MCMC Diagnostics

written by Patrick Breheny and Heejung Shim

MCMC Diagnostics - Introduction

As we have seen, there are two main concerns with the use of MCMC methods for posterior inference:

- Has our chain converged in distribution to the posterior (target distribution)?
- Correlation in samples simulated by MCMC can cause inefficiencies in simulations.

Let's learn about methods - both informal (graphical/visual checks) and formal - for assessing each of the above concerns.

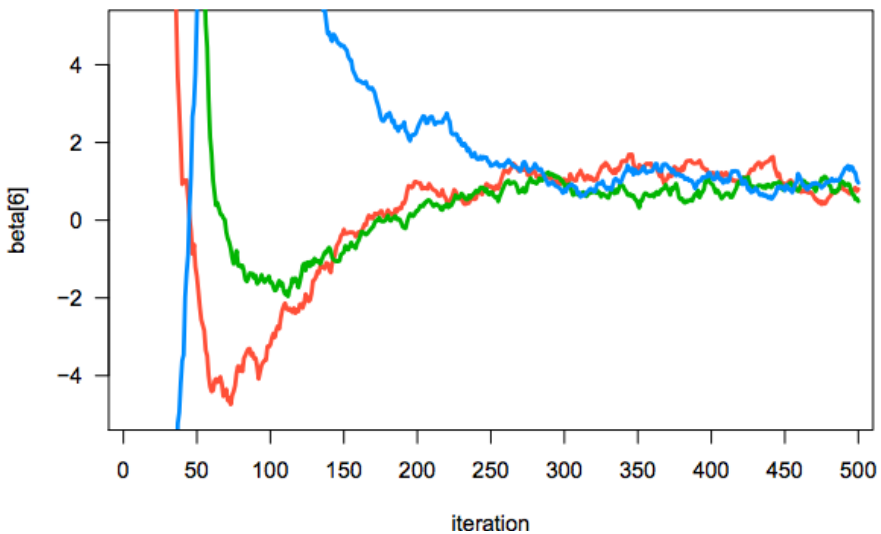
Has our chain converged in distribution to the posterior (target distribution)?

Convergence

Most approaches for detecting convergence, both formal and informal, rest on the idea of

- running multiple Markov chains with overdispersed initial values and
- observing whether they come together and start to behave similarly (if they do, we can pool samples from each chain).

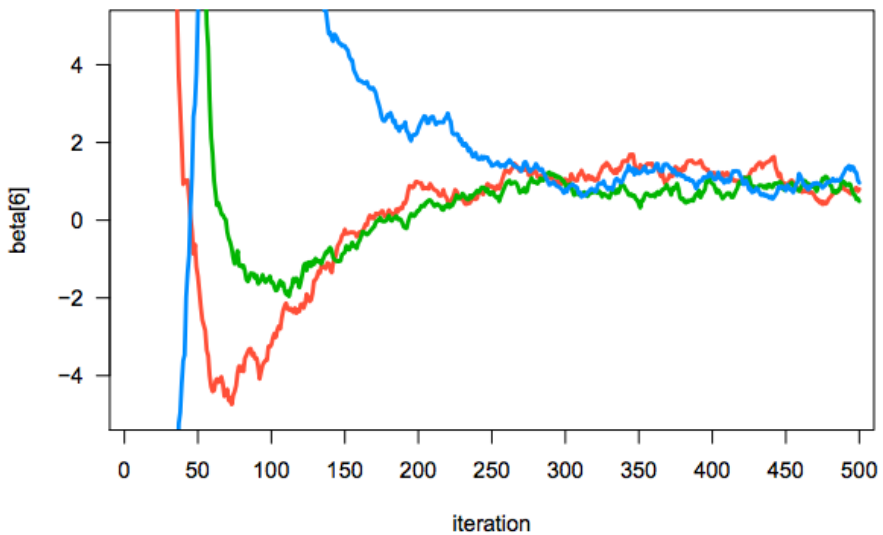
'Overdispersed' means 'more variable than the target distribution' (i.e., posterior distribution).



burn-in

To diminish the effect of initial values, we generally discard the first half of each chain and focus attention on the second half.

The practice of discarding early iterations in Markov Chain simulation is referred to as 'burn-in'; depending on the context, different burn-in fractions can be appropriate.



Brooks-Gelman-Rubin (BGR) diagnostic

- Although looking at trace/history plots is certainly useful, it is also desirable to obtain an objective, quantifiable measure of convergence.
- Numerous methods exist, although we will focus on the measure originally proposed in Gelman and Rubin (1992).

Brooks-Gelman-Rubin (BGR) diagnostic

Suppose that we have run M independent chains each for time $2T$, and we wish to decide if the final T observations look stationary (i.e., converged in distribution to target distribution)?

The diagnostic works best if the M initial values are widely separated.

Brooks-Gelman-Rubin (BGR) diagnostic

The basic idea of the estimator is as follows (the actual estimator makes a number of modifications to account for degrees of freedom):

- Let B denote the standard deviation of the pooled sample of all MT iterations (the between-chain variability).
- Let W denote the average of the M within-chain standard deviations from each chain.
- Put $\hat{R} = B/W$ then if $\hat{R} < 1.05$ then we accept the last T observations of each chain as stationary.
- Collect the MT samples from the second halves of all the chains together and treat them as a sample from the target distribution.

Note that if $\hat{R} >> 1$, this is clear evidence that the chains have not converged, and as $T \to \infty$, $\hat{R} \to 1$.

Brooks-Gelman-Rubin (BGR) diagnostic

Even if Markov chain appears to converge and has passed the test of convergence, it still may actually be far from convergence if important areas of the target distribution were not captured by initial values and are not easily reachable by the simulation algorithm.

Choose overdispersed initial values!!

Brooks-Gelman-Rubin (BGR) diagnostic

We apply the BGR diagnostic systemmatically to choose a burn-in period.

Suppose we have n samples from each of M chains.

- Split the samples up into Q batches of size a, then for $q = 1, \ldots, Q$ calculate $\hat{R}(q) = B(q)/W(q)$, where $B(q)$ and $W(q)$ are calculated using the second half of the sample from 1 to qa.
- Plot $\hat{R}(q)$, $B(q)$ and $W(q)$ against q , and look for the point where B and W have converged, and $\hat{R}(q) < 1.05$.
- We then discard samples 1 to $\frac{1}{2}qa$ as burn-in.

MCMC Diagnostics

As we have seen, there are two main concerns with the use of MCMC methods for posterior inference:

- Has our chain converged in distribution to the posterior (target distribution)?
- Correlation in samples simulated by MCMC can cause inefficiencies in simulations.

Has our chain converged in distribution to the posterior (target distribution)?

- trace/history plots.
- $\hat{R} < 1.05$

Our chain has not converged!?

- Increase the length of chain.
- Add efficient proposal methods.

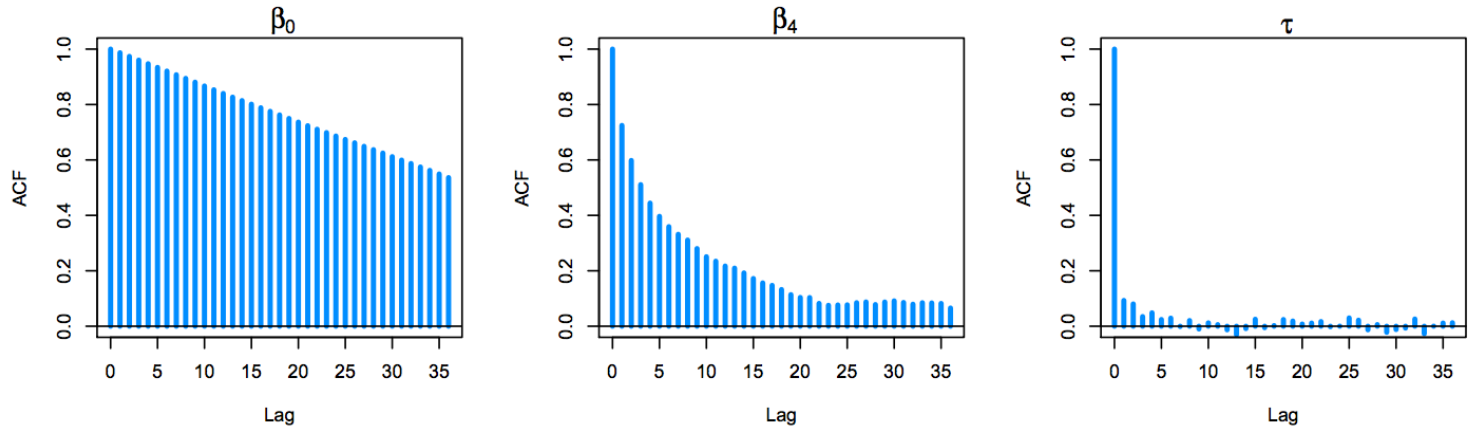
Correlation in samples simulated by MCMC can cause inefficiencies in simulations.

ex) Compared to 4000 independent samples, 4000 correlated samples contain less information about the posterior.

- Quantify correlation?
- How to deal with correlation?
- How long must we run our chains to obtain reasonably accurate estimates of the posterior?

Quantify correlation

- The autocorrelation between two states s and t of a Markov chain is defined, simply, as the correlation between $\theta^{(s)}$ and $\theta^{(t)}$.
- If the chain is stationary, in the sense that its mean and variance are not changing with time, then the correlation between $\theta^{(t)}$ and $\theta^{(t+k)}$ does not depend on t; this is known as the lag-k autocorrelation.



How to deal with correlation? Thinning

- A somewhat crude, yet reasonably effective, method dealing with autocorrelation is to only keep every kth simulated sample from each Markov chain and discard the rest; this is known as thinning the chain.
- The advantages of thinning are (a) simplicity and (b) a reduction in memory usage.
- The disadvantage is that we are clearly throwing away information.

How long must we run our chains to obtain reasonably accurate estimates of the posterior?

Effective Sample Size (ESS) for a parameter is the sample size of an independent sample giving the same amount information about the posterior as the correlated MCMC sample.

ESS $>$ 4,000

How to estimate ESS?

ESS $=$ [posterior standard deviation/MC error]^2

- Read Diagnostics_additional.pdf to see ideas about how some software packages computes MC error.

ESS $= \frac{n}{1+2\sum_{i=1}^{\infty}\rho_i} = n\frac{1-\rho_1}{1+\rho_1}$, where ρ_i is the lag i autocorrelation.

- Read Diagnostics_additional.pdf to see the derivation.