1 Posterior Averages

In Bayesian inference, the posterior distribution is

$$p(\theta \mid x) = \frac{1}{Z} p(x \mid \theta) p(\theta), \tag{1}$$

where θ denotes the model parameters, x denotes the data, $p(x \mid \theta)$ is the likelihood, $p(\theta)$ is the prior, and

$$Z = \int p(x \mid \theta) p(\theta) \, \mathrm{d}\theta \tag{2}$$

is the evidence. In many problems, Z is difficult to compute directly because it is an integral over the full parameter space.

Usually we are not interested in the value of the posterior density at every point. We want posterior averages. For example, if $h(\theta)$ is one parameter, a derived physical quantity, or a model prediction, then the posterior expectation is

$$\mathbb{E}[h(\theta) \mid x] = \int h(\theta) p(\theta \mid x) \, \mathrm{d}\theta. \tag{3}$$

If we had independent samples

$$\theta_1, \theta_2, \ldots, \theta_N \sim p(\theta \mid x), \tag{4}$$

then we could estimate this expectation by

$$\bar{h} = \frac{1}{N} \sum_{n=1}^{N} h(\theta_n). \tag{5}$$

For independent samples,

$$\mathrm{Var}(\bar{h}) = \frac{1}{N} \mathrm{Var}[h(\theta)]. \tag{6}$$

This is the basic Monte Carlo result. The error in the sample average decreases like $N^{-1/2}$.

The difficulty is that in realistic Bayesian problems, we usually cannot draw independent samples from $p(\theta \mid x)$ directly. MCMC replaces independent sampling with a correlated random walk whose long-time distribution is the posterior.

The purpose of MCMC is not to find the maximum of the posterior. It is to generate samples whose density in parameter space is proportional to the posterior distribution. These samples can then be used to compute posterior averages, credible intervals, and derived quantities.

2 Markov Chain Monte Carlo

A Markov chain produces a sequence of parameter values

$$\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, \ldots. \tag{7}$$

The next point depends on the current point through a transition rule,

$$\theta^{(n+1)} \sim K(\theta^{(n)}, \cdot). \tag{8}$$

In MCMC, this transition rule is chosen so that the posterior distribution is stationary. After the chain has run long enough and has forgotten its initial condition, the samples should be distributed according to $p(\theta \mid x)$.

This immediately gives two different questions. First, has the chain forgotten its initial condition? This is the burn-in question. Second, after burn-in, how many effectively independent samples do we have? This is the autocorrelation question.

The second question is necessary because MCMC samples are not independent. The sample at step $n + 1$ was generated from the sample at step n. Nearby samples therefore tend to be similar. A long chain may contain many samples, but the amount of independent information can be much smaller than the raw number of samples.

3 The Goodman–Weare Ensemble Sampler

`emcee` implements the affine-invariant ensemble sampler introduced by Goodman and Weare. Instead of evolving one chain, it evolves an ensemble of walkers,

$$\theta_1, \theta_2, \ldots, \theta_{N_{\mathrm{walker}}}. \tag{9}$$

Each walker is a point in parameter space. The ensemble as a whole moves through the posterior distribution.

The most important move in the Goodman–Weare sampler is the stretch move. To update walker i, the algorithm chooses another walker j and proposes

$$\theta_i' = \theta_j + z(\theta_i - \theta_j), \tag{10}$$

where z is a random stretch factor. The proposed point lies on the line connecting two walkers.

This is the sense in which the walkers are "holding hands." A walker does not move in isolation. It proposes its next position using the current position of another walker. The ensemble therefore contains information about the shape of the posterior.

This is useful when the posterior is stretched, tilted, or anisotropic. A simple random-walk Metropolis sampler proposes moves using a fixed local scale, so it can struggle with long, narrow distributions. The Goodman–Weare sampler uses the relative positions of walkers, so it adapts better to affine transformations of parameter space.

The ensemble structure helps emcee explore parameter space, but it does not make the samples independent. The walkers are coupled through the ensemble, and the ensemble still evolves step by step. Autocorrelation time measures how long this evolving ensemble remembers its previous state.

4 From an MCMC Chain to a Time Series

To define autocorrelation, we first turn the chain into a scalar time series. Suppose the chain produces samples

$$\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(N)}. \tag{11}$$

Choose a scalar function $f(\theta)$ and define

$$f_n = f(\theta^{(n)}). \tag{12}$$

Here f could be one parameter, the log posterior, or a derived physical quantity. The MCMC estimate of its posterior mean is

$$\bar{f} = \frac{1}{N} \sum_{n=1}^{N} f_n. \tag{13}$$

If the f_n were independent, then

$$\mathrm{Var}(\bar{f}) = \frac{1}{N} \mathrm{Var}(f). \tag{14}$$

But MCMC samples are correlated. If f_n is above the mean, then f_{n+1} and f_{n+2} are also likely to be above the mean. The sample average therefore fluctuates more than it would for independent samples.

5 Autocorrelation Function

The autocovariance function is

$$C_f(t) = \langle (f_n - \langle f \rangle)\,(f_{n+t} - \langle f \rangle) \rangle\,, \tag{15}$$

where t is the lag. At zero lag,

$$C_f(0) = \mathrm{Var}(f). \tag{16}$$

The normalized autocorrelation function is

$$\rho_f(t) = \frac{C_f(t)}{C_f(0)}. \tag{17}$$

If $\rho_f(t) \approx 1$, then samples separated by t steps are still strongly correlated. If $\rho_f(t) \approx 0$, then samples separated by t steps are nearly independent.

For a finite chain, emcee estimates the autocovariance using

$$\hat{c}_f(\tau) = \frac{1}{N-\tau} \sum_{n=1}^{N-\tau} (f_n - \mu_f)\,(f_{n+\tau} - \mu_f)\,, \tag{18}$$

where

$$\mu_f = \frac{1}{N} \sum_{n=1}^{N} f_n. \tag{19}$$

The normalized estimate is

$$\hat{\rho}_f(\tau) = \frac{\hat{c}_f(\tau)}{\hat{c}_f(0)}. \tag{20}$$

6 Integrated Autocorrelation Time

The integrated autocorrelation time is the sum of the normalized autocorrelation function over all lags. In the emcee convention,

$$\tau_f = \sum_{\tau=-\infty}^{\infty} \rho_f(\tau). \tag{21}$$

Since $\rho_f(-\tau) = \rho_f(\tau)$, this is

$$\tau_f = 1 + 2\sum_{\tau=1}^{\infty} \rho_f(\tau). \tag{22}$$

This number tells us how many MCMC steps correspond to one effectively independent sample. If τ_f is large, the chain remembers its previous state for many steps. If τ_f is small, the chain forgets quickly.

For correlated MCMC samples, the variance of the sample mean becomes approximately

$$\mathrm{Var}(\bar{f}) \approx \frac{\tau_f}{N} \mathrm{Var}(f). \tag{23}$$

This should be compared with the independent-sample result,

$$\mathrm{Var}(\bar{f}) = \frac{1}{N} \mathrm{Var}(f). \tag{24}$$

Thus an MCMC chain with N samples behaves roughly like

$$N_{\text{eff}} \sim \frac{N}{\tau_f} \tag{25}$$

independent samples.

The raw number of samples is not the number of independent samples. If $\tau_f = 100$, then 10^5 MCMC samples contain roughly the same information as 10^3 independent samples.

7 A Factor-of-Two Convention

Different references use slightly different definitions of integrated autocorrelation time. Some define

$$\tau_{\text{int}} = \frac{1}{2} + \sum_{\tau=1}^{\infty} \rho_f(\tau). \tag{26}$$

With this convention,

$$\text{Var}(\bar{f}) \approx \frac{2\tau_{\text{int}}}{N} \text{Var}(f). \tag{27}$$

The `emcee` convention instead uses

$$\tau_f = 2\tau_{\text{int}} = 1 + 2\sum_{\tau=1}^{\infty} \rho_f(\tau). \tag{28}$$

In these notes, τ_f always refers to the `emcee` convention.

8 Monte Carlo Error and Posterior Uncertainty

Autocorrelation time is about numerical accuracy, not physical uncertainty.

The posterior uncertainty is the width of the posterior distribution. For example, if θ_k has posterior standard deviation σ_{θ_k}, that width is determined by the data, the model, and the prior. Running the chain longer does not make the posterior narrower.

The Monte Carlo error is the numerical error in our estimate of a posterior quantity. If we estimate the posterior mean of θ_k, then

$$\text{MCSE}(\bar{\theta}_k) \approx \sqrt{\frac{\tau_{\theta_k}}{N}}\, \sigma_{\theta_k}. \tag{29}$$

Running the chain longer reduces this numerical error.

For example, suppose

$$\tau_f \sim 500. \tag{30}$$

If we want the Monte Carlo error in the posterior mean of f to be about one percent of the posterior standard deviation of f, then

$$\sqrt{\frac{\tau_f}{N}} \sim 0.01. \tag{31}$$

Therefore,

$$N \sim \frac{\tau_f}{(0.01)^2} \sim 5 \times 10^6. \tag{32}$$

This does not mean the parameter is measured to one percent. It means the numerical error in the estimated posterior mean is one percent of the posterior standard deviation.

Posterior uncertainty tells us how uncertain the parameter is given the data. Monte Carlo error tells us how accurately the finite chain estimates the posterior. These are different quantities.

9 How emcee Estimates Autocorrelation Time

In theory,

$$\tau_f = 1 + 2\sum_{\tau=1}^{\infty} \rho_f(\tau). \tag{33}$$

In practice, we only have a finite chain. The estimated autocorrelation function becomes noisy at large lag. If we sum all the way to the end of the chain, the large-lag noise can dominate the estimate.

Therefore emcee estimates

$$\hat{\tau}_f(M) = 1 + 2\sum_{\tau=1}^{M} \hat{\rho}_f(\tau) \tag{34}$$

using a finite window M.

The choice of M is a bias-variance compromise. If M is too small, we cut off real correlation and underestimate τ_f. If M is too large, we include mostly noise and make the estimate unstable.

A commonly used windowing rule is to choose the smallest M such that

$$M \geq C\hat{\tau}_f(M), \tag{35}$$

where C is a constant of order a few. This says that we sum the autocorrelation function out to several estimated autocorrelation times, but not all the way to the noisy end of the chain.

10 The emcee Chain-Length Criterion

The autocorrelation time is itself estimated from the chain. This estimate is only reliable if the chain is much longer than the autocorrelation time.

If the chain length is only

$$N \sim 10\tau_f, \tag{36}$$

then the estimate of τ_f can be unstable. A more practical condition is

$$N_{\mathrm{step}} \gtrsim 50\tau_f. \tag{37}$$

Here N_{step} means the number of steps per walker. This is important. For an ensemble sampler,

$$N_{\mathrm{flat}} = N_{\mathrm{walker}} N_{\mathrm{step}} \tag{38}$$

can be very large, but the autocorrelation time is still measured in steps. The walkers help reduce the noise in estimating τ_f, but they do not make all flattened samples independent.

For emcee, the relevant convergence condition is a condition on the number of steps per walker, not just the total number of flattened samples.

11 Interpreting the Example Figures

Figure 1 compares the true autocorrelation function with estimates from finite chains of different lengths.

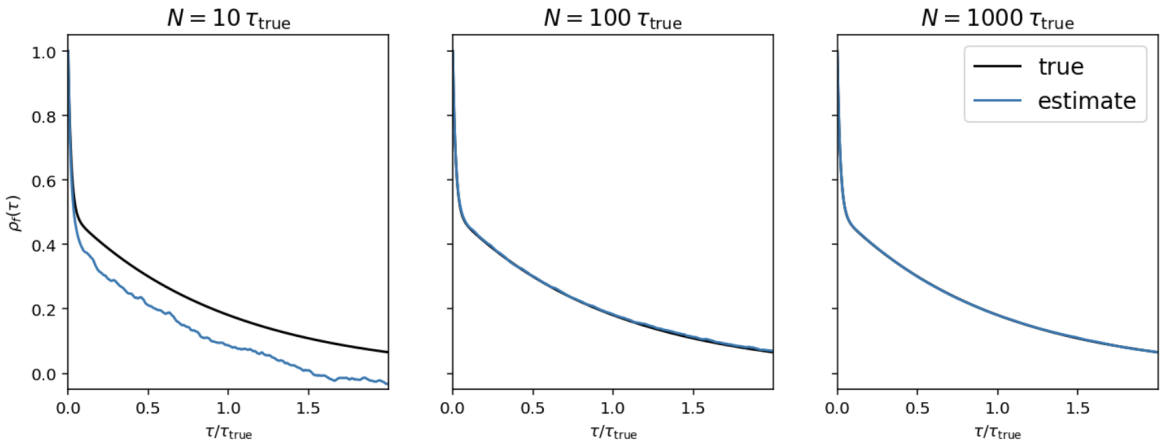

Figure 1: Estimated autocorrelation function compared to the true autocorrelation function. When the chain is only about $10\tau_{\mathrm{true}}$ long, the estimate is noisy and biased. When the chain is about $100\tau_{\mathrm{true}}$ long, the estimate is much better. By about $1000\tau_{\mathrm{true}}$, the estimated autocorrelation function closely follows the true one.

The lesson is that autocorrelation time cannot be reliably estimated from a chain that is too short. A chain needs to contain many autocorrelation times before τ_f itself becomes stable.

Figure 2 shows autocorrelation-time estimates as a function of chain length.

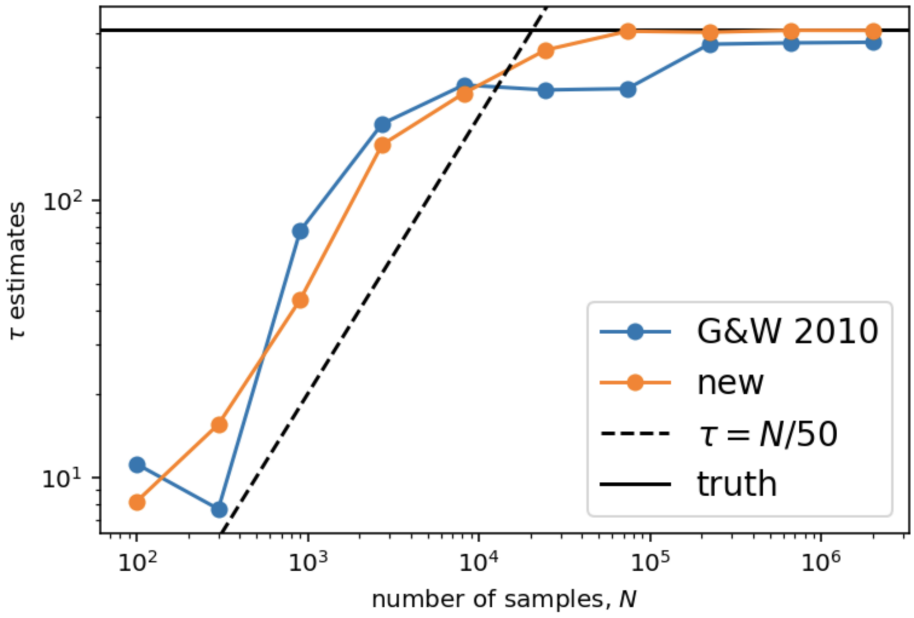

Figure 2: Autocorrelation-time estimates as a function of chain length. The dashed line shows $\tau = N/50$, and the horizontal line shows the true autocorrelation time. The condition $N \gtrsim 50\tau$ asks that the chain be many autocorrelation times long before trusting the estimate.

The dashed line

$$\tau = \frac{N}{50} \tag{39}$$

corresponds to

$$N = 50\tau. \tag{40}$$

When the estimated autocorrelation time is below this line, the chain is at least fifty autocorrelation times long. This is not a proof of convergence, but it is a useful practical diagnostic.

12 Practical Use

A reasonable workflow is:

1. Initialize walkers in a physically reasonable part of parameter space.
2. Run the sampler.
3. Inspect trace plots and the log probability.

4. Discard burn-in.
5. Estimate autocorrelation times from the post-burn-in chain.
6. Check whether the production chain has length $N_{\mathrm{step}} \gtrsim 50\tau_f$.
7. Use the post-burn-in samples for posterior summaries.

In practice, one usually computes τ_f for each parameter and uses the largest value,

$$\tau_{\max} = \max_k \tau_{\theta_k}. \tag{41}$$

The slowest-mixing parameter controls the required chain length.

Trace plots are still necessary. Autocorrelation time is not a complete convergence proof. A chain can have a finite estimated autocorrelation time but still have bad initialization, stuck walkers, or unresolved multimodality. The autocorrelation diagnostic should be used together with trace plots, log-probability evolution, and physical judgment about the model.

13 Summary

emcee samples a posterior distribution using an ensemble of walkers. The walkers move together: each walker proposes moves using the positions of other walkers. This makes the sampler effective for stretched or tilted posterior distributions.

The samples are still correlated in time. Autocorrelation time measures how many steps the chain needs to forget its previous state. If the autocorrelation time is τ_f, then N MCMC samples contain roughly

$$N_{\mathrm{eff}} \sim \frac{N}{\tau_f} \tag{42}$$

independent samples.

This is why convergence is not just about producing many samples. The chain must be long compared to its autocorrelation time, and for emcee the usual practical condition is

$$N_{\mathrm{step}} \gtrsim 50\tau_f. \tag{43}$$

A Derivation of the Autocorrelation Correction

Let

$$\bar{f} = \frac{1}{N}\sum_{i=1}^{N} f_i. \tag{44}$$

Then

$$\mathrm{Var}(\bar{f}) = \mathrm{Var}\left(\frac{1}{N}\sum_{i=1}^{N} f_i\right) \tag{45}$$

$$= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{Cov}(f_i, f_j). \tag{46}$$

For a stationary chain, the covariance depends only on the lag $t = |i - j|$:

$$\mathrm{Cov}(f_i, f_j) = C_f(|i - j|). \tag{47}$$

Therefore,

$$\mathrm{Var}(\bar{f}) = \frac{1}{N^2}\left[NC_f(0) + 2\sum_{t=1}^{N-1}(N - t)C_f(t)\right]. \tag{48}$$

Dividing and multiplying by $C_f(0)$ gives

$$\mathrm{Var}(\bar{f}) = \frac{C_f(0)}{N}\left[1 + 2\sum_{t=1}^{N-1}\left(1 - \frac{t}{N}\right)\rho_f(t)\right]. \tag{49}$$

Since $C_f(0) = \mathrm{Var}(f)$,

$$\mathrm{Var}(\bar{f}) = \frac{\mathrm{Var}(f)}{N}\left[1 + 2\sum_{t=1}^{N-1}\left(1 - \frac{t}{N}\right)\rho_f(t)\right]. \tag{50}$$

For a long chain, if the autocorrelation decays before t becomes comparable to N, then

$$1 - \frac{t}{N} \approx 1 \tag{51}$$

over the part of the sum that matters. Thus

$$\mathrm{Var}(\bar{f}) \approx \frac{\mathrm{Var}(f)}{N}\left[1 + 2\sum_{t=1}^{\infty}\rho_f(t)\right]. \tag{52}$$

Using

$$\tau_f = 1 + 2\sum_{t=1}^{\infty}\rho_f(t), \tag{53}$$

we obtain

$$\mathrm{Var}(\bar{f}) \approx \frac{\tau_f}{N}\mathrm{Var}(f). \tag{54}$$

References

- Goodman, J., & Weare, J. (2010), "Ensemble samplers with affine invariance," *Communications in Applied Mathematics and Computational Science*, 5, 65–80.
- Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. (2013), "emcee: The MCMC Hammer," *Publications of the Astronomical Society of the Pacific*, 125, 306–312. https://arxiv.org/abs/1202.3665
- emcee documentation, "Autocorrelation analysis & convergence." https://emcee.readthedocs.io/en/stable/tutorials/autocorr/
- Berg, B. A. (2008), "Markov Chain Monte Carlo Simulations and Their Statistical Analysis." Lecture notes, Chapter 7. http://www.hep.fsu.edu/~berg/teach/mcmc08/material/lecture07mcmc3.pdf
- Sokal, A. D. (1997), "Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms." Lecture notes, Cargèse Summer School.