STSCI 4780
Bayesian computation:
MCMC output analysis

Tom Loredo, CCAPS & SDS, Cornell University

2020-03-12

Numerical computation as probable inference

"Bayesian numerical analysis," Persi Diaconis (1988)

1. INTRODUCTION

Consider a given function $f: [0,1] \to \mathbf{R}$ such as

$$f(x) = \exp\left\{\cosh\left(\frac{x + 2x^2 + \cos x}{3 + \sin x^3}\right)\right\}. \tag{1}$$

If you require $\int_0^1 f(x)dx$, a formula such as (1) isn't of much use and leads to questions like "What does it mean to 'know' a function?" The formula says some things (e.g. f is smooth, positive, and bounded by 20 on $[0,1]$) but there are many other facts about f that we don't know (e.g., is f monotone, unimodal, or convex?).

Once we allow that we don't know f, but do know some things, it becomes natural to take a Bayesian approach to the quadrature problem:

- Put a prior on continuous functions $C[0,1]$
- Calculate f at $x_1, x_2, \ldots, x_n$
- Compute a posterior
- Estimate $\int_0^1 f$ by the Bayes rule

Most people, even Bayesians, think this sounds crazy when they first hear about it. The following examples may help.

Stochastic process terminology

Stochastic process: A probabilistic model of a process evolving/developing in time and/or space (any type of space—3D space, energy, wavelength, on a sphere. . .)

- Index set: The set labeling locations in time and/or space — integer time, continuous time, Cartesian grid, latitude & longitude. . .
- State space: The possible values of the process (duplicated for each choice of index) — heads/tails, price, luminosity, concentration, velocity (vector). . .

SP (formally): A *joint dist'n* (or family of dist'ns, e.g., for different numbers of indices) over indexed copies of a state space, or a set of *rules for building such joint distributions*

SPs are special joint distributions, with every variable representing the same type of quantity, specified via extendible rules

Processes and paths (realizations)

- *Stochastic process:* The joint distribution
- *Sample path or realization:* One sample from a stochastic process — a *time series* or *field* of specific state values over a set of indices

Bernoulli: Indices are times (trials #s) $t = 1, 2, \ldots$; states are binary outcomes $o = 0$ or1

- Bernoulli process: $P(o_1, o_2, \ldots) = \prod_i \alpha^{o_i}(1-\alpha)^{1-o_i}$
- Bernoulli sample path: 001011100100011... (binary sequence)

Poisson: Indices are non-negative real times, states are natural numbers $n(l, u)$, the event count in time interval $[l, u]$

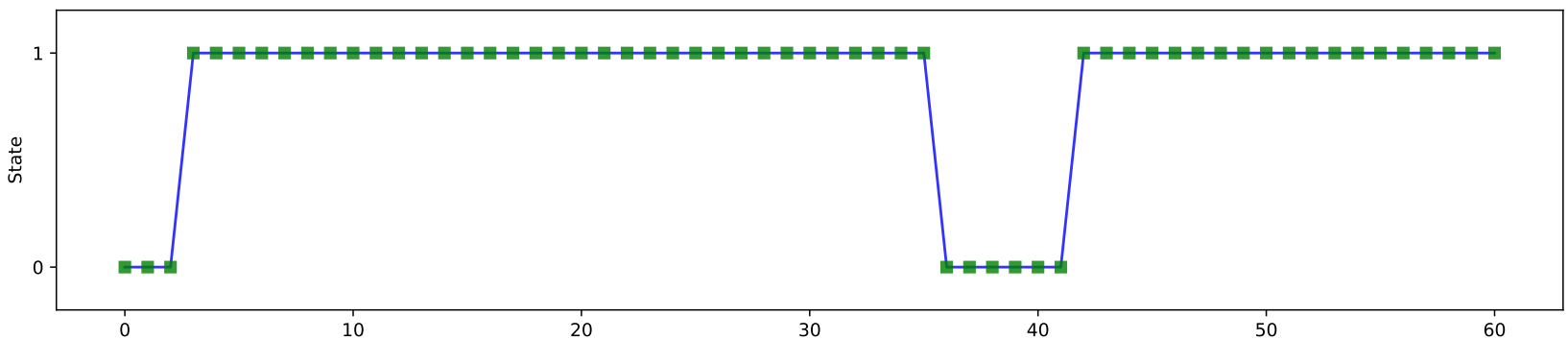
- Poisson process: Rule for $P(n(l_1, u_1), n(l_2, u_2), \ldots)$ for any set of intervals
- Poisson sample path: A particular set of discrete, separate points at times $t \in [0, T]$ (T may be ∞); this defines the states

2-state discrete stationary Markov process: Indices are natural number times, states are binary outcomes $\theta_i = 0$ or 1

- Markov chain:

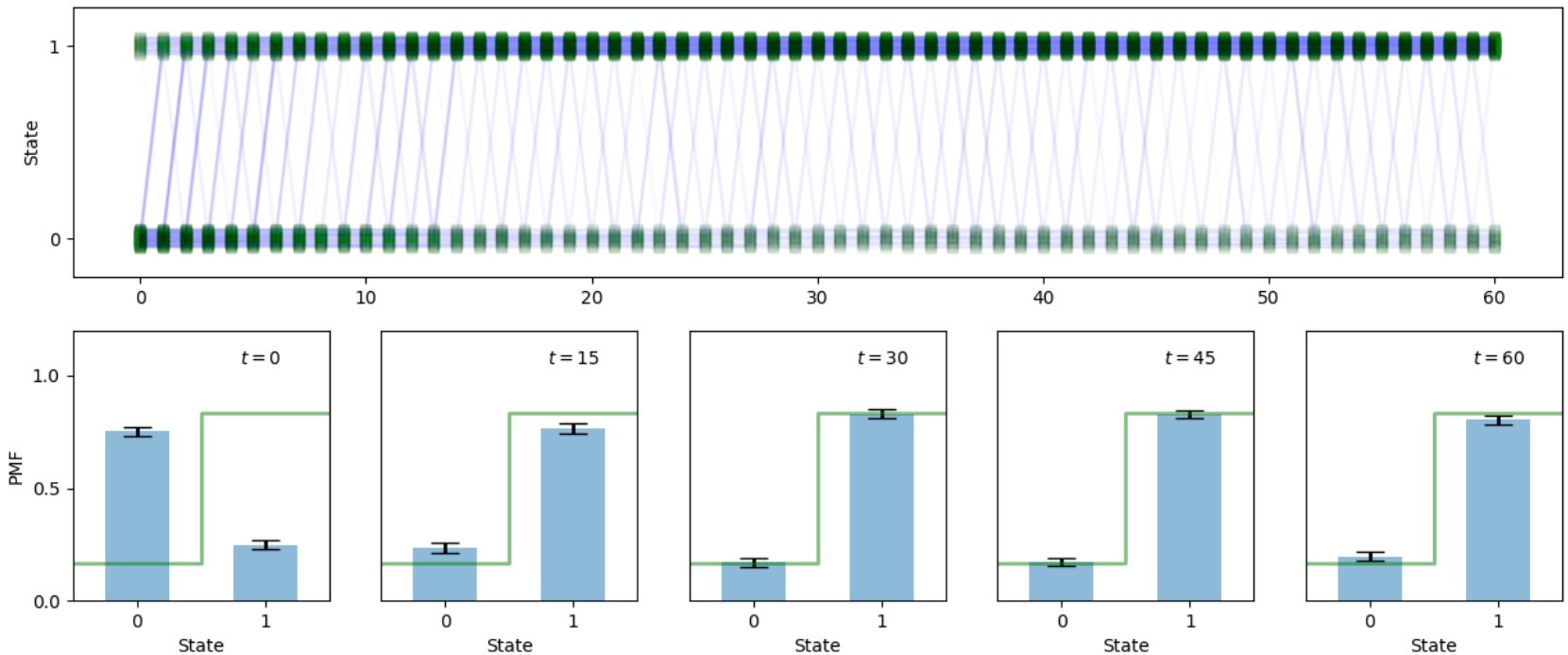
$$p(\theta_0, \theta_1, \theta_2, \ldots) = p(\theta_0) \times T(\theta_1|\theta_0) \times T(\theta_2|\theta_1) \times \cdots$$

- Markov chain sample path for $T_{10} = 0.1,\ T_{01} = 0.02$:



In general, to learn about a SP we need to have many realizations:

400 2-state Markov chain paths
p_{eq} (green histo), estimated marginals (blue histos)



If we didn't know p_{eq}, we could estimate it (or any of its properties — mean, variance, probability for an interval) by making a late-time histogram using many sample paths

Bernoulli process suggests it is sometimes possible to learn about a process from a *single* sample path — IID *replication* makes this possible

Ergodic SP: A SP for which some/all properties may be learned from a single, long sample path (all ESPs are stationary)

Technical definition: Time average (along a single path) $\approx$ Expectation (wrt. p_{eq})

Stationary Markov processes with equilibrium dist'ns (irreducible, aperiodic) are ergodic!

Stationarity + limited memory $\rightarrow$ there is enough replication along a long sample path to learn properties of p_{eq}

The Good News

The Metropolis-Hastings algorithm enables us to draw a few time series realizations (sample paths) $\{\theta_t\}$, $t = 0$ to N, from a Markov chain with a specified stationary distribution $p(\theta)$

The algorithm works for any $f(\theta) \propto p(\theta)$, i.e., *$Z$ needn't be known*

Denote the marginal distribution at each time as $p_t(\theta)$

- *Stationarity*: If $p_0(\theta) = p(\theta)$, then $p_t(\theta) = p(\theta)$
- *Convergence to equil'm*: If $p_0(\theta) \neq p(\theta)$, eventually

  $$||p_t(\theta), p(\theta)|| < \epsilon$$

  for an appropriate norm between distributions
- *Ergodicity*:

  $$\bar{g} \equiv \frac{1}{N} \sum_t g(\theta_t) \to \langle g \rangle \equiv \int d\theta \; g(\theta) p(\theta)$$

  long-enough time averages = posterior expectations

The Bad News

- We never have $p_0(\theta) = p(\theta)$: we have to figure out how to initialize a realization, and we are always in the situation where $p_t(\theta) \neq p(\theta)$ (but hopefully close)
- "Eventually" means $t < \infty$; that's not very comforting!
- After convergence at time $t = t_c$, $p_t(\theta) \approx p(\theta)$, but θ values at different times are *dependent*, so the simple IID behavior, expected MSE $= \sigma^2/N$, doesn't hold
- We have to learn about $p_t(\theta)$ (or expectations over it) from just a few time series realizations (maybe just one)

MCMC output analysis

Diagnostics

Posterior sample diagnostics use *single* chains, $\{\theta_t\}$, or *multiple* chains, $\{\theta_{tc}\}$, to diagnose:

- **Initialization bias:** How long until starting values are forgotten? (Discard initial *burn-in* segment or run long enough so averages "forget" initialization bias)
- **Mixing:** How quickly/efficiently are we sampling the full posterior? (Make finite-sample Monte Carlo uncertainties small)

Estimation & summarization

How should we use MCMC output to estimate posterior expectations, with uncertainty quantification that accounts for dependence of samples?

Outputs: means, variances, marginals (1-D and 2-D), HPD regions, tabulation/visualization...

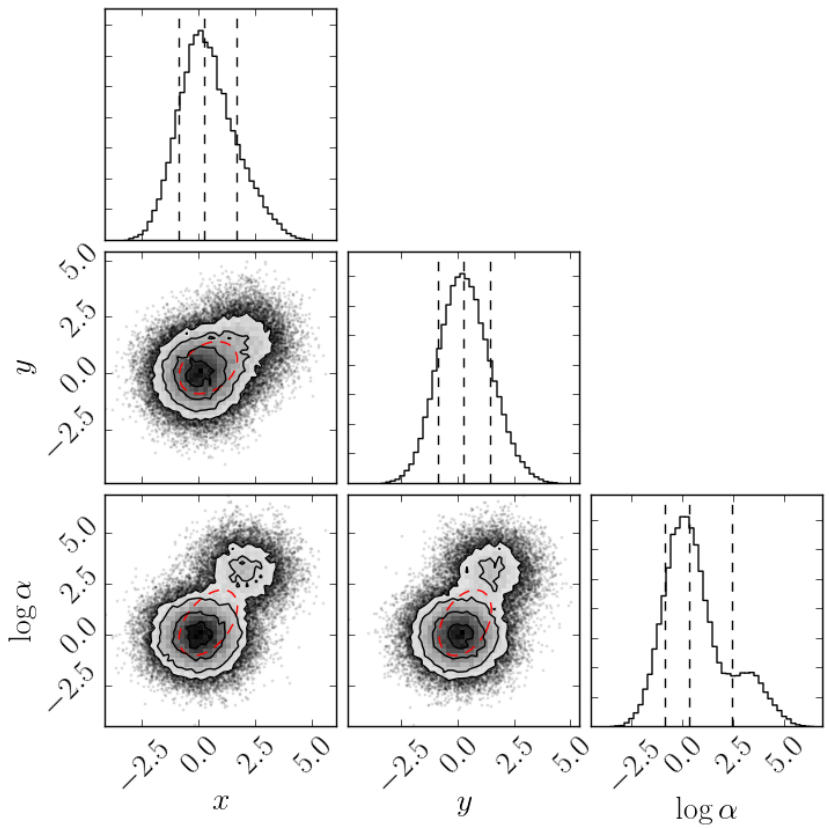
Common MCMC estimators

Posterior means & standard deviations

Use *sample averages*: $\langle \theta \rangle = \sum_t \theta_t$

1-D and 2-D marginals

Marginalization is *just sample projection*!
Use histograms, KDE, pair/corner plots, scatterplot matrix



Estimation with marginally ID samples

Recall definitions:

$$\text{Posterior expectation } \mu \equiv \int d\theta\, g(\theta)p(\theta)$$

$$\text{Sample mean } m \equiv \bar{g} = \frac{1}{N}\sum_t g_t \quad \text{with } g_t \equiv g(\theta_t)$$

For *dependent* samples with identical marginals:

$$\langle m \rangle \equiv \mathbb{E}(m) = \mu$$

Error from a particular sample is

$$m - \mu = \frac{1}{N}\sum_t g_t - \mu = \frac{1}{N}\sum_t (g_t - \mu)$$

Then the *expected mean-squared error* (EMSE) is:

$$\left\langle (m-\mu)^2 \right\rangle = \frac{1}{N^2}\left\langle \left[\sum_t (g_t - \mu)\right] \times \left[\sum_s (g_s - \mu)\right] \right\rangle$$

Let $\sigma^2 \equiv \left\langle (g_t - \mu)^2 \right\rangle$ (ind. of t!); then

$$\begin{aligned} \left\langle (m-\mu)^2 \right\rangle &= \frac{\sigma^2}{N}\left[1 + \frac{2}{N}\sum_{t=1}^{N}\sum_{s=t+1}^{N}\left\langle \frac{g_t-\mu}{\sigma}\cdot\frac{g_s-\mu}{\sigma}\right\rangle\right] \\ &= \text{IID EMSE} \times \text{autocorrelation factor} \end{aligned}$$

Stationarity further implies

$$\begin{aligned} \left\langle (g_t-\mu)\cdot(g_s-\mu)\right\rangle &= \left\langle (g_{t+\delta}-\mu)\cdot(g_{s+\delta}-\mu)\right\rangle \\ &\equiv C_{|t-s|}, \text{ (using autocovariance)} \\ &\equiv \sigma^2\rho_{|t-s|}, \text{ (using autocorrelation)} \end{aligned}$$

Effective sample size ESS or N_{eff} is defined so

$$\left\langle (m-\mu)^2 \right\rangle = \frac{\sigma^2}{N_{\mathrm{eff}}}$$

Markov chain CLT

The Markov chain CLT says that, in equillibrium (asymptotically),

$$\bar{g} \sim N(\langle g \rangle, \sigma_g^2)$$

$$\sigma_g^2 = \left\langle (m - \mu)^2 \right\rangle$$

Note: All of these results are *expectations*, using the 1-D and 2-D marginal PDFs for the Markov process

In calculations, we don't know any PDFs, and we have to estimate expectations with sample averages or time series techniques

Simpler expedients (details later):

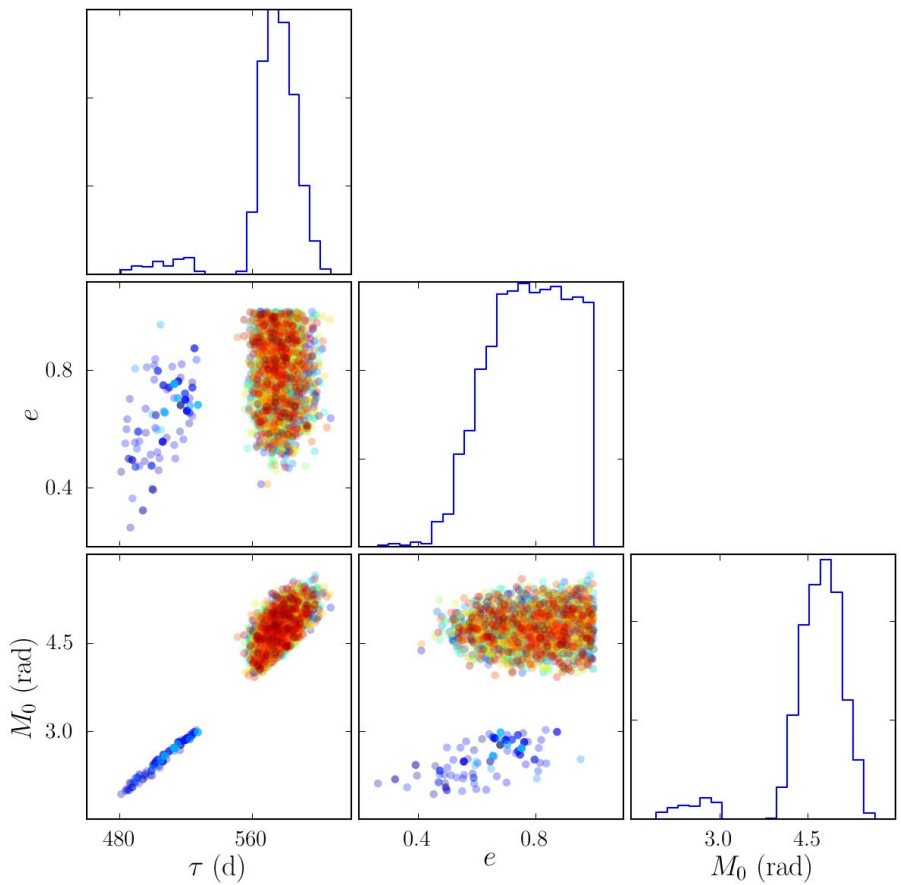
- Thin the chain to size N_{eff}; treat as independent samples
- Consistent batch means—uses full chain

Diagnosing convergence

Qualitative/visual

- Trace plots—trends indicate initialization bias
- Diagnostic plots; e.g., running mean should converge
- Color-coded pair plots

Exoplanet parameter estimation using RV data from HD 222582 and Ter Braak's differential evolution MCMC



Quantitative

- Gelman-Rubin-Brooks potential scale-reduction statistic $\sqrt{R}$: *multiple* chains, compare within- and between-chain variance
- Geweke: single chain, consistency of early/late means
- Heidelberger & Welch: single chain, checks for Brownian motion signature of stationarity (root-N growth of accumulated motion), estimates burn-in
- Fan-Brooks-Gelman score statistic:

$$U_k(\theta) = \frac{\partial \log p(\theta)}{\partial \theta_k}$$

Uses $\langle U_k \rangle_p = 0$ (but requires derivatives)

Use diagnostics for all quantities of interest!
Check all parameters, and functions of them

Diagnosing mixing

Qualitative/visual

- Trace plots—does chain get stuck, have slow trends?
- Diagnostic plots; e.g., running mean, *sample (path) autocorrelation function* (ACF)

Quantitative

Use estimators with uncertainties that account for dependence:

- Estimate expected MSE using ACF to estimate covariances
- Use ACF to estimate ESS; use thinned chain to compute results
- Consistent batch means
- AR and spectral analysis estimators

Autocorrelation

Recall the *expected MSE*

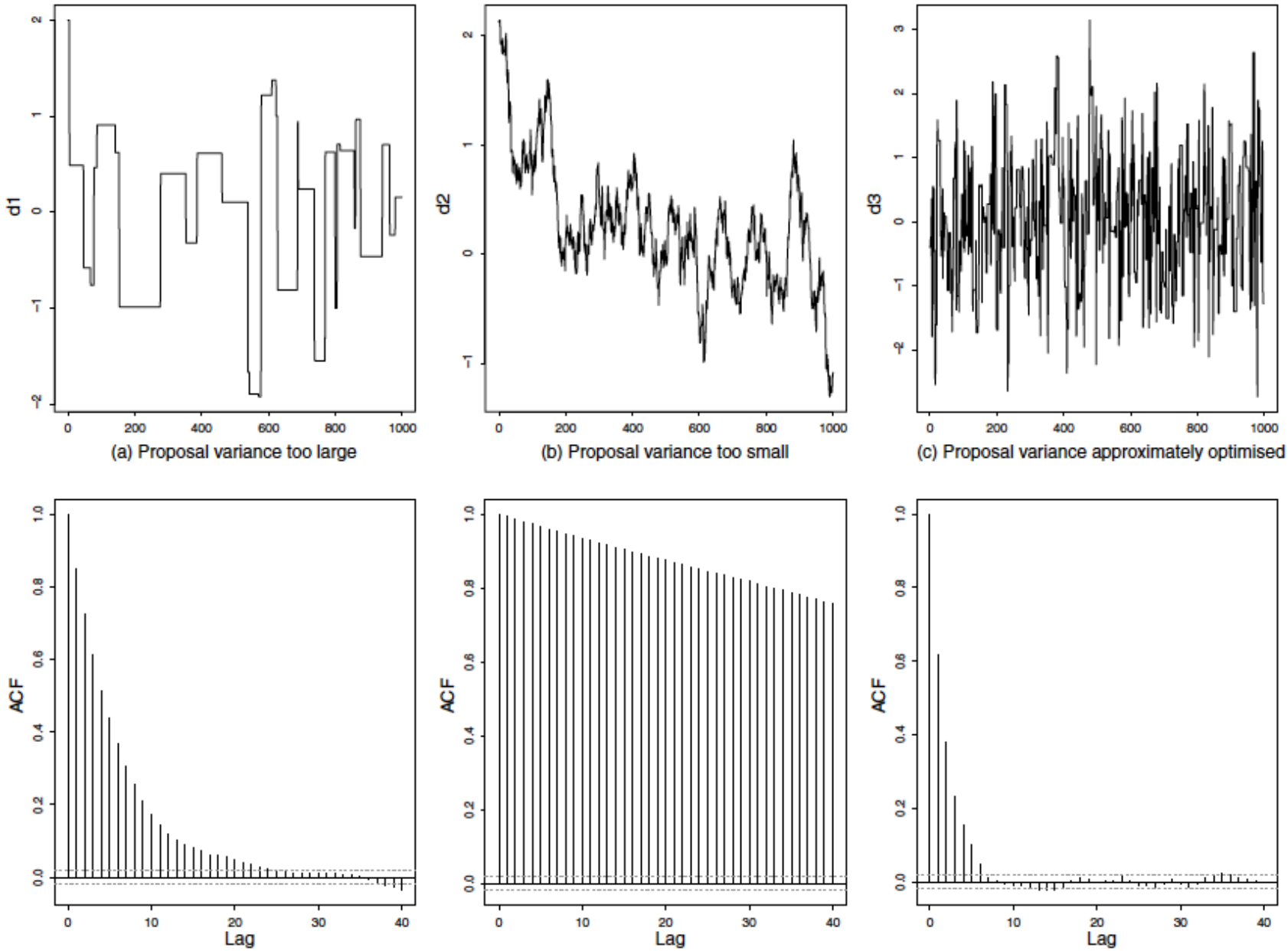
$$\begin{aligned}\left\langle (m-\mu)^2 \right\rangle &= \frac{\sigma^2}{N}\left[1 + \frac{2}{N}\sum_{t=1}^{N}\sum_{s=t+1}^{N}\left\langle \frac{g_t-\mu}{\sigma}\cdot\frac{g_s-\mu}{\sigma}\right\rangle\right] \\ &= \frac{\sigma^2}{N}[1+\rho_{|t-s|}] \quad \text{from stationarity} \\ &= \text{IID EMSE} \times \text{autocorrelation factor}\end{aligned}$$

Estimate ρ_l at *lag* l via *sample (path) ACF*:

$$\begin{aligned}\rho_l &\equiv \langle (g_t-\mu)\cdot(g_{t-l}-\mu)\rangle \\ &\approx \frac{1}{(N-l)s^2}\sum_{t=l+1}^{N}(g_t-\bar{g})(g_{t-l}-\bar{g})\end{aligned}$$

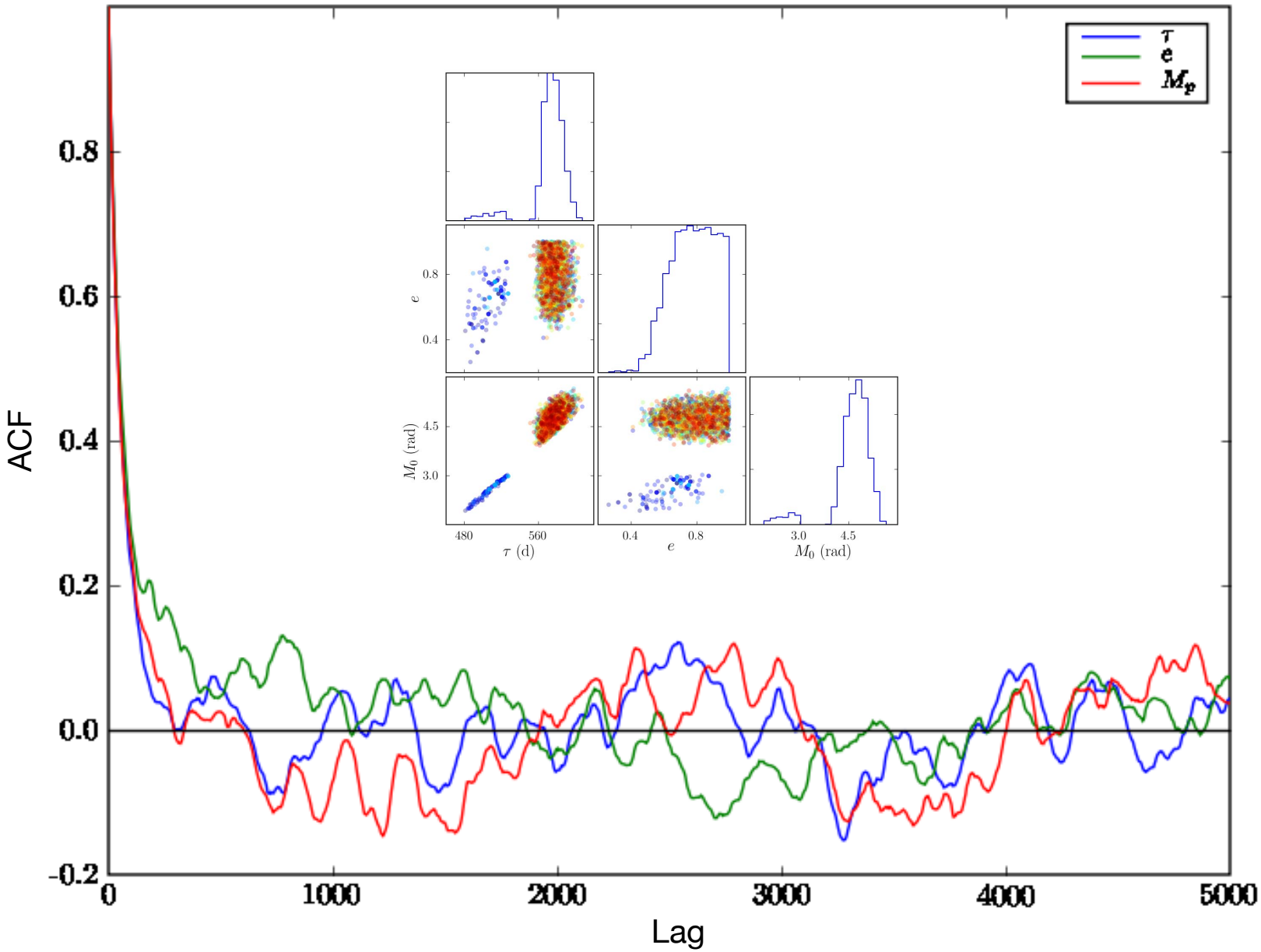
with $s^2 = \frac{1}{N}\sum_t (g_t-\bar{g})^2$

For long chains, can use Fourier (periodogram) methods to quickly compute ACF



Roberts & Rosenthal (2001)

HD 222582 exoplanet example



Consistent batch means

Write $N = Bn$ for B batches of size n; each batch has sample mean

$$\bar{y}_b = \frac{1}{n} \sum_{t=(b-1)n+1}^{bn} g_t$$

Estimate the uncertainty for the estimate $\bar{g}$ by

$$\hat{\sigma}_g^2 = \frac{n}{B-1} \sum_{b=1}^{B} (\bar{y}_b - \bar{g})^2$$

For the estimate to be consistent, must let the batch size and the number of batches increase with N:

$$\begin{aligned} n &\approx N^{1/2} \\ B &= \lfloor N/n \rfloor \end{aligned}$$

Software

Most packages include descriptions of methods and references:

Python packages:

- **PyMC**
  `http://pymc-devs.github.io/pymc/`
  Includes support for exporting MCMC data for R's **coda**
- **triangle**
  `https://github.com/dfm/triangle.py`
  Attractive pair plots
- **seaborn**
  `https://seaborn.pydata.org/generated/seaborn.pairplot.html`
  Attractive pair plots

R packages (more extensive):

- **boa**
  `http://cran.r-project.org/web/packages/boa/index.html`
- **coda**
  `http://cran.r-project.org/web/packages/coda/index.html`
- **batchmeans**
  `http://cran.r-project.org/web/packages/batchmeans/index.html`
- **bayesplot**
  `https://cran.r-project.org/web/packages/bayesplot/`, Stan page

Bayesian Inference and the Joint Distribution

Recall that Bayes's theorem comes from the *joint distribution for data and hypotheses* (parameters/models):

$$\begin{aligned} p(\theta, D|M) &= p(\theta|M)\, p(D|\theta, M) \\ &= p(D|M)\, p(\theta|D, M) \end{aligned}$$

Bayesian inference takes $D = D_{\text{obs}}$ and solves RHS for the posterior:

$$\rightarrow p(\theta|D_{\text{obs}}, M) = \frac{p(\theta|M)p(D_{\text{obs}}|\theta, M)}{p(D_{\text{obs}}|M)}$$

MCMC is nontrivial technology for building RNGs to sample θ values from the *intractable posterior*, $p(\theta|D_{\text{obs}}, M)$

Posterior sampling is hard, but sampling from the other distributions is often easy:

- Often easy to draw θ^* from $\pi(\theta)$
- Typically easy to draw D_{sim} from $p(D|\theta, M)$
- Thus we can sample the joint for (θ, D) by sequencing:

$$\theta^* \sim \pi(\theta)$$
$$D_{\text{sim}} \sim p(D|\theta^*, M)$$

- $\{D_{\text{sim}}\}$ from above are samples from prior predictive,

$$p(D|M) = \int d\theta\, \pi(\theta) p(D|\theta, M)$$

Now note that $\{D_{\text{sim}}, \theta\}$ with $\theta \sim p(\theta|D_{\text{sim}}, M)$ (via MCMC) are also samples from the joint distribution

Joint distribution methods check the consistency of these two joint samplers to validate a posterior sampler implementation

Example: "Calibration" of credible regions

How often may we expect an HPD region with probability P to include the true value if we analyze many datasets? I.e., what's the frequentist coverage of an interval rule $\Delta(D)$ defined by calculating the Bayesian HPD region each time?

Suppose we generate datasets by picking a parameter value from $\pi(\theta)$ and simulating data from $p(D|\theta)$

The fraction of time θ will be in the HPD region is:

$$Q = \int d\theta\, \pi(\theta) \int dD\, p(D|\theta)\, [\![\theta \in \Delta(D)]\!]$$

Note $\pi(\theta)p(D|\theta) = p(\theta, D) = p(D)p(\theta|D)$, so

$$Q = \int dD \int d\theta\, p(\theta|D)\, p(D)\, [\![\theta \in \Delta(D)]\!]$$

$$
\begin{aligned}
Q &= \int dD \int d\theta \; p(\theta|D)\, p(D) \; [\![\theta \in \Delta(D)]\!] \\
&= \int dD \; p(D) \int d\theta \; p(\theta|D) \; [\![\theta \in \Delta(D)]\!] \\
&= \int dD \; p(D) \int_{\Delta(D)} d\theta \; p(\theta|D) \\
&= \int dD \; p(D) P \\
&= P
\end{aligned}
$$

The HPD region includes the true parameters $100P\%$ of the time

This is exactly true for any problem, even for small datasets

Keep in mind it involves drawing θ from the prior; credible regions are "calibrated with respect to the prior"

A Tangent: Average Coverage

Recall the original Q integral:

$$\begin{aligned} Q &= \int d\theta\, \pi(\theta) \int dD\, p(D|\theta)\, [\![\theta \in \Delta(D)]\!] \\ &= \int d\theta\, \pi(\theta) C(\theta) \end{aligned}$$

where $C(\theta)$ is the (frequentist) coverage of the HPD region when the data are generated using θ

This indicates Bayesian regions have accurate *average coverage*

The prior can be interpreted as quantifying how much we care about coverage in different parts of the parameter space

Basic Bayesian Calibration Diagnostics

Encapsulate your sampler: Create an MCMC posterior sampling algorithm for model M that takes data D as input and produces posterior samples $\{\theta_i\}$, and a $100\,P\%$ credible region $\Delta_P(D)$

Initialize counter $Q = 0$
Repeat $N \gg 1$ times:

1. Sample a "true" parameter value θ^* from $\pi(\theta)$
2. Sample a dataset D_{sim} from $p(D|\theta^*)$
3. Use the encapsulated posterior sampler to get $\Delta_P(D_{\text{sim}})$ from $p(\theta|D_{\text{sim}}, M)$
4. If $\theta^* \in \Delta_P(D)$, increment Q

Check that $Q/N \approx P$

Easily extend the idea to check *all* credible region sizes:

Initialize a list that will store N probabilities, P
Repeat $N \gg 1$ times:

1. Sample a "true" parameter value θ^* from $\pi(\theta)$
2. Sample a dataset D_{sim} from $p(D|\theta^*)$
3. Use the encapsulated posterior sampler to get $\{\theta_i\}$ from $p(\theta|D_{\text{sim}}, M)$
4. Find P so that θ^* is on the boundary of $\Delta_P(D)$; append to list $[P = \text{fraction of } \{\theta_i\} \text{ with } q(\theta_i) > q(\theta^*)]$

Check that the Ps follow a uniform distribution on $[0, 1]$

Other Joint Distribution Tests

- Geweke 2004: Calculate means of scalar functions of (θ, D) two ways; compare with z statistics
- Cook, Gelman, Rubin 2006: Posterior quantile test, expect $p[g(\theta) > g(\theta^*)] \sim$ Uniform (HPD test is special case)

What Joint Distribution Tests Accomplish

Suppose the prior and sampling distribution samplers are well-validated

- **Convergence verification:** If your posterior sampler is bug-free but was not run long enough $\rightarrow$ unlikely that inferences will be calibrated
- **Bug detection:** An incorrect posterior sampler implementation will not converge to the correct posterior distribution $\rightarrow$ unlikely that inferences will be calibrated, even if the chain converges

Cost: Prior and data sampling is often cheap, but posterior sampling is often expensive, and joint distribution tests require you run your MCMC code *hundreds* of times

Compromise: If MCMC cost grows with dataset size, running the test with small datasets provides a good bug test, and *some* insight on convergence; could also test a simplified model

Experts Speak

All the methods can fail to detect the sorts of convergence failure they were designed to identify. We recommend a combination of strategies. . . it is not possible to say with certainty that a finite sample from an MCMC algorithm is representative of an underlying stationary distribution.

— Cowles & Carlin review of 13 diagnostics

[A]ll methods based solely upon sampler output can be fooled. . . and multiple-chain-based diagnostics, while safer than single-chain-based diagnostics, can still be highly dependent upon the starting points of the simulations. . . . in practice, it may be useful to combine a number of the alternative approaches. . . .

— Brooks & Gelman 1998

In more than, say, a dozen dimensions, it is difficult to believe that a few, even well-chosen, scalar statistics give an adequate picture of convergence of the multivariate distribution.

— Peter Green 2002

Handbook of Markov Chain Monte Carlo (2011)

Your humble author has a dictum that *the least one can do is to make an overnight run*. What better way for your computer to spend its time? In many problems that are not too complicated, this is millions or billions of iterations. *If you do not make runs like that, you are simply not serious about MCMC.* Your humble author has another dictum (only slightly facetious) that one should start a run when the paper is submitted and keep running until the referees' reports arrive. This cannot delay the paper, and may detect pseudo-convergence.

— Charles Geyer

When all is done, compare inferences to those from simpler models or approximations. Examine discrepancies to see whether they represent programming errors, poor convergence, or actual changes in inferences as the model is expanded.

— Gelman & Shirley

From: *Handbook of Markov Chain Monte Carlo*