Markov Chain Monte Carlo

Brendon J. Brewer

Department of Statistics
The University of Auckland

https://www.stat.auckland.ac.nz/~brewer/

Emphasis

I will try to emphasise the underlying ideas of the methods.

I will not be teaching specific software packages (e.g. *DNest4*, *emcee*, *JAGS*, *MultiNest*, *Stan*), though I may mention them.

Ingredients I

Bayesian inference need the following inputs:

- A **hypothesis space** describing the set of possible answers to our question ("parameter space" in fitting is the same concept).
- A **prior distribution** $p(\theta)$ describing how plausible each of the possible solutions is, not taking into account the data.

Ingredients II

Bayesian inference need the following inputs:

- $p(D|\theta)$, describing our knowledge about the connection between the parameters and the data.

When D is known, this is a function of θ called the **likelihood**.

The Posterior Distribution

The data helps us by changing our prior distribution to the **posterior distribution**, given by

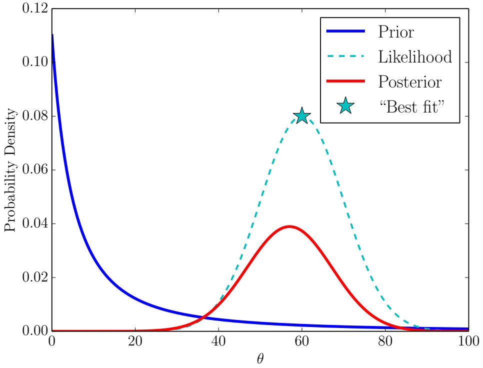
$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)}$$

where the denominator is the normalisation constant, usually called either the **marginal likelihood** or the **evidence**.

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

Posterior Distribution vs. Maximum Likelihood

The practical difference between these two concepts is greater in higher dimensional problems.

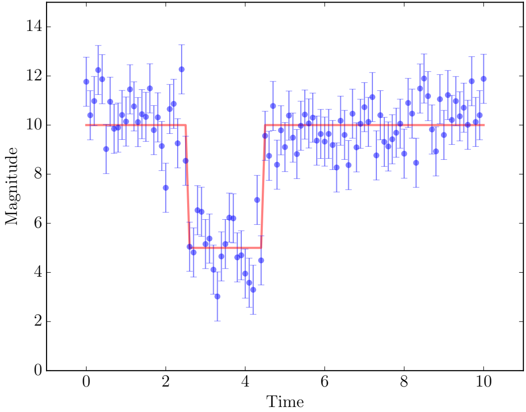


Transit Example

This example is quite simple, yet it is complex enough to demonstrate many important principles.

It is also closely related to many astronomical situations!

Transit Example



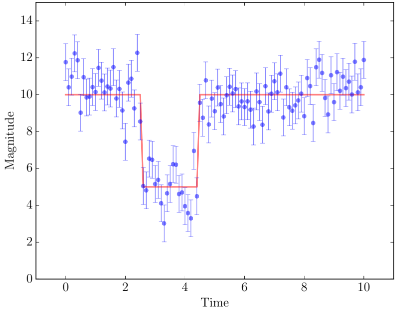
Related to the transit example...

- Realistic exoplanet transits
- Finding emission/absorption lines in spectra
- Finding stars/galaxies in an image
- ¡Y mucho más!

Transit Example: The Truth

The red curve was:

$$\mu(t) = \begin{cases} 10, & 2.5 \leq t \leq 4.5 \\ 5, & \text{otherwise.} \end{cases}$$



Transit Example: The Truth

The red curve was:

$$\mu(t) = \begin{cases} 10, & 2.5 \leq t \leq 4.5 \\ 5, & \text{otherwise.} \end{cases}$$

and the noise was added like this:

```
# Add noise
sig = 1.
y += sig*rng.randn(y.size)
```

Transit Example: Inference

Let's fit the data with this model:

$$\mu(t) = \begin{cases} A, & (t_c - w/2) \leq t \leq (t_c + w/2) \\ A - b, & \text{otherwise.} \end{cases}$$

We don't know A, b, t_c, and w. But we do know the data D.

Transit Example: Parameters

We don't know A, b, t_c, and w. These are our unknown parameters. Let's find the posterior.

$$p(A, b, t_c, w|D) = \frac{p(A, b, t_c, w)p(D|A, b, t_c, w)}{p(D)}$$

Transit Example: Problems I

The posterior is given by:

$$p(A, b, t_c, w|D) = \frac{p(A, b, t_c, w)p(D|A, b, t_c, w)}{p(D)}$$

But...

How do we choose the *prior*, $p(A, b, t_c, w)$?

How do we choose the *likelihood*, $p(D|A, b, t_c, w)$?

How do we find $p(D)$?

Choosing priors

The prior $p(A, b, t_c, w)$ describes what values are plausible, without taking the data into account.

Using the product rule, we can break this down:

$$p(A, b, t_c, w) = p(A)p(b|A)p(t_c|b, A)p(w|t_c, b, A)$$

Often, we can assume the prior factorises like this (i.e. the priors are **independent**):

$$p(A, b, t_c, w) = p(A)p(b)p(t_c)p(w)$$

Choosing priors

Often, before we get the data, we have a lot of uncertainty about the values of the parameters. That's why we wanted the data!

This motivates **vague priors**.

Uniform Priors

Let's just use wide uniform priors.

e.g.

$$p(A) = \begin{cases} \frac{1}{200}, & -100 \leq A \leq 100 \\ 0, & \text{otherwise.} \end{cases}$$

Abbreviated:

$$p(A) \sim \text{Uniform}(-100, 100)$$

Or even more concisely:

$$A \sim U(-100, 100)$$

Uniform Priors

For all four parameters:

$$
\begin{aligned}
A &\sim U(-100, 100) \\
b &\sim U(0, 10) \\
t_c &\sim U(t_{\min}, t_{\max}) \\
w &\sim U(0, t_{\max} - t_{\min})
\end{aligned}
$$

Where $t_{\min}$ and $t_{\max}$ give the time range of the dataset. Question: is this legitimate? Are we using the data to set our priors?

Sampling Distribution / Likelihood

Let's assume "gaussian noise":

$$p(y_i|A, b, t_c, w) \quad = \quad \prod_{i=1}^{N} \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left[-\frac{1}{2\sigma_i^2}\left(y_i - m(t_i; A, b, t_c, w)\right)^2\right].$$

or more concisely:

$$y_i|A, b, t_c, w \sim \mathcal{N}\left(m(t_i; A, b, t_c, w), \sigma_i^2\right).$$

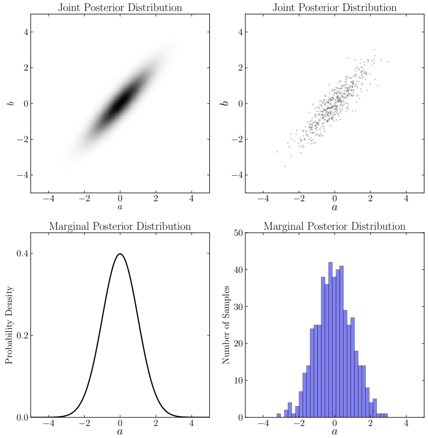
Transit Example: Problems II

Even if we can calculate the posterior $p(A, b, t_c, w|D)$, it is still a probability distribution over a four-dimensional space.

How can we understand and visualise it?

Answer to Problem II: Monte Carlo

- **Marginalisation** becomes trivial
- We can quantify all uncertainties we might be interested in



Answer to Problem II: Monte Carlo

e.g. Posterior mean of w:

$$\int w p(A, b, t_c, w|D)\, dA\, db\, dt_c\, dw \approx \frac{1}{N}\sum_{i=1}^{N} w_i \tag{1}$$

(i.e. just the arithmetic mean). Probability of being in some region R:

$$\int_R p(A, b, t_c, w|D)\, dA\, db\, dt_c\, dw \approx \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(\theta_i \in R\right) \tag{2}$$

(i.e. just the fraction of the samples in R).

Monte Carlo

Samples from the posterior are very useful, but how do we generate them?

Answer: **Markov Chain Monte Carlo**

This is not the *only* answer, but it's the most popular.

Monte Carlo

Samples from the posterior are very useful, but how do we generate them?

https://www.youtube.com/watch?v=Vv3f0QNWvWQ

The Metropolis Algorithm

- Start at some point θ in the hypothesis space.
- Loop

  $\{$

  - Generate **proposal** from some distribution $q(\theta'|\theta)$ (e.g. slightly perturb the current position).
  - With probability $\alpha = \min\left(1, \frac{p(\theta')p(D|\theta')}{p(\theta)p(D|\theta)}\right)$, accept the proposal (i.e. replace θ with θ').
  - Otherwise, stay in the same place.

  $\}$

Acceptance Probability

The full acceptance probability is

$$\alpha = \min\left(1, \frac{q(\theta|\theta')}{q(\theta'|\theta)}\frac{p(\theta')}{p(\theta)}\frac{p(D|\theta')}{p(D|\theta)}\right) \tag{3}$$

We'll usually make choices where the qs cancel out, and sometimes we'll choose the qs to also cancel out the prior ratio (easier than it sounds).

Implementing the Metropolis Algorithm

To use Metropolis on the Transit Problem, we'll need functions to:

- Generate a starting point (I like to draw the parameters from the prior)
- Make proposals
- Evaluate the prior distribution at any point
- Evaluate the likelihood at any point

Coding...

Note the use of logarithms to avoid overflow and underflow.

Random Walk Proposals

```
# Generate a proposal
L = 1.
proposal = x + L*rng.randn()
```

Problem: Efficiency depends strongly on L. The only way to know the optimal value of L is to have already solved the problem! Oh dear.

Heavy-Tailed Random Walk Proposals

```
# Generate a proposal
L = jump_size*10.**(1.5 - 6.*rng.rand())
proposal = x + L*rng.randn()
```

where `jump_size` $\approx$ prior width. Don't need steps much bigger than the prior width, may need them to be much smaller.

Acceptance Probability

The full acceptance probability is

$$\alpha = \min\left(1, \frac{q(\theta|\theta')}{q(\theta'|\theta)}\frac{p(\theta')}{p(\theta)}\frac{p(D|\theta')}{p(D|\theta)}\right) \tag{4}$$

For the random walk proposal, the q ratio is equal to 1. Do you understand why?

Proposing one parameter at a time

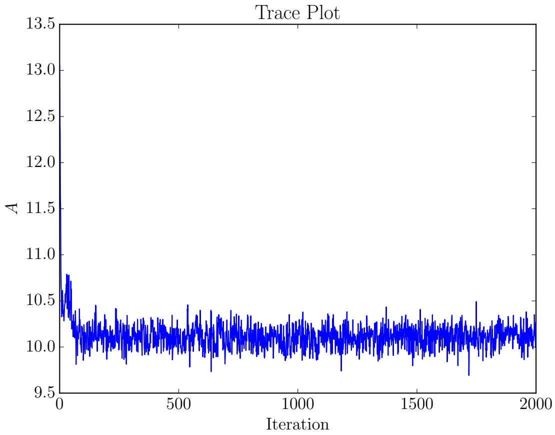
```python
def proposal(params):
  new = copy.deepcopy(params)

  which = rng.randint(num_params) # Parameter to change
  L = jump_sizes[which]*10.**(1.5 - 6.*rng.rand())
  new[which] += L*rng.randn()
  return new
```

Useful Plots: The Trace Plot

```
# Trace plot of the first parameter
plt.plot(keep[:,0])
```

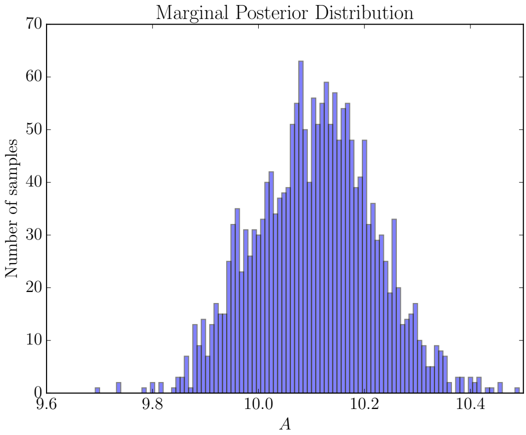
Useful Plots: The Trace Plot



Useful Plots: Marginal Posterior

```
# Marginal posterior for first parameter
# Excluding first 2000 points
plt.hist(keep[:,0], 100)
```

Useful Plots: Marginal Posterior



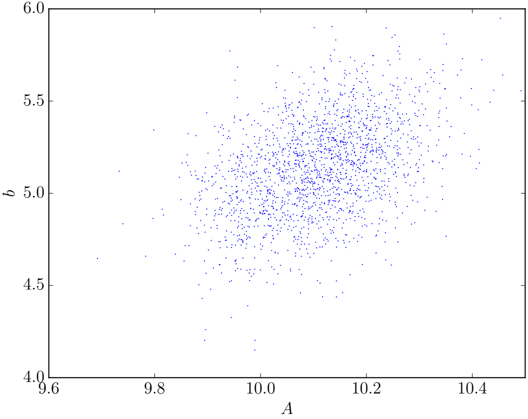
Comment on Histograms

If your histograms have so many points that they look perfectly smooth, you are working on an **easy problem**!

Useful Plots: Joint Posterior

```
# Joint posterior for first two parameters
# excluding first 2000 points
plt.plot(keep[:,0], keep[:,1], 'b.')
```

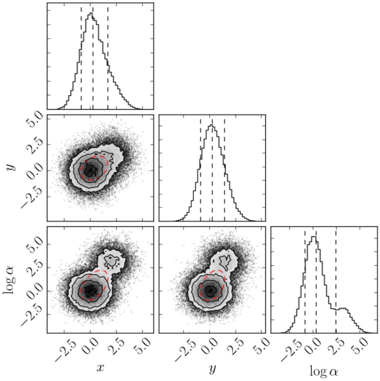
Useful Plots: Joint Posterior



Useful Plots: "Corner" or "Triangle" Plots

I like the package `corner.py` by Dan Foreman-Mackey

(`https://github.com/dfm/corner.py`)



Useful Summaries

Posterior distributions can be complicated. Often, we want a simple statement of the uncertainty. This leads to:

- Point estimates
- Credible intervals

Calculating Summaries

```
# Posterior mean and sd
np.mean(keep[:,0])
np.std(keep[:,0])

# For median and credible interval
x = np.sort(keep[:,0].copy())
# Credible interval (68%)
x[0.16*len(x)]
x[0.84*len(x)]
```

Bayes' Rule

Here is Bayes' rule again, with the background information (or assumptions) made explicit:

$$p(\theta|D, I) = \frac{p(\theta|I)p(D|\theta, I)}{p(D|I)}$$

In any particular application, we make a definite choice of the prior and the sampling distribution, as well as what θ, D, and I are.

What is a parameter?

What is a parameter?

- A quantity whose value you would like to know; or
- A quantity you think you need in order to write down $p(D|\theta)$.

The latter are often called **nuisance parameters**. For example, in the transit problem we might be interested only in w, but we can't use our "gaussian noise" assumption without also including A, b, and t_c.

In the Transit Example

Our parameters were:

$$\theta \equiv \{A, b, t_c, w\}$$

What was our data D? We had a data file with three columns: times $\{t_i\}$, measurements $\{y_i\}$, and "error bars" $\{\sigma_i\}$. Was this all our data D?

Answer: No!

Only the $\{y_i\}$ from the data file was our data. Why? We wrote down $p(\{y_i\}|\theta, I)$, but we did not write down $p(\{t_i\}|\theta, I)$, or $p(\{\sigma_i\}|\theta, I)$.

Therefore:

$$
\begin{aligned}
\theta &\equiv \{A, b, t_c, w\} \\
D &\equiv \{y_i\} \\
I &\equiv \{\{t_i\}, \{\sigma_i\}, \text{etc.}\}
\end{aligned}
$$

Assigning Priors

When assigning our priors (and sampling distribution), it is **completely legitimate** to use two out of the three columns of our "data" file!

Why use the log-uniform prior?

Let $\theta =$ the mass of a galaxy, in solar masses.

"Prior ignorance" might motivate this prior:

$$\theta \sim U(0, 10^{15}).$$

Why use the log-uniform prior?

"Prior ignorance" might motivate this prior:

$$\theta \sim U(0, 10^{15}).$$

But this implies:

$$\begin{aligned} P(\theta \geq 10^{14}) &= 0.9 \\ P(\theta \geq 10^{12}) &= 0.999. \end{aligned}$$

i.e. we are not ignorant at all, with respect to some questions!

Why use the log-uniform prior?

$$\log_{10}(\theta) \sim U(5, 15).$$

implies:

$$\begin{aligned} P(\theta \geq 10^{14}) &= 0.1 \\ P(\theta \geq 10^{12}) &= 0.3 \end{aligned}$$

or

$$P(\theta \in [10^{10}, 10^{11}]) = P(\theta \in [10^{11}, 10^{12}]) = P(\theta \in [10^{12}, 10^{13}])...$$

Using the log-uniform prior in Metropolis

Easiest way: just make $\theta' = \log(\theta)$ the parameter:

- Define proposals, etc, in terms of θ', which has a uniform prior
- Just exponentiate it ($\theta = e^{\theta'}$) before using it in the likelihood.

Let's apply this to the w (width) parameter in the transit model.

Using the log-uniform prior in Metropolis

Coding...

Safety Features

In "(data) = (model) + noise" type models, be sceptical of the gaussian noise assumption. For example, with $N = 1000$ data points and $\sigma_i = 1$ for all i, one consequence of the sampling distribution (really a prior) is:

$$P\left(\frac{1}{N}\sum_{i=1}^{N}(y_i - m(t_i;\theta)) \in [-0.06, 0.06]\right) \approx 95\% \tag{5}$$

Really? Seems a bit confident.

Safety Features

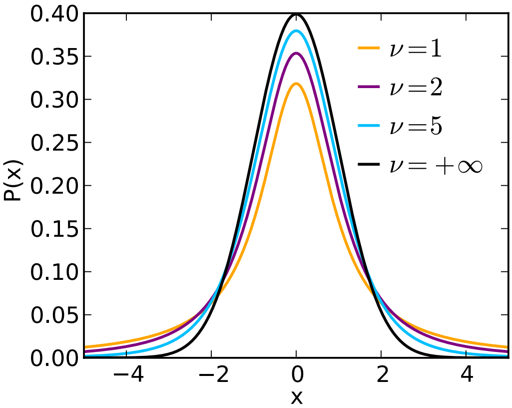
There are many ways to do this kind of thing. This is just my favourite. Replace:

$$y_i | A, b, t_c, w \sim \mathcal{N}\left(m(t_i; A, b, t_c, w), \sigma_i^2\right)$$

with

$$y_i | A, b, t_c, w \sim \text{Student-}t\left(m(t_i; A, b, t_c, w), (K\sigma_i)^2, \nu\right).$$

t Distributions from Wikipedia



t Density

For a single variable...

$$p(x|\nu, \mu, \sigma) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sigma\sqrt{\pi\nu}}\left[1 + \frac{1}{\nu}\frac{(x-\mu)^2}{\sigma^2}\right]^{-\frac{\nu+1}{2}}$$

Our likelihood is a product of N terms like this, and we have to code up the log of the likelihood. Also, remember we're scaling the widths σ by a factor K.

Priors for K and ν

Let's use

$$\log(\nu) \sim U(\log(0.1), \log(100)) \tag{6}$$

And for $K \geq 1$, let's use

$$p(K) = \frac{1}{2}\delta(K-1) + \frac{1}{2}e^{-K}. \tag{7}$$

Prior for K

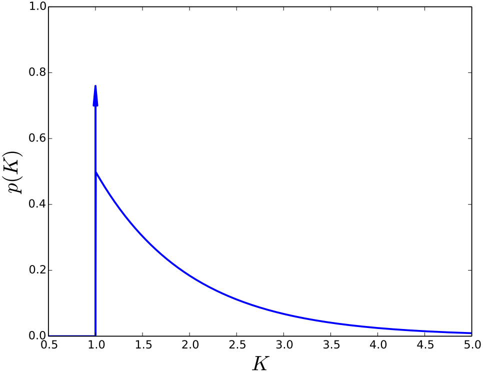
The prior

$$p(K) = \frac{1}{2}\delta(K-1) + \frac{1}{2}e^{-(K-1)}. \tag{8}$$

implies K might be precisely 1, or not. Computationally, there are two approaches:

- Make a $K = 1$ model and a $K \neq 1$ model, run them separately with a method that calculates marginal likelihoods (e.g. Nested Sampling)
- Make a single model which includes both possibilities.

Prior for K



Prior for K

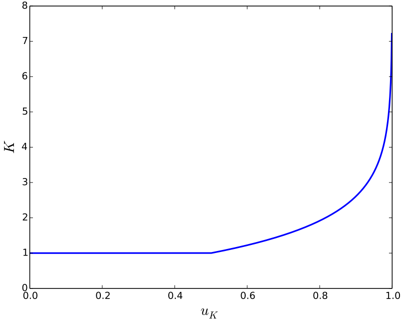
The prior

$$p(K) = \frac{1}{2}\delta(K-1) + \frac{1}{2}e^{-(K-1)}. \tag{9}$$

can be implemented by using u_K as a parameter with a $U(0,1)$ prior, and letting

$$K = \begin{cases} 1, & u_K < 0.5 \\ 1 - \log\left(1 - (2u_K - 1)\right), & \text{otherwise.} \end{cases}$$

Relationship between K and u_K



Let's implement this and find the posterior probability that $K = 1$.