereinafter referred to as the "Intro", Section 1.6)

4 Estimates of Expectations of Functionals of a Markov Chain

If we cannot calculate, either by hand or by using a computer algebra system, an expectation

$$\mu = E\{g(X)\}$$

for some random object X and some real-valued function g on the state space (the set where X takes values), then we estimate it by simulating a Markov chain X_1, X_2, …, whose unique equilibrium distribution is the distribution of X, and using

$$\hat{\mu}_n = \frac{1}{n} \sum_{i=1}^{n} g(X_i)$$

as our estimator (these are equations (1.1) and (1.2) in the Intro).

5 Asymptotic Variance and Autocovariances

The Markov chain central limit theorem says that $\hat{\mu}_n$ is a consistent and asymptotically normal estimator of μ with asymptotic variance σ^2/n, where

$$\sigma^2 = \gamma_0 + 2 \sum_{k=1}^{\infty} \gamma_k$$

where

$$\gamma_k = \text{cov}\{g(X_i), g(X_{i+k})\}$$

(these are equations (1.9) and (1.8) in the Intro Intro), where here the covariances refer to the stationary Markov chain having the same transition probabilities as the Markov chain actually being run so that there is no dependence on i as is required for this to be a good definition (more on this in Section 1.8 of the intro).

The natural estimator of the "little gammas" γ_k is

$$\hat{\gamma}_k = \frac{1}{n} \sum_{i=1}^{n-k} \{[g(X_i) - \hat{\mu}_n][g(X_{i+k}) - \hat{\mu}_n]\}$$

(this is equation (1.10) in the Intro).

Each γ_k is called an *autocovariance* (of this functional of the Markov chain) and the function $k \mapsto \gamma_k$ is called the *autocovariance function*. R function `acf` estimates autocovariance functions.

6 Behavior of the Autocovariance Estimates

The behavior of the autocovariance estimates is extremely bad. For large k, when γ_k is nearly zero, the variance of $\hat{\gamma}_k$ does not depend on k and hence does not go to zero. Similarly, the covariance of $\hat{\gamma}_k$ and $\hat{\gamma}_{k+m}$ does not depend on k, and hence does not decrease with k. Thus, the part of the autocovariance function that corresponds to k such that γ_k is nearly zero, looks like a stationary time series. Geyer (1992, Practical Markov Chain Monte Carlo, *Statistical Science*, Section 3.1, discusses this, not deriving anything new but rather citing earlier work by Bartlett and Priestly).

As a simple example, we take the Markov chain from the examples section of the help page for R function `initseq`

```
n <- 2e4
rho <- 0.99
x <- arima.sim(model = list(ar = rho), n = n)
```

This is an AR(1) time series, which is also a Markov chain (Section 1.9 of the Intro).

Unlike in the Intro), we look at the boring part of the estimated autocovariance function (where the true autocovariance function is nearly zero). The following code

```
aout <- acf(x, type = "covariance", lag.max = n / 5, plot = FALSE)
bound <- max(abs(aout$acf[aout$lag >= 400]))
plot(aout, xlim = c(400, max(aout$lag)), ylim = c(-1, 1) * bound)
```

produces the following figure.

Although we think the true autocovariance function should go to zero as k goes to infinity (more on this later), the estimated autocovariance function does not do this. Both the figure and the theory discussed above show this.

Thus the naive estimator of σ^2 which just adds up the estimated autocovariances

$$\hat{\gamma}_0 + 2 \sum_{k=1}^{n-1} \hat{\gamma}_k$$

is a *terrible* estimator! It isn't even consistent. It's variance is $O(1)$ not $O(1/n)$ as one would expect for a good estimator (Geyer, 1992, Section 3.1, cited above).

Incidentally, this explains why we divided by n rather than by $n - k$ in the expression for $\hat{\gamma}_k$. If the estimator is already terrible for large k, then dividing by $n - k$ rather than n would only make it worse. Also our definition agrees with what R function `acf` computes and with most of the time series literature.

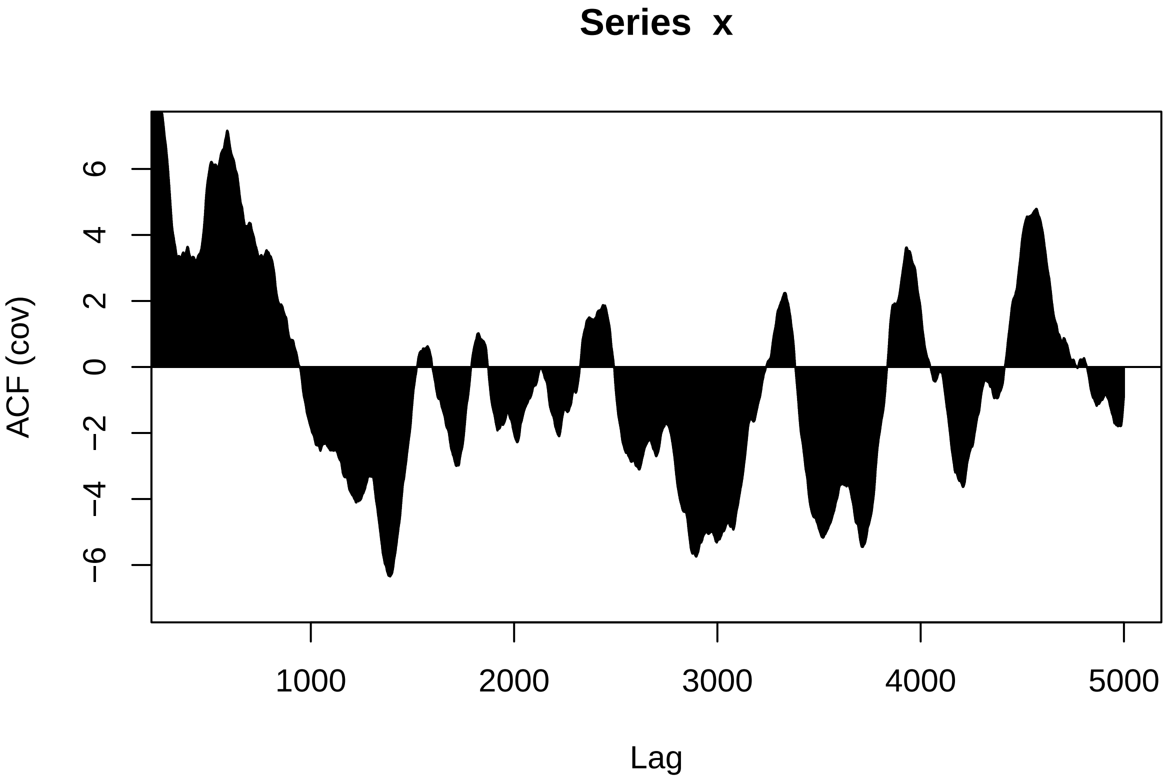

Figure 1: Estimated Autocovariance Function

This phenomenon was well understood in the time series literature long before most statisticians became interested in MCMC. So many methods of dealing with the problem for stationary time series (that are not necessarily functionals of Markov chains) have been proposed. But none is so good as to be obviously better than the others, and none is anywhere as good for functionals of reversible Markov chains as the methods proposed by Geyer (1992, cited above, Section 3.3) and called *initial sequence methods* and implemented in R function `initseq` in CRAN package `mcmc`.

7 Theory of Initial Sequence Estimators

The infinite-dimensional analog of eigenvalues and eigenvectors, which is called the *spectral decomposition of a bounded operator on a Hilbert space*, is used by Geyer (1992, cited above, Section 3.3) to prove things about the behavior of the sequence of "big gammas"

$$\Gamma_k = \gamma_{2k} + \gamma_{2k+1}$$

But first we look at the little gammas

$$\gamma_k = \int \lambda^k \, E_g(d\lambda)$$

where E_g is some positive measure concentrated on the interval $(-1, 1)$ of the real line if the Markov chain is irreducible and aperiodic, which it must be in order to be useful in MCMC (Chan and Geyer, 1994, Discussion of the paper by Tierney, *Annals of Statistics*, Lemma 6 and the surrounding discussion).

In this equation the λ are the infinite-dimensional analog of eigenvalues. There are infinitely many of them. The infinite-dimensional analog of eigenvectors is hidden in the measure E_g. The details don't matter because we cannot calculate E_g. Nor do we even need to know what bounded operator on a Hilbert space this comes from. It is the transition operator of the Markov chain (Chan and Geyer, 1994, cited above, pp. 1750 ff.), but that doesn't help us do any calculations.

Nor does this representation tell us much about the autocovariance function, but when we use this representation on the big gammas we get

$$\Gamma_k = \int \lambda^{2k}(1 - \lambda) \, E_g(d\lambda)$$

from which we immediately see that, unlike the little gammas, the big gammas are strictly positive (because even powers are positive and $1 - \lambda$ is positive because λ is concentrated on the interval $(-1, 1)$). Actually, a point that Geyer (1992) missed, is that E_g could be concentrated at zero, but this can only happen when the Markov chain is actually independent and identically distributed, which is not interesting (no source for this, it comes from a paper I am working on — a triviality but it needs to be mentioned in order to make the theorem about initial sequence estimators correct).

It is also obvious from this representation that the big gammas are strictly decreasing in k, because λ^{2k} is decreasing in k when λ is less than one in absolute value, which it is almost everywhere with respect to E_g. It is a little less obvious that the big gamma sequence is a strictly convex function of k but Geyer (1992, cited above) proves this in just a few lines.

Much less obvious is that

$$
\begin{aligned}
\sigma^2 &= -\gamma_0 + 2 \sum_{k=0}^{\infty} \Gamma_k \\
&= -\gamma_0 + 2 \sum_{k=0}^{\infty} \gamma_k \\
&= \lim_{n \to \infty} \int \left(-1 + 2 \sum_{k=0}^{n} \lambda^k \right) E_g(d\lambda) \\
&= \lim_{n \to \infty} \int \left(-1 + 2 \frac{1 - \lambda^n}{1 - \lambda} \right) E_g(d\lambda) \\
&= \lim_{n \to \infty} \int \frac{1 + \lambda - 2\lambda^n}{1 - \lambda} \, E_g(d\lambda) \\
&= \int \frac{1 + \lambda}{1 - \lambda} \, E_g(d\lambda)
\end{aligned}
$$

by monotone and dominated convergence.

The Kipnis-Varadhan Markov chain central limit theorem for functionals of reversible Markov chains (Kipnis and Varadhan, 1986, Central limit theorem for additive functionals of reversible Markov processes and applications to simple exclusions, *Communications in Mathematical Physics*) says the central limit theorem holds if the integral above is finite.

So that tells us the Γ_k must decrease to zero as k goes to infinity if there is to be a central limit theorem.

8 Example of Initial Sequence Estimators

8.1 Initial Positive Sequence Estimator

So let's look at the initial sequence estimators for our example. The following code

```r
library("mcmc")
iout <- initseq(x)
plot(seq(along = iout$Gamma.pos) - 1, iout$Gamma.pos,
    ylab = expression(Gamma[k]), xlab = expression(k))
abline(h = 0)
```

produces the following figure.

(this figure and the other figures in this handout change every time the document is processed by R function `render` in R package `rmarkdown` because we do not set random number generator seeds anywhere in this document).

This is already a very good estimator, we first estimated all of the big gammas by the obvious

$$
\widehat{\Gamma}_k = \hat{\gamma}_{2k} + \hat{\gamma}_{2k+1}
$$

and then set to zero all of the negative $\widehat{\Gamma}_k$ and all of the $\widehat{\Gamma}_k$ after any that are negative. That's what is shown in the figure. Already we have gotten rid of all the spurious wiggles of the autocovariance function for large lag (by setting them to zero as just described). Summing these gives a pretty good estimator.

8.2 Initial Decreasing Sequence Estimator

We can do a little better by imposing monotonicity. The following code

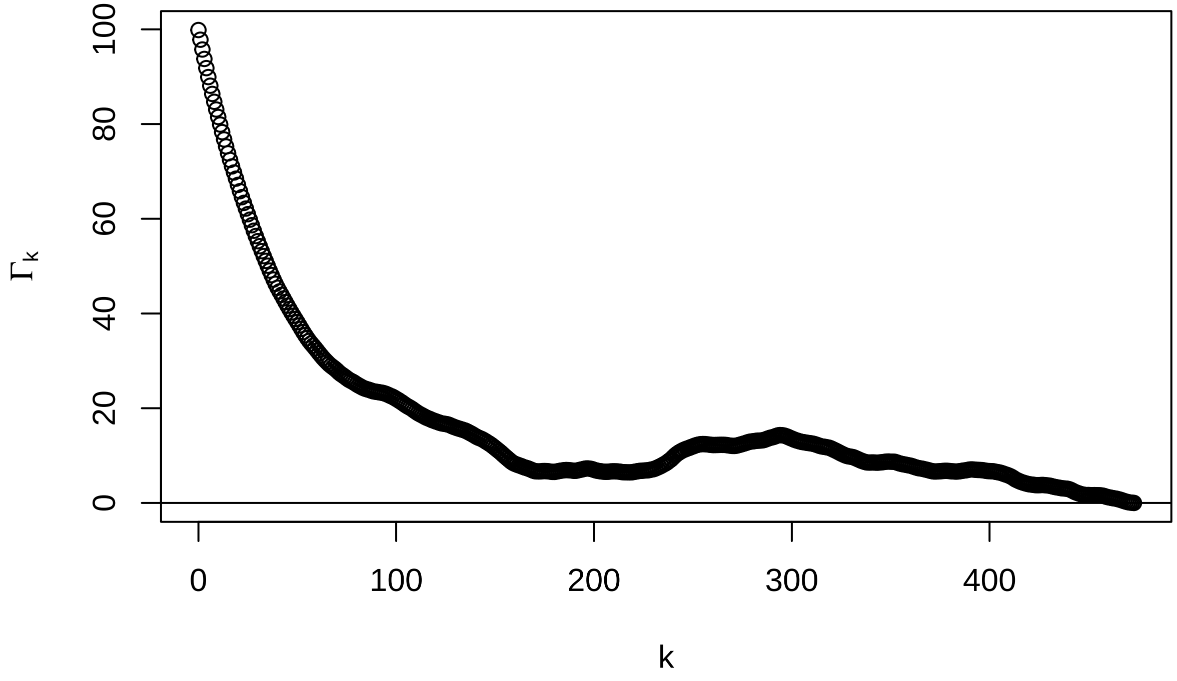

Figure 2: Initial Positive Sequence Estimator

```
plot(seq(along = iout$Gamma.pos) - 1, iout$Gamma.pos,
    ylab = expression(Gamma[k]), xlab = expression(k))
abline(h = 0)
lines(seq(along = iout$Gamma.con) - 1, iout$Gamma.con, col = "blue", lwd = 3)
```

produces the following figure.

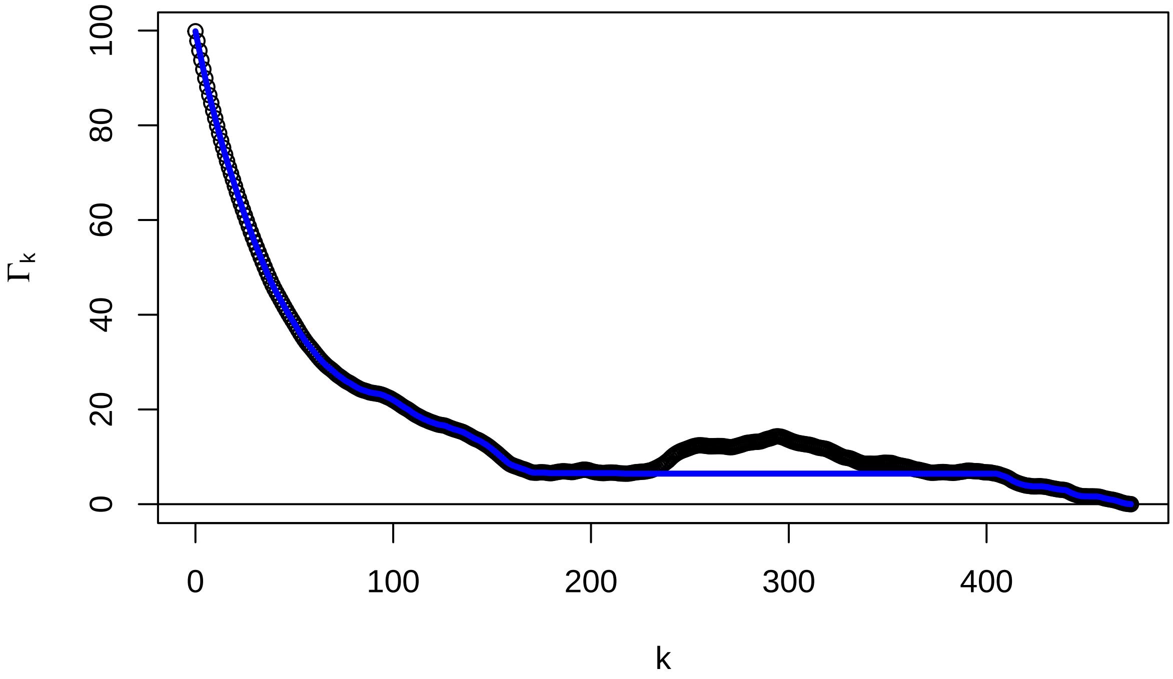

Figure 3: Initial Decreasing Sequence Estimator. Dots are initial positive sequence estimator. Blue line is initial decreasing sequence estimator

The improvement cannot even be seen on the plot, but there is some.

```
iout$var.pos
```

```
## [1] 15888.96
```

```
iout$var.dec
```

```
## [1] 14600.92
```

```
iout$var.pos - iout$var.dec
```

```
## [1] 1288.041
```

```
(iout$var.pos - iout$var.dec) / iout$var.dec
```

```
## [1] 0.08821638
```

about a 8.8 percent decrease in the estimator.

8.3 Initial Convex Sequence Estimator

We can do a little better by imposing convexity. The following code

```
plot(seq(along = iout$Gamma.dec) - 1, iout$Gamma.dec,
    ylab = expression(Gamma[k]), xlab = expression(k))
abline(h = 0)
lines(seq(along = iout$Gamma.con) - 1, iout$Gamma.con, col = "red", lwd = 3)
```

produces the following figure.

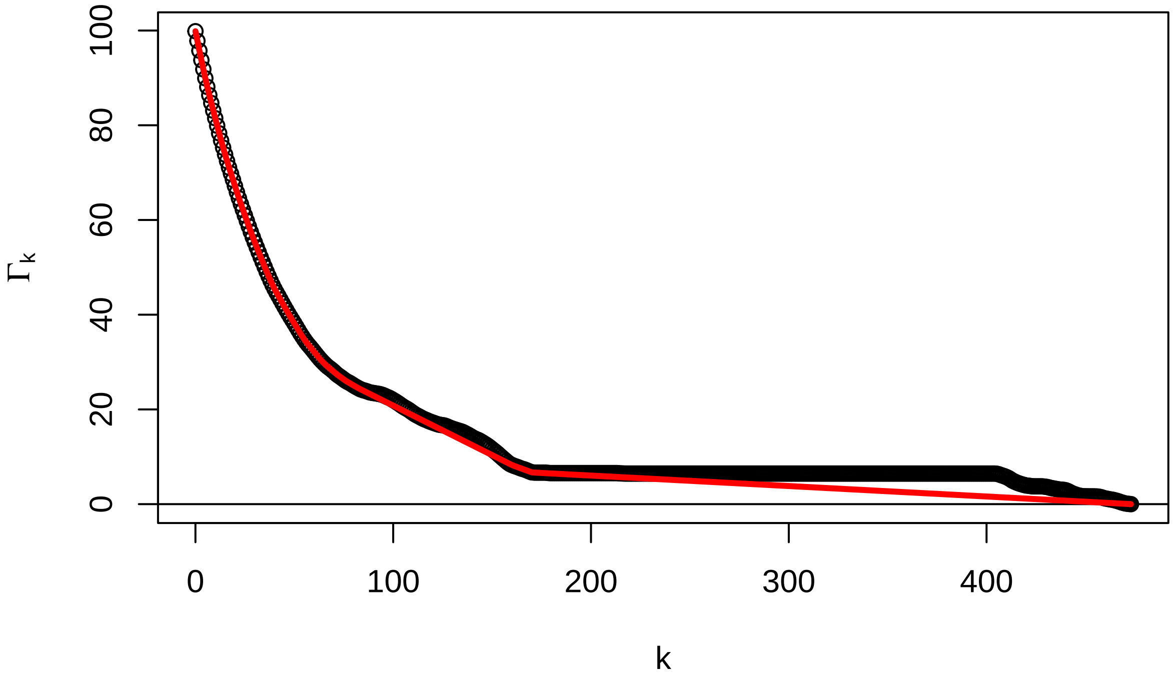

Figure 4: Initial Convex Sequence Estimator. Dots are initial decreasing sequence estimator. Red line is initial convex sequence estimator

The improvement cannot even be seen on the plot, but there is some.

```
iout$var.dec
```

```
## [1] 14600.92
```

```
iout$var.con
```

```
## [1] 13051.15
```

```
iout$var.dec - iout$var.con
```

```
## [1] 1549.774
```

```
(iout$var.dec - iout$var.con) / iout$var.con
```

```
## [1] 0.1187462
```

about a 11.9 percent decrease in the estimator.

8.4 Some More Examples

Let's just try this a bunch of times.

```
nsim <- 20
result <- NULL
for (i in 1:nsim) {
    x <- arima.sim(model = list(ar = rho), n = n)
    iout <- initseq(x)
    tmp <- iout[c("var.pos", "var.dec", "var.con")]
    tmp$imp.dec <- with(iout, 100 * (var.pos - var.dec) / var.dec)
    tmp$imp.con <- with(iout, 100 * (var.dec - var.con) / var.con)
    result <- rbind(result, unlist(tmp))
}
round(result, 2)
```

```
##        var.pos  var.dec  var.con imp.dec imp.con
##  [1,] 11855.34 11449.27 11246.00    3.55    1.81
##  [2,]  7833.80  7833.80  7816.81    0.00    0.22
##  [3,] 15663.81 14719.74 13711.82    6.41    7.35
##  [4,] 14370.66 13464.26 13038.44    6.73    3.27
##  [5,]  8299.29  8299.29  8239.74    0.00    0.72
##  [6,]  6874.06  6871.66  6846.03    0.04    0.37
##  [7,]  8757.19  8747.25  8672.22    0.11    0.87
##  [8,]  8694.83  8675.99  8497.94    0.22    2.10
##  [9,]  6264.28  6264.28  6231.62    0.00    0.52
## [10,]  9045.71  8989.54  8850.59    0.62    1.57
## [11,]  8062.66  8056.73  8009.11    0.07    0.59
## [12,]  9191.10  9191.10  9188.02    0.00    0.03
## [13,]  8966.03  8664.25  8287.15    3.48    4.55
## [14,] 13104.39 13103.68 12690.30    0.01    3.26
## [15,]  9304.05  9304.05  9145.16    0.00    1.74
## [16,]  8425.31  8424.34  8322.46    0.01    1.22
## [17,]  6835.82  6835.82  6802.62    0.00    0.49
## [18,] 10586.13 10585.55 10229.67    0.01    3.48
## [19,] 10229.96 10229.96 10226.69    0.00    0.03
## [20,]  9881.26  9881.26  9824.75    0.00    0.58
```

We can see that the amount of improvement (fourth and fifth columns, relative improvements in percent) is quite variable. Sometimes negligible, sometimes appreciable. Most of the time, the initial positive sequence estimator is just as good as the others. But rarely, the initial decreasing sequence estimator or the initial convex sequence estimator produce big wins.

We can see that the estimators themselves (first, second, and third columns) are also quite variable. That is because of the small Monte Carlo sample size. Increasing `n` would improve the estimates (of course).