# Revisiting the Gelman-Rubin Diagnostic

Christina Knudson

March 28, 2019

## 1 Introduction

The Gelman-Rubin (GR) diagnostic has been one of the most popular diagnostics for MCMC convergence. The GR diagnostic framework relies on $m$ parallel chains ($m \geq 1$), each run for $n$ steps. The GR statistic (denoted $\hat{R}$) is the square root of the ratio of two estimators for the target variance. In finite samples, the numerator overestimates this variance and the denominator underestimates it. Each estimator converges to the target variance, meaning that $\hat{R}$ converges to 1 as $n$ increases. When $\hat{R}$ becomes sufficiently close to 1, the GR diagnostic declares convergence.

## 2 Effective Sample Size

For an estimator, Effective Sample Size (ESS) is the number of independent samples with the same standard error as a correlated sample.

The following is an expression for the lugsail-based psrf $\hat{R}_L^p$ for $m$ chains, each of length $n$ with p components.

$$\hat{R}_L^p = \sqrt{\left(\frac{n-1}{n}\right) + \frac{m}{\widehat{\text{ESS}}_L}},$$

Rearranging this yields an estimator of effective sample size:

$$\widehat{\mathrm{ESS}}_L = \frac{m}{\left(\hat{R}_L^p\right)^2 - \left(\frac{n-1}{n}\right)}.$$

*Remark* 1. **Doots!!** I remember we discussed and mentioned that the total ESS is not just the sum of the individual chains' ESSs. But this kind of is. So... am I using the wrong equation?

*Remark* 2. **?** explain that a minimum simulation effort must be set to safeguard from premature termination due to early bad estimates of $\sigma^2$. We concur and suggest a minimum simulation effort of $n = M_{\alpha,\epsilon,p}$.

## 3 To Do

■ Add ESS calculation (now in `gr.diag` and `n.eff`).

□ Add to `n.eff`.

1. Call `target.psrf` using the info in their `mcmc.list` using defaults.
2. Compare ESS to output of target.psrf and tell them whether this is sufficient for convergence.
3. If insufficient, calculate how many more samples needed using

$$\frac{n_{\text{current}}}{n_{\text{current eff}}} \approx \frac{n_{\text{target}}}{n_{\text{target eff}}} \quad \Longrightarrow \quad \frac{n_1 \; n_{\text{target eff}}}{n_{1\text{eff}}} \approx n_{\text{target}}.$$

4. Add more options: add the ability to change the defaults for `gr.diag` and `target.psrf`.

□ Clean up documentation

□ Add an example to `gr.diag` and `n.eff`

□ Check citations

□ Check descriptions and theory

□ Read through manual (pdf version)