Monitoring the Convergence of MCMC Chains

STAT 548 Qualifying Paper Report

Justin J. Zhang

February 18, 2026

1 Introduction

With the development of *Markov Chain Monte Carlo* (MCMC) sampling algorithms, statisticians and researchers are able to estimate parameters of interest in complex problems ranging from planet detection to disease modelling. However, one prevalent concern that has not been completely solved in the intervening decades is accurately diagnosing whether MCMC chains have converged (unless otherwise stated, this is convergence to a stationary distribution). A popular statistic to do so is the scale reduction factor $\hat{R}$ (Gelman, Rubin, 1992) which measures how well multiple chains initialized at different points can recover the same estimate, scaled by distributional variance. In essence, we want to see how well the chains "forget" their respective starting points, and converge to a stationary distribution (note that this may not be the target distribution). In most standard problems, $\hat{R}$ does an excellent job of monitoring convergence, but there are a number of significant underlying issues that can arise:

1. What threshold implies a "good" $\hat{R}$ value is somehwat arbitrary and problem dependent.
2. $\hat{R}$ may confuse non-convergence (high value) with a short sampling phase, i.e. we simply need to sample longer chains.
3. Since long sampling phases are needed for $\hat{R}$, the necessary computational power needed for MCMC is high.

Margossian et al. (henceforth denoted MEA) create a generalization of $\hat{R}$ called *nested* $\hat{R}$, denoted $\hat{R}_v$, that leverages the *many short-chains regime* to mitigate the 2nd and 3rd issues. Here, we are sampling many more MCMC chains, but for a far lesser number of sampling iterations. To aid in the effectiveness of $\hat{R}_v$ is the recent rise in Graphics Processing Unit (GPU) programming, which allows us to get a far greater number of samples for marginally more computational cost.

In this paper, we will first provide some relevant background on $\hat{R}$, GPU programming and its corresponding GPU-compatible MCMC algorithms. Then we will formally introduce $\hat{R}_v$, and prove some convergence properties. Lastly, we wil discuss strengths and limitations of this new method, along with some further research avenues.

2 Relevant Background Work

We starting by introducing core ideas to $\hat{R}_v$, namely the standard $\hat{R}$ diagnostic, the rise of GPU programming, and associated algorithms.

2.1 Scale Reduction Factor

Iterative finite-length simulation (i.e. running multiple chains) was introduced in order to reduce the bias induced by the starting point in MCMC samples (Gelman, Rubin, 1992). Multiple chains allow us to measure the variability of our MCMC estimator, which in turn helps track convergence. Intuitively, we hope to see that the variance between chains is dominated by the variance within chains, as this means our chains are "forgetting" its starting point and producing estimators in close agreement. Formally, for M MCMC chains of length

N, let $f(x_n^{(m)})$ be the estimator of the nth point from the mth chain, $\hat{f}_N^{(m)}$ be the estimator of the mth chain and $\bar{f}_N$ be the estimator across all chains. We define the *scale reduction factor* $\hat{R}$ as

$$\hat{B} = \frac{1}{M-1}\sum_{m=1}^{M}(\hat{f}_N^{(m)} - \bar{f}_N)^2 \qquad \hat{W} = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{N-1}\sum_{n=1}^{N}(f(x_n^{(m)}) - \hat{f}_N^{(m)})^2$$

$$\hat{R} = \sqrt{\frac{N-1}{N} + \frac{\hat{B}}{\hat{W}}}$$

Generally, $\hat{R} < 1.01$ signals convergence (REF), but this is a debated topic and there is no concrete threshold. Probabilistic software like STAN often use 4 independently initialized chains to estimate $\hat{R}$, but this may requires software to run very long chains to provide accurate estimates, and so MEA introduce $\hat{R}_v$ as an alternative.

2.2 GPU Programming

In order to run many chains without exhausting compute power, programs must make use of *parallel accelerators* like GPU's, that can run multiple chains *at the same time*. As an example, MEA show that GPU's allow us to run 512 chains in $\sim 20\%$ more compute time than 4 chain when sampling from the Rosenbrock distribution, giving us 128 times more samples for a modest increase in time. Unfortuanately, we cannot simply apply our favorite MCMC method on GPU's to get good results, as many algorithms are not GPU-compatible. This requires algorithms to use *Single-Instruction Multiple-Data* (SIMD) instructions, meaning they run the exact same code at the same time (Sountsov et al, 2024). Methods that run different number of steps for each chain will have significant wasted computation as it "slows down" to match the max number of steps. NUTS, a popular HMC tuning algorithm is not suited for parallelism as it is control-flow heavy and each chain can have different orbit lengths (leapfrog steps). Parallelization can be done through libraries like JAX for Python (BlackJAX for algorirthmic support), which we will demonstrate in the project section. Though GPU programming is far faster than CPU programming, a tradeoff is that its memory capacity is significantly lower, which may affect its ability to handle extremely high-dimensional data.

2.3 Parallelizable MCMC Methods

In order to leverage parallelization to increase computational efficiency we must use algorithms that follow SIMD. One such option is *Change in Estimator of Expected Square* Hamiltonian Monte Carlo (ChEES HMC), which is used in MEA. This algorithm tunes each chain *synchronously*, ensuring the same number of leapfrog steps are taken (Hoffman, Radul, Sountsov), though different number of leapfrog steps may be taken between iterations. For each HMC step, we tune each chain by maximizing the weighted gradients of autocorrelation of (centred) second moments

$$\nabla \text{ChEES} = \nabla \frac{1}{4}\mathbb{E}[(||\theta' - \mathbb{E}(\theta)||^2 - ||\theta - \mathbb{E}(\theta)||^2)^2]$$

The step size is tuned across chain to achieve a specific harmonic-mean acceptance probability. ChEES HMC outperforms NUTS with respect to ESS per gradient calculation, a measure of computational efficiency (Hoffman, et al). Moreover, it performs substantially more (between double and 10-fold) gradient calculations per second, and so has an even greater advantage looking at ESS per second. Now, since we are waiting for the "slowest" chain to catch up by tuning shorter trajectory lengths, we will see higher autocorrelation and hence a lower raw ESS. This implies that in lower-dimensional problems where computation time is less of an issue, non-GPU compatible methods like NUTS would work better. There may also be concern with complex, multimodal distributions where the modes have differing variances, as step size and trajectory may be trapped at small values by chains initialized in "narrow" modes. This is to say, parallelization success is algorithmic dependent and not always the solution.

3 Nested $\hat{R}$

We will now introduce the superchain regime, which leverages GPU's to run many chains in parallel, and produces a diagnostic, $\hat{R}_v$, that accurately reflects convergence. In contrast to $\hat{R}$, we initialize groups of M chains from the same point, and call these clusters of *constrained superchains* (there are also *naive superchaina* initialized at different points, but we assume superchains to be constrained in this paper). This construction allows us to remove the uncertainty in our MCMC estimate from a single initialization in our between-chain variance, because M independent chains from an identical starting point will have variance a factor of M less than a single chain. Thus, the between-chain variance (from $\hat{R}$) becomes a proxy for convergence to a stationary distribution. We formalize this idea throughout this section.

3.1 Definitions

Introducing superchains adds another layer to our notation. For the remainder of this paper, consider the superchain regime introduced by MEA where we have K superchains initialized independently from a distribution $x_0^{(k)} \sim p_0$. For theoretical guarantees to hold, we wish p_0 to be overdispersed. For each superchain, we run M independent subchains starting at $x_0^{(k)}$ for W warmup phases and N sampling phases. Let $f^{(nmk)}$ be the nth (sampling phase) sample from the mth subchain of kth superchain. We can define the respective subchain, superchain, and overall means for $m = 1, \ldots, M$ and $k = 1, \ldots, K$.

$$\bar{f}^{(.mk)} = \frac{1}{N}\sum_{n=1}^{N} f^{(nmk)} \qquad \bar{f}^{(..k)} = \frac{1}{M}\sum_{m=1}^{M} \bar{f}^{(.mk)} \qquad \bar{f}^{(...)} = \frac{1}{M}\sum_{k=1}^{K} \bar{f}^{(..k)}$$

As before, we define between-superchain and within-superchain variance, however this

time the latter consists of both between-subchain variance as well as within-subchain variance.

$$\hat{B}_v = \frac{1}{K-1}\sum_{k=1}^{K}(\bar{f}^{(..k)} - \bar{f}^{(...)})^2 \qquad \hat{W}_v = \frac{1}{K}\sum_{k=1}^{K}(\tilde{B}_k + \tilde{W}_k)$$

$$\tilde{B}_k = \begin{cases} \frac{1}{M-1}\sum_{m=1}^{M}(\bar{f}^{(.mk)} - \bar{f}^{(..k)})^2 & M > 1 \\ 0 & M = 1 \end{cases}$$

$$\tilde{W}_k = \begin{cases} \frac{1}{M}\sum_{m=1}^{M}\frac{1}{N-1}\sum_{n=1}^{N}(\bar{f}^{(nmk)} - \bar{f}^{(.mk)})^2 & N > 1 \\ 0 & N = 1 \end{cases}$$

We formally define $\hat{R}_v$ as the ratio of the stadard deviations of all superchains to the average within-superchain standard deviation.

$$\hat{R}_v = \sqrt{\frac{\hat{W}_v + \hat{B}_v}{\hat{W}_v}} = \sqrt{1 + \frac{\hat{B}_v}{\hat{W}_v}} \tag{1}$$

3.2 Decomposing $\hat{R}_v$

We now wish to understand how $\hat{R}_v$ behaves asymptotically. By the (strong) law of large numbers and law of total variance.

$$\begin{aligned}\hat{B}_v \overset{K\to\infty}{\to} B_v = \mathrm{Var}(\bar{f}^{(..k)}) &= \mathrm{Var}[\mathbb{E}(\bar{f}^{(..k)} \mid x_0^{(k)})] + \mathbb{E}[\mathrm{Var}(\bar{f}^{(..k)} \mid x_0^{(k)})] \\ &= \mathrm{Var}[\mathbb{E}(\bar{f}^{(.mk)} \mid x_0^{(k)})] + \frac{1}{M}\mathbb{E}[\mathrm{Var}(\bar{f}^{(.mk)} \mid x_0^{(k)})]\end{aligned}$$

The first term, $\mathrm{Var}[\mathbb{E}(\bar{f}^{(.mk)} \mid x_0^{(k)})]$, we call *non-stationary variance*, which quantifies how well each chain "forgets" its initial value. This goes to 0 when the estimators of each chain is in close agreement, and hence variance between chains decays. In this case we say the chain has converged to a stationary distribution, which makes non-stationary variance the quantity we wish to monitor. The second term, $\mathbb{E}[\mathrm{Var}(\bar{f}^{(.mk)} \mid x_0^{(k)})]$, we call *persistent variance*, which quantifies the variance of the chain itself. This will not go to 0 (as $K \to \infty$), but we do expect it to converge to $\mathrm{Var}\, f$ if our chain reaches stationarity. In the standard $\hat{R}$ setting, $M = 1$, and so persistent variance is a substantial component of $\hat{B}$, meaning $\hat{R}$ does not explicitly model non-stationary variance. To "kill" persistent variance for a single chain, we would have to run very long chains so that the ESS is substantial and $\mathbb{E}[\mathrm{Var}(\bar{f}^{(.mk)} \mid x_0^{(k)})] \approx \frac{1}{ESS}\mathbb{E}[\mathrm{Var}(\bar{f}^{(nmk)} \mid x_0^{(k)})]$. This is a major weakness of $\hat{R}$ we discussed in Section 1 that is mitigated by the superchain regime.

Now $\hat{R}_v$ does not exactly monitor non-stationary variance. From Eq. (1) we see it is scaled by within-superchain variance $\tilde{W}_v$. The asymptotic behaviour of $\tilde{W}_v$ is difficult to measure due to within-subchain variance (APPENDIX) but in the special case of $N = 1$, that goes away. In fact, MEA show that by (strong) law of large numbers, for $N = 1, M > 1$

$$\hat{W}_v \overset{K\to\infty}{\to} W_v = \mathbb{E}(\tilde{B}_k) = \mathbb{E}[\mathrm{Var}(\bar{f}^{(.mk)} \mid x_0^k)]$$

This proof is not immediate and will be left for appendix. Now we can rewrite Eq. (1) asymptotically (in K) by Continuous Mapping Theorem

$$\hat{R}_v \overset{K\to\infty}{\to} \sqrt{1+\frac{B_v}{W_v}} = \sqrt{1+\frac{\mathrm{Var}[\mathbb{E}(\bar{f}^{(\cdot mk)} \mid x_0^k)] + \frac{1}{M}\mathbb{E}[\mathrm{Var}(\bar{f}^{(\cdot mk)} \mid x_0^{(k)})]}{\mathbb{E}[\mathrm{Var}(\bar{f}^{(\cdot mk)} \mid x_0^{(k)})]}} \tag{2}$$
$$= \sqrt{1+\frac{1}{M}+\frac{\mathrm{Var}[\mathbb{E}(\bar{f}^{(\cdot mk)} \mid x_0^{(k)})]}{\mathbb{E}[\mathrm{Var}(\bar{f}^{(\cdot mk)} \mid x_0^{(k)})]}}$$

Persistent variance should converge (in N) to $\mathrm{Var}\, f$, as we will discuss in the project section. Given this, we have the relationship $\hat{R}_v < \epsilon \iff \mathrm{Var}[\mathbb{E}(\bar{f}^{(\cdot mk)} \mid x_0^{(k)})] < \epsilon$ for some small tolerance ϵ.

3.3 Further Considerations

To properly monitor superchain convergence to the target distribution, we must also consider sample bias in addition to variance. In fact we can break down the MCMC error into squared bias, non-stationary variance (monitored by $\hat{R}_v$), and persistent variance (negligeable with superchains)

$$\mathbb{E}((\bar{f}^{(\cdot\cdot k)} - Ef)^2) = (\mathbb{E}\bar{f}^{(\cdot\cdot k)} - \mathbb{E}f)^2 + \mathrm{Var}(\mathbb{E}(\bar{f}^{(\cdot\cdot k)} \mid x_0^{(k)})) + \mathbb{E}(\mathrm{Var}(\bar{f}^{(\cdot\cdot k)} \mid x_0^{(k)}))$$

Generally, a substantial warmup phase will eliminate the bias, but it is not immediately clear how long that should be. A more pressing problem is that bias may not converge to 0 if our chains are not finding the target distribution. A clear example of this is a multimodal distribution, say a mixture of Gaussians, with underdispersed initial distribution. In this case we may have small non-stationary variance as the chains do converge to a stationary distribution (single mode), but clearly that is not the target. In the Project section, we will explore conditions where non-stationary variance actually bounds squared bias, and so a small $\hat{R}_v$ directly implies that the expected error of MCMC has decayed.

Another issue we have that carries over from Section 1 is what threshold on $\hat{R}_v$ actually implies convergence. As we have seen in Eq. (2), the magnitude of persistent variance (and indirectly the target variance) will impact how well $\hat{R}_v$ monitors non-stationary variance. Specifically, if we set a threshold on $\hat{R}_v$, then our non-stationary variance will be bounded

$$\hat{R}_v \le \delta \iff \frac{\mathrm{Var}[\mathbb{E}(\bar{f}^{(\cdot mk)} \mid x_0^{(k)})]}{\mathbb{E}[\mathrm{Var}(\bar{f}^{(\cdot mk)} \mid x_0^{(k)})]} \le \delta^2 - 1 - \frac{1}{M} = \epsilon$$

Of course for this bound to even be feasible, we need $\delta^2 - 1 > \frac{1}{M}$, which for $\delta = 1.01$, we need at least $M = 50$. Moreover, when persistent variance is large, this bound can still be substantial, in which case non-stationary variance may not have converged. MEA propose to instead set a tolerance for scaled non-stationary variance

$$\frac{\mathrm{Var}[\mathbb{E}(\bar{f}^{(\cdot mk)} \mid x_0^k)]}{\mathbb{E}[\mathrm{Var}(\bar{f}^{(\cdot mk)} \mid x_0^k)]} \le \tau \implies \hat{R}_v \le \sqrt{1+\frac{1}{M}+\tau}$$

This bound is clearly context specific but should be small next to tolerable square error. In the project section, we will discuss ways how tight these bounds are.

4 Analysis and Discussion

MEA give an excellent breakdown of how $\hat{R}_v$ improves upon the problems with $\hat{R}$ mentioned in Section 1. However, this is not to say that it is inherently more useful in all situations. In one sense, efficiently running superchains hinges on algorithms that leverage parallel computation, and we have already touched on problems that can be faced there. In this section, we will analyze (BLANK) conditions where our proposed $\hat{R}_v$ computation falls short.

4.1 Algorithmic Dependence

Nested $\hat{R}_v$ relies on SIMD-compatible algorithms to run many chains in parallel. We provide ChEES HMC as one algorithm, but it may not be an efficient option for certain target distributions, and in some cases will return extremely biased estimators. Consider a (suitably distanced) multimodal distribution, for which ChEES HMC will trap individual chains in the respective modes, clearly not converging in general cases. A far better algorithm would allow for mode jumping, for example parallel tempering or annealing (Earl, 2005), these algorithms have their own drawbacks of longer chains, and need to set good temperature schedules between chains. What is so nice with ChEES HMC is that everything is automated and does not have to be manually tuned, whereas other algorithms like parllel tempering require that extra effort. Nowadays, there is seemingly a MCMC algorithm for every use case, but generalizing them to be SIMD-compatible can be a chore, and users must be able to recognize this when deciding to use the many-chains regime. When the resulting (parallel) sampling algorithms deliver biased estimators, it would still be better to sample few chains for a large number of samples with a CPU-compatible method.

4.2 False Convergence

There are specific use cases where $\hat{R}_v$ gives false signals for convergence in both directions. We have previously seen in Section 3.3 that a multimodal Gaussian with initial distribution in a single mode will have $\hat{R}_v$ signalling convergence, though it does not. Consider also a specific case where the target distribution is $0.5N(n, 1) + 0.5N(-n, 1)$ for some $n > 0$ with initial distribution normally distributed about 0. Because of the symmetry here, we expect (for large K) that about half of the chains will initialize and converge within each respective mode, and produce an estimator around 0. However, $\hat{R}_v$ will be large because the between-superchain variance is substantial here. This highlights the difference between the MCMC estimator converging (in probability) and the chains themselves actually mixing well. Note that when K is small here, we can underestimate between-superchain variance drastically if most initializations occur in the same mode. Though these examples are quite contrived, it does merit consideration in real applications when we think the target distribution may be multimodal.

4.3 Thresholds? Maybe move here from previous section

4.4 Alternative $\hat{R}$ Use Cases

A class of target distributions that have issues with standard $\hat{R}$ are heavy tailed distributions. If the target has infinite variance, then within-subchain variance will dominate between-superchain variance, and $\hat{R}_v$ will show convergence as well regardless if the chains have found the same estimate. A proposed fix in the standard regime is to use rank-normalized or folded $\hat{R}$, which use ranks and deviation from median respectively as opposed to raw estimates (vehtari et al). This can also be applied in our many-chains regime, producing equivalent modifications for $\hat{R}_v$. Moreover, there are other variations including split-$\hat{R}$ (Gelman, 2013), local-$\hat{R}$ (Moins, 2022) that can be used with the nested scheme, depending on use case.

1 Introduction

An overarching problem in Markov Chain Monte Carlo (MCMC) sampling is how to correctly monitor convergence (we refer to this as convergence to a stationary distribution). One diagnostic that was recently developed is *nested* $\hat{R}$, denoted $\hat{R}_v$, which monitors the variance of MCMC chains (Margossian, et al). It measures how well chains forget their starting point and produce estimators in agreement better than many popular diagnostics. However, to fully determine whether our chains have found the target distribution, we must ensure bias also decays, which is not monitored by $\hat{R}_v$. In general, we expect bias to decay with a sufficiently long warmup phase, but it is not immediately clear how long that should be. A more pressing problem is that bias may not converge to 0 if our chains are not finding the target distribution. A clear example of this is a multimodal distribution, say a mixture of Gaussians, with underdispersed initial distribution. In this case we may have small non-stationary variance as the chains do converge to a stationary distribution (single mode), but clearly that is not the target. Of course we would want to have a diagnostic that also monitors bias, but that is quite difficult in practice when we do not know the true mean of our target. Instead, we will show that with an overdispersed initial distirbution, squared bias is bounded by *non-stationary variance* (to be defined), which is directly monitored by $\hat{R}_v$, and hence we can use it to diagnose convergence. In this paper, we will introduce the fundamentals of MCMC convergence and the $\hat{R}_v$ diagnostic, propose and prove bounds for squared bias and non-stationary variance, and verify our claims with numerical experiments.

1.1 Nested $\hat{R}$

We starting by introducing the convergence diagnostic $\hat{R}_v$ (Margossian, et al). To compute this, we run K *constrained superchains*, which are groups of M independent MCMC subchains initialized at the same point $x_0^{(k)} \sim p_0$. To run many chains efficiently, we use SIMD-compatible parallel MCMC methods on GPU's like ChEES HMC, which in favorable conditions can run hundreds more chains than CPU methods in marginally more time (JAX paper REF). This construction allows us to reduce the effect within-chain variability in our MCMC samples from a single initialization, because M independent chains from an identical starting point will have variance a factor of M less than a single chain. Convergence occurs if our chains are "forgetting" its starting point and finding the same estimated values. We define $\hat{R}_v$ to measure the ratio of variance between superchains, and variance within superchains. A formal derivation is in (Appendix).

1.2 Convergence of MCMC Chains

To analyze error of MCMC superchains, we can break it down into bias and variance components. Let $f^{(nmk)}$ be the nth sample from mth subchain in kth superchain, $\bar{f}^{(.mk)}$ be the mean of mth subchain in kth superchain, and $\bar{f}^{(..k)}$ be the mean of kth superchain. Then,

$$\mathbb{E}((\bar{f}^{(..k)} - Ef)^2) = (\mathbb{E}\bar{f}^{(..k)} - \mathbb{E}f)^2 + \mathrm{Var}(\mathbb{E}(\bar{f}^{(..k)} \mid x_0^{(k)})) + \mathbb{E}(\mathrm{Var}(\bar{f}^{(..k)} \mid x_0^{(k)}))$$

The last term, *persistent variance* is negligeable in our superchain regime, as $\mathbb{E}(\mathrm{Var}(\bar{f}^{(..k)} \mid x_0^{(k)})) = \frac{1}{M}\mathbb{E}(\mathrm{Var}(\bar{f}^{(.mk)} \mid x_0^{(k)}))$, which goes to 0 as M increases. The second term, *non-stationary*

variance, will dissapear as chains converge to stationarity, and is monitored directly by $\hat{R}_v$ scaled by persistant variance (Margossian, et al). Together, using $\hat{R}_v$ in the superchain regime directly monitors variance of our MCMC chains. Estimating the first term, sqaured bias, requires us to have a good estimate of the sample mean, but this is never known in practice or else sampling is pointless. Thus, we require a proxy measure of bias, in order to determine convergence, which we will introduce in the next section.

1.3 Related Works

A detailed analysis of $\hat{R}_v$ properties and performance is found in Margossian et al. The question of what threshold on $\hat{R}_v$ (equivalently standard $\hat{R}$) actually implies convergence to a stationary distribution is prevalent, with no definitive answers (Vats, Knudson, 2021). Margossian et al. do argue that it suffices to have a tolerable error on non-stationary variance, though that is problem-dependent as well. There are unbiased MCMC algorithms that remove squared bias through coupling, and thus allow $\hat{R}_v$ to monitor squared error (Jacobs et al. 2020). However, the natural tradeoff is the relative inefficiency (i.e. higher variance) and far longer compute time (in fact it is not SIMD compatible and so the benefits of parallelization are not applicable). Additionally, there are methods like annealed importance sampling (Neale et al, 2001) and sequential Monte Carlo (Del Moral et al, 2006) that control bias without running egregiously long warmups.

2 Analysis of Convergence

We now look to show that with an overdispersed distribution, the squared bias is bounded by the non-stationary variance. The problem of how to choose a distribution that is both overdispersed and similar to our target distribution (in the sense that it does not initialize in areas with negligeble density) is important but will not be touched on. In this section we will theoretically derive bounds for non-stationary variance and bias, and show that they decay at the same rate. We base our ideas on unpublished works of Margossian.

2.1 Bounding Non-stationary Variance

For a given superchain initiazlized at $x_0^{(k)}$, we write the conditional bias

$$|\mathbb{E}(\hat{f}^{(.mk)} \mid x_0^{(k)}) - \mathbb{E}f| = b(x_0^{(k)})h(x_0^{(k)}, N) \tag{3}$$

where $b(x_0^{(k)}) = |f(x_0^{(k)}) - \mathbb{E}f|$ is the initial bias and h is a decay function with $h(x_0^{(k)}, 0) = 0$. Here, N is the total number of sampling steps inclding warmup. For a chain that converges to stationarity, we expect $h(x_0^{(k)}, N) \to 0$ in our sampling phase (but may not hold otherwise). We first state a bound for non-stationary variance in terms of these quantities.

Theorem 2.1. *Given Eq. (3), we can derive the following bound:*

$$\mathrm{Var}(b(x_0^{(k)})h(x_0^{(k)}, N)) \leq \mathrm{Var}\,\mathbb{E}(\hat{f}^{(.mk)} \mid x_0^{(k)}) \leq \mathbb{E}(b^2(x_0^{(k)})h^2(x_0^{(k)}, N)) \tag{4}$$

Proof. To obtain these bounds, we apply variance expansions and Eq. (3)

$$\begin{aligned}\operatorname{Var}(b(x_0^{(k)})h(x_0^{(k)}, N)) = \operatorname{Var}|\mathbb{E}(\hat{f}^{(\cdot mk)} \mid x_0^{(k)}) - \mathbb{E}f| &\leq \operatorname{Var}(\mathbb{E}(\hat{f}^{(\cdot mk)} \mid x_0^{(k)}) - \mathbb{E}f) \\ &= \mathbb{E}[(\mathbb{E}(\hat{f}^{(\cdot mk)} \mid x_0^{(k)}) - \mathbb{E}f)^2] \\ &= \mathbb{E}[b^2(x_0^{(k)})h^2(x_0^{(k)}, N)]\end{aligned}$$

The first inequality holds intuitively because absolute value reduces variation (also by monotonicity of expectation – further proof in appendix??). The second equality holds by variance expansion (should be obvious – elaborate??). Since $\mathbb{E}f$ is constant, we can substitute $\operatorname{Var}(\mathbb{E}(\hat{f}^{(\cdot mk)} \mid x_0^{(k)}) - \mathbb{E}f) = \operatorname{Var}\mathbb{E}(\hat{f}^{(\cdot mk)} \mid x_0^{(k)})$ in our above equations to get the desired bounds. □

We originally assumed that h depends on both initialization and sampling length, but we can relax this by assuming our decay $h(N)$ is solely dependent on sampling step. It has been shown that this is possible under uniform boundedness of the target to proposal ratio using the Metropolis Hastings algorithm (Wang, 2022). Alternatively, it would be worthwhile to investigate upper and lower bounds of the decay function depending on initialization. Under this relaxation we get the following corollary.

Corollary 2.2. *Suppose a decay function $h(N)$ independent of of the initialization. Then we can rewrite Eq. (4) as*

$$\operatorname{Var}(b(x_0^{(k)}))h^2(N) \leq \operatorname{Var}\mathbb{E}(\hat{f}^{(\cdot mk)} \mid x_0^{(k)}) \leq \mathbb{E}(b^2(x_0^{(k)}))h^2(N)$$

Under these assumptions, we see that the non-stationary variance will also decay at the rate $h(N)$. How tight these bounds are depends on the difference $\mathbb{E}(b^2(x_0^{(k)})) - \operatorname{Var}(b(x_0^{(k)})) = (\mathbb{E}b^2(x_0^{(k)}))^2$. When the expected initial bias is very small, this gives an exact equation for the decay of non-stationary variance.

2.2 Bounding Squared Bias

To derive bounds on the marginal bias, we write the initial variance as

$$\operatorname{Var}\mathbb{E}(\hat{f}^{(\cdot mk)} \mid x_0^{(k)}) = \operatorname{Var} f(x_0^{(k)})g(N)$$

where $\operatorname{Var} f(x_0^{(k)})$ is the initial variance and $g(N)$ is some decay function. We are assuming that the decay of non-stationary variance is independent of initialization but can relax that and use decay $g(x_0^{(k)}, N)$.

Proposition 2.3. *Suppose the conditional bias and non-stationary variances have decary rates $h(N), g(N)$ that are independent of initial point. Furthermore, assume that $g(N) = h^2(N)$. Then,*

$$\operatorname{Var} f(x_0^{(k)}) \geq (\mathbb{E}b(x_0^{(k)}))^2 \implies \operatorname{Var}\mathbb{E}(\hat{f}^{(\cdot mk)} \mid x_0^{(k)}) \geq (\mathbb{E}\hat{f}^{(\cdot mk)} - \mathbb{E}f)^2$$

Proof. We first bound the squared bias by the conditional squared bias with Jensen's inequality using the fact $\mathbb{E}\mathbb{E}f = \mathbb{E}f$ since it is constant,

$$|\mathbb{E}\hat{f}^{(\cdot mk)} - \mathbb{E}f| = |\mathbb{E}(\mathbb{E}(\hat{f}^{(\cdot mk)} \mid x_0^{(k)}) - \mathbb{E}f)| \leq \mathbb{E}|\mathbb{E}(\hat{f}^{(\cdot mk)} \mid x_0^{(k)}) - \mathbb{E}f| = h(N)\mathbb{E}|b(x_0^{(k)})|$$

It follows directy by assumption that

$$(\mathbb{E}\hat{f}^{(\cdot mk)} - \mathbb{E}f)^2 \leq h^2(N)(\mathbb{E}b(x_0^{(k)}))^2 \leq h^2(N)\operatorname{Var} f(x_0^{(k)}) = \operatorname{Var}\mathbb{E}(\hat{f}^{(\cdot mk)} \mid x_0^{(k)})$$

as desired. □

Notice that our condition here is exactly that of overdispersion, requiring the variance of our initial distribution to be greater than the sqaured expected distance (variance) from the true mean (formalize this – maybe with math). Note however that the converse is not true in general unless the conditional and marginal biases are equal.

2.3 Bounding Persistant Variance

For $\hat{R}_v$ to monitor non-stationary variance, we require persistant variance to be constant with respect to N and x_0^k, otherwise $\hat{R}_v$ estimate will be unduly influenced by the sampling step. Moreover, it is of interest how it behaves relative to the target variance, $\operatorname{Var} f$, which it should converge to. The following proposition provides bounds for persistant variance.

Proposition 2.4. *Suppose as before that the conditional bias has decay rate independent of initial point. Given Eq. (3), we can bound the persistant variance around the target variance*

$$\operatorname{Var} f - \mathbb{E}(b^2(x_0^{(k)}))h^2(N) \leq \mathbb{E}(\operatorname{Var}(\bar{f}^{(\cdot\cdot k)} \mid x_0^{(k)})) \leq \operatorname{Var} f + C\mathbb{E}(b^2(x_0^{(k)}))h^2(N)$$

where the constant $C = 3 + 2|\mathbb{E}f|$.

We leave the proof for the appendix.

3 Numerical Experiments

We made amazing contributions to the world of musical fractal pasta (McDonald, 2017; Tibshirani, 2013). We use Natbib, so be sure to use (Stein, 1981) for parenthetical references. Or you can say, according to Hastie et al. (2009), we should strive to balance truth and lies.

References

Hastie, T., Tibshirani, R. and Friedman, J. (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Verlag.

McDonald, D. J. (2017) Minimax Density Estimation for Growing Dimension. In *Proceedings of the* 20^{th} *International Conference on Artificial Intelligence and Statistics (AISTATS)* (eds. A. Singh and J. Zhu), vol. 54, 194–203. PMLR.

Stein, C. M. (1981) Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, **9**, 1135–1151.

Tibshirani, R. J. (2013) The lasso problem and uniqueness. *Electronic Journal of Statistics*, **7**, 1456–1490.