emcee: The MCMC Hammer

Daniel Foreman-Mackey
David W. Hogg
Dustin Lang
Jonathan Goodman

See arXiv:1202.3665v3

emcee: The MCMC Hammer

"The goal of this project has been to make a sampler that is a useful tool for a large class of data analysis problems – a 'hammer' if you will."

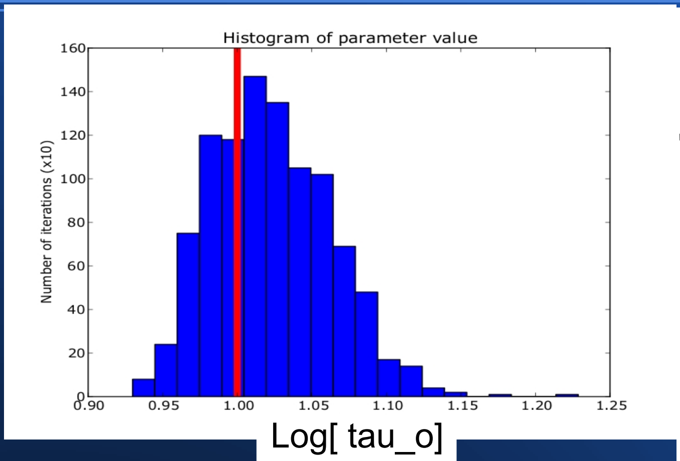
Outline

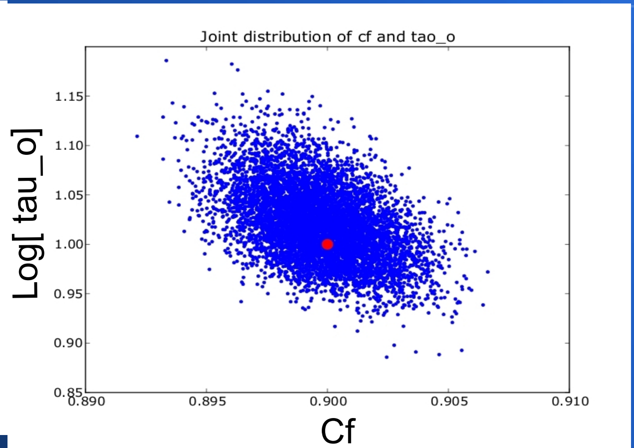
- Why use Markov chain Monte Carlo?
- A little Bayesian inference
- The Metropolis-Hastings algorithm
- The emcee algorithm
- Using emcee
- Identifying and solving potential problems

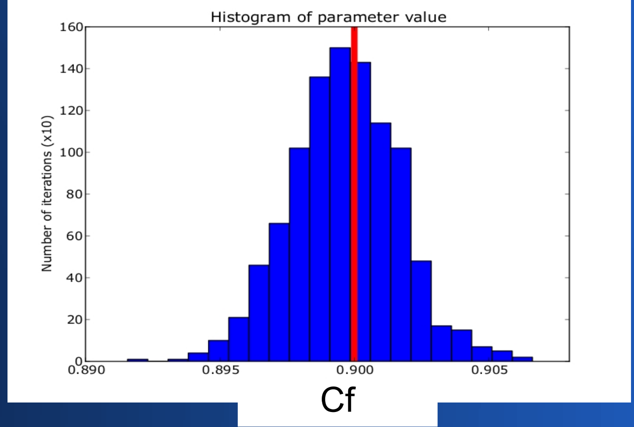
Why use MCMC?

- Determine best parameter values for a model with believable estimates of the uncertainty.
- Can explore high-dimensional parameter spaces.
- Can include uncertainty in various assumed constants.
- Explores degeneracy between parameters.

Parameter Correlations







Bayesian Inference

- Question: Given data D, what is the probability that D is described by hypothesis (model) H?
- Use the rules of probability theory:

prob(X|info) + prob(X*|info) = 1

prob(X, Y|info) = prob(X| Y, info) x prob(Y|info)

Bayes' Theorem

- Derivation:

prob(X, Y|info) = prob(X| Y, info) x prob(Y|info)

prob(Y, X|info) = prob(Y| X, info) x prob(X|info)

but, prob(X, Y|info) = prob(Y, X|info), so:

$$\text{prob}(X|\,Y, \text{info}) = \frac{\text{prob}(Y|\,X, \text{info}) \times \text{prob}(X|\text{info})}{\text{prob}(Y|\text{info})}$$

Bayes' Theorem

- In terms of data D and hypothesis H:

$$\text{prob}(H|\ D, \text{info}) = \frac{\text{prob}(D|\ H, \text{info}) \times \text{prob}(H|\text{info})}{\text{prob}(D|\text{info})}$$

Bayes' Theorem

- In terms of data D and hypothesis H:

$$\text{prob}(H|\,D,\,\text{info}) = \frac{\text{prob}(D|\,H,\,\text{info}) \times \text{prob}(H|\text{info})}{\text{prob}(D|\text{info})}$$

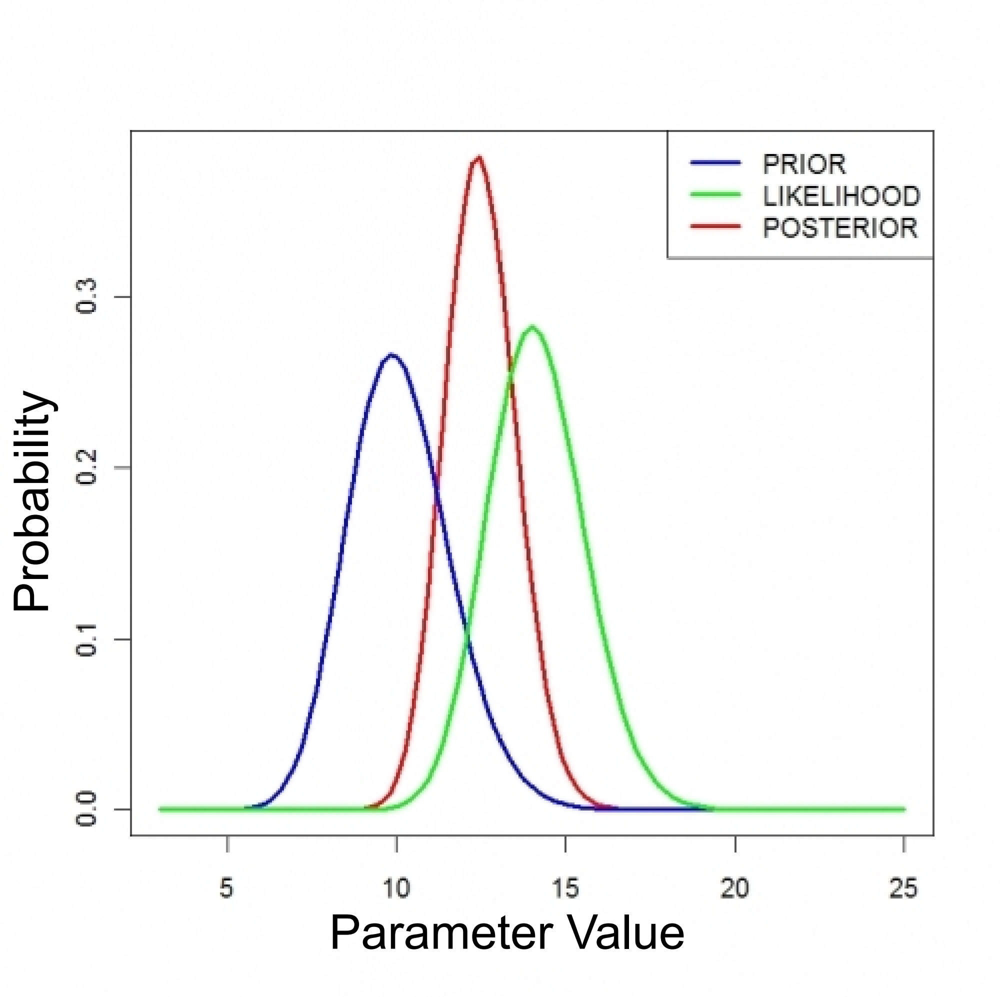
Posterior probability

Likelihood function

Evidence

Prior probability

Bayes' Theorem



Bayes' Theorem

- In terms of data D and hypothesis H:

$$\text{prob}(H|\ D, \text{info}) = \frac{\text{prob}(D|\ H, \text{info}) \times \text{prob}(H|\text{info})}{\text{prob}(D|\text{info})}$$

- Answers our question! Gives us the probability of the hypothesis H being true given the data D.

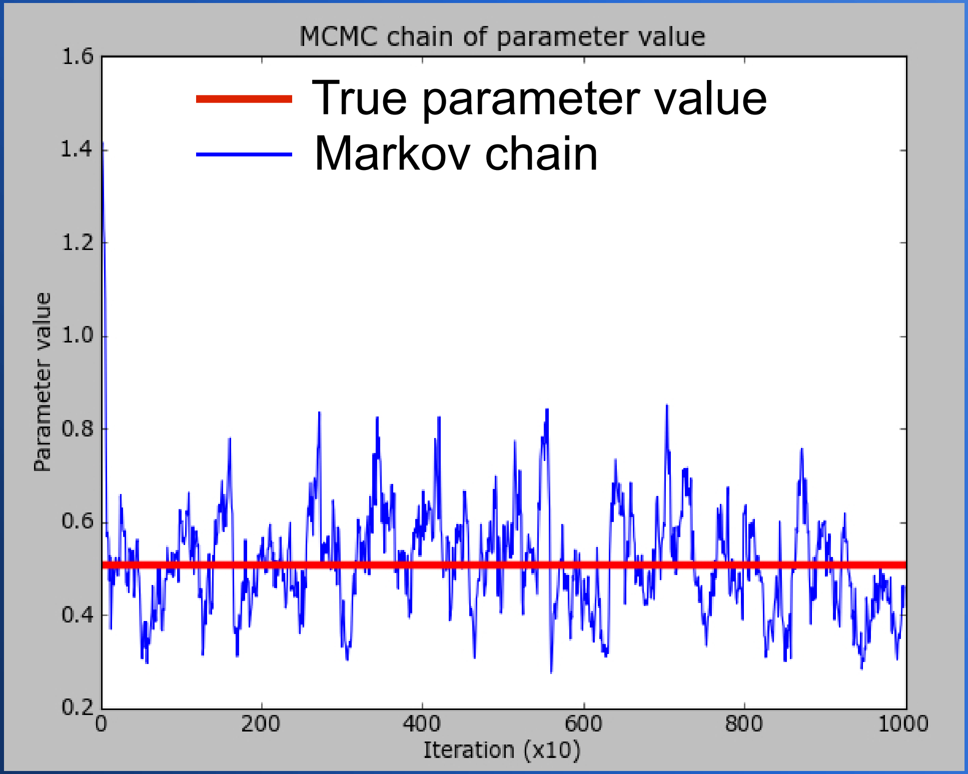
See Sivia and Skilling 2006 for details and examples!

How does this relate to MCMC?

- *The research question is:* given some data and a model with one or more parameters, what are the parameter values that best describe the data?
- We want to calculate the posterior probability distributions for the model parameters, but how do we do this when the parameter space is multi-dimensional?
- Instead of trying to recover the analytic form of the posterior pdfs, we sample from them using MCMC.

How MCMC works, qualitatively

- Step around multi-dimensional parameter space so that # of steps with a given parameter value is α to the posterior probability of that parameter.



What does this entail?

1) Start with an initial guess for the parameter values.
2) Calculate the likelihood that the data came from a model with the starting parameter values.
3) Perturb the parameters using some approved algorithm.
4) Calculate the likelihood that the data came from the model with the new parameter values.
5) Compare the likelihoods to decide whether to take the step and accept the new parameter values into the Markov chain, or to keep the old parameter values.

What counts as an approved algorithm?

- Need to maintain detailed balance:

  The ratio of the probability of taking the step from parameter value x1 to x2 to the probability of taking the step from x2 to x1 bust be the same as the ratio of the likelihoods of x1 and x2.

- Or in math:

$$\frac{P(X[i+1] = y \mid X[i] = x)}{P(X[i+1] = x \mid X[i] = y)} = \frac{f(y)}{f(x)}$$

Where we would like the chain to explore f, which is proportional to the true probability density of X.

See Brendon Brewer's thesis

Other Important Details

- Posterior probability also depends upon the prior probability

Bayes' Theorem

- In terms of data D and hypothesis H:

$$\text{prob}(H|\, D, \text{info}) = \frac{\text{prob}(D|\, H, \text{info}) \times \text{prob}(H|\text{info})}{\text{prob}(D|\text{info})}$$

Posterior probability

Likelihood function

Evidence

Prior probability

Other Important Details

- Posterior probability also depends upon the prior probability

  → when we choose whether to accept new parameter values, we compare the difference between the likelihoods as well as the difference in the prior probabilities.

- We neglect the evidence and just go with proportionality

  → need Nested Sampling to calculate the evidence: needed for model selection!

Metropolis-Hastings algorithm

- Propose to change parameter values using a compact distribution (e.g. Gaussian) around the current position in parameter space.
- Must determine size of the proposal distribution, so need to set N^2 tuning parameters (elements of the proposal covariance matrix).
- Not good for problems with large N!

The emcee algorithm

- Use a different proposal distribution and lots of “walkers” to explore parameter space simultaneously.
- Only 1 or 2 tuning parameters for N model parameters.
- Parallelized for multiple computer cores.

The stretch move

- Consider an ensemble of k "walkers" that explore parameter space.
- To move the walker X_k, randomly choose one of the other walkers, X_j and propose to move:
  $X_k(t) \rightarrow Y = X_j + Z\,(X_k(t) - X_j)$
- Where Z is a random variable drawn from g(z) α 1/sqrt(z) if z is in the set 1/a to a

The stretch move

- This means that walkers move along vectors between the walkers.
- You could start all the walkers evenly distributed over the parameter space, or you could start all the walkers in a dense clump around your best guess for the model parameters and the walkers will diffuse from there.

Using emcee

- You can probably set the tuning parameter "a" to 2 (good for almost all applications).
- Make sure that the acceptance rate for proposals is between 0.2 and 0.5 (out of 1).
- Make sure you don't have too many or too few samples $\rightarrow$ in the first case you waste your time and in the second case you don't sample the model parameter space well!

So how many samples do we need?

- You can calculate the autocorrelation time: the time (number of proposal moves) needed to obtain independent samples.
- Only need to run the code for a few (~10) autocorrelation times to get a basic answer.

Possible problems

- But what if you calculate the autocorrelation time and it's always a large fraction of the samples you've obtained so far?
  - Well, then there's a problem, and it may be that the parameter space is multi-model, which means emcee won't really work!
- There is also the assumption that parameter values can be acted upon by linear operations (so integer parameter values may not work)

Possible solutions

- Re-parameterize the model so that all parameters can be acted upon by a linear operator.
- Or, if the parameter space is multi-modal, we can consider a different algorithm, such as nested sampling.

Questions?

How do you choose when to accept?

- "Given a position X(t), sample a proposal position Y from the transition distribution Q(Y; X(t))" then "accept this proposal with probability"

  $\min(1, p(Y|D)/p(X(t)|D) \times Q(X(t); Y)/Q(Y; X(t)))$
- "Q(Y; X(t)) is an easy-to-sample probability distribution for the proposal Y given a position X(t)." e.g. a Gaussian centered around X(t)
- So $X(t+1) = Y$ or $X(t+1) = X(t)$

What counts as an approved algorithm?

- Or, as described by Sivia & Skilling:

  The transitions are in detailed balance when A is populated proportionately of L(A)... and B proportionately to L(B), because the forward and backward fluxes then balance.”
- … assuming that the acceptance probability of going from A to B is proportional to L(B)