

Supernova MCMC and HMC exercise

ICIC Data Analysis Workshop, 8-11 September 2014

1 Cosmology with Supernovae Ia

The aim of this exercise is to write an MCMC code to estimate cosmological parameters from supernova Ia data. Supernova Ia are standard candles (or can be made so), so can be used to measure the contents of the Universe.

2 Theory and parameters

The flux from a supernova of luminosity L is given by

$$f = \frac{L}{4\pi D_L^2}$$

where D_L is the *Luminosity Distance*. In Big Bang cosmology it is given by

$$D_L = \frac{(1+z)c}{H_0\sqrt{|1-\Omega|}} S_k(r),$$

where

$$r(z) = \sqrt{|1-\Omega|} \int_0^z \frac{dz'}{\sqrt{\Omega_m(1+z')^3 + \Omega_v + (1-\Omega)(1+z')^2}}.$$

and $S_k(r) = \sin r, r, \sinh r$, depending on whether $\Omega \equiv \Omega_m + \Omega_v$ is $> 1, = 1$, or < 1 , and z is the observed redshift of the supernova. Ω_m, Ω_v and H_0 are the density parameters (today) in matter, vacuum energy, and the Hubble constant. It is beyond the scope of these notes to derive this, but it is standard material for an undergraduate cosmology course.

For a flat Universe ($\Omega = 1$), this simplifies to

$$D_L(z) = 3000h^{-1}(1+z) \int_0^z \frac{dz'}{\sqrt{\Omega_m(1+z')^3 + 1 - \Omega_m}} \text{ Mpc},$$

where $H_0 = 100h \text{ km s}^{-1} \text{ Mpc}^{-1}$. To avoid evaluating integrals to calculate D_L , we can use an accurate fitting formula (valid for flat universes only), given by U.-L. Pen, ApJS, 120:4950, 1999:

$$D_L(z) = \frac{c}{H_0}(1+z) \left[\eta(1, \Omega_m) - \eta\left(\frac{1}{1+z}, \Omega_m\right) \right]$$

where

$$\eta(a, \Omega_m) = 2\sqrt{s^3 + 1} \left[\frac{1}{a^4} - 0.1540\frac{s}{a^3} + 0.4304\frac{s^2}{a^2} + 0.19097\frac{s^3}{a} + 0.066941s^4 \right]^{-1/8}$$

and $s^3 \equiv (1 - \Omega_m)/\Omega_m$. This is accurate to better than 0.4% for $0.2 \leq \Omega_m \leq 1$.

Fluxes are usually expressed in magnitudes, where $m = -2.5 \log_{10} F + \text{constant}$. The distance modulus is $\mu = m - M$, where M is the absolute magnitude, which is the value of m if the source is at a distance 10pc. With D_L in Mpc¹, this is

$$\mu = 25 - 5 \log_{10} h + 5 \log_{10} \left(\frac{D_L^*}{\text{Mpc}} \right)$$

¹There is a simplification in the exercise here; we assume we know what the absolute magnitude (or luminosity) of type I supernovae are, but in fact unless we have supernovae with known distances, we don't. In fact M and h are degenerate, since M is set from low-redshift supernovae where we assume Hubble's law to give us the distance. For the purpose of this exercise, we will cheat.

The Hubble constant has been factored out of D_L : $D_L^* \equiv D_L(h = 1)$.

If we have measurements of μ , then we can use Bayesian arguments to estimate the parameters Ω_m, Ω_v, h . For anyone unfamiliar with cosmology, these numbers are somewhere between 0 and 1.

3 Data

The data file (from the Riess et al 2007 ‘gold’ sample) consists of measurements of the redshift (assumed precisely accurate), and a distance modulus μ_i , with associated measurement errors σ_i . The sample file (SN.txt from the website, at <http://astro.ic.ac.uk/content/icic-data-analysis-workshop-2014-0>) contains one header line and there are $n = 291$ supernovae in total, with $z < 1.8$.

SN	z	mu	sigma	quality
SN90O	0.030	35.90	0.21	Gold
SN90T	0.040	36.38	0.20	Gold
SN90af	0.050	36.84	0.22	Gold

...

Plot these data on a graph.

4 Exercise

Write an MCMC code to estimate h and Ω_m from the supernova dataset, assuming the Universe is flat and the errors are gaussian, i.e. assume that the likelihood is

$$L \propto \exp \left[-\frac{1}{2} \sum_{i=1}^n \frac{(\mu_i - \mu_{th}(z_i))^2}{\sigma_i^2} \right]$$

where μ_{th} is the theoretical value of the distance modulus, for which you will need to compute the integral for D_L^* numerically, using the fitting formula for a flat universe.

- For the errors, assume there is an intrinsic scatter of 0.1 mag in the supernova luminosity, which should be added in quadrature to the errors in the table.
- Assume uniform priors on the parameters (so you estimate the likelihood)
- You might like to start with a very simple ‘top-hat’ proposal distribution, where the new point is selected from a rectangular region centred on the old point. For this you will need a simple random number generator.
- Explore visually the chain when you have (a) a very small proposal distribution, and (b) a very large proposal distribution, for a maximum of 1000 trials. What do you conclude?
- Show how the acceptance probability changes as you change the size of the proposal distribution from very small (say 0.001) to very large (say 100).
- Once you have settled on a ‘reasonable’ proposal distribution, compute the average value of the parameter estimates under the posterior distribution, and their variances and covariance.
- Optionally, generalise to non-flat Universes and include Ω_v as an independent parameter.

4.1 Tips

If you are estimating h , Ω_m and Ω_v , you can precalculate D_L for $h = 1$ as a function of Ω_m and Ω_v , and do a bilinear interpolation when you are running the chains (and divide by h). This will be much faster than computing D_L every time you change parameters. This is *not* necessary if your parameters are h and Ω_m only.

5 Extensions

- Importance sampling. Consider a non-flat prior, so the target distribution is the posterior, not the likelihood. We can still sample from the likelihood (as you have been doing), and construct the posterior by weighting the points with the prior to get the target. This is an example of *importance sampling*, where we sample from a different distribution from the one we eventually want. Apply a prior on the Hubble constant to your chain, assuming a gaussian prior with mean 0.738 and standard deviation 0.024. Now plotting all the points in the chain will give a graph which looks the same as your previous graphs, so what should you do? Compute the mean h , Ω_m with and without the prior.
- Write and apply a Gelman-Rubin convergence test, and deduce roughly how long the chains should be for convergence.
- Extend to perform Hamiltonian Monte Carlo. You might like to try to compare the performance of MCMC and HMC; you will need to decide what the right criterion is.

For HMC, the algorithm is (from Hajian 2006):

Hamiltonian Monte Carlo

```
1: initialize  $\mathbf{x}_{(0)}$ 
2: for  $i = 1$  to  $N_{samples}$ 
3:    $\mathbf{u} \sim \mathcal{N}(0, 1)$  (Normal distribution)
4:    $(\mathbf{x}_{(0)}^*, \mathbf{u}_{(0)}^*) = (\mathbf{x}_{(i-1)}, \mathbf{u})$ 
5:   for  $j = 1$  to  $N$ 
6:     make a leapfrog move:  $(\mathbf{x}_{(j-1)}^*, \mathbf{u}_{(j-1)}^*) \rightarrow (\mathbf{x}_{(j)}^*, \mathbf{u}_{(j)}^*)$ 
7:   end for
8:    $(\mathbf{x}^*, \mathbf{u}^*) = (\mathbf{x}_{(N)}^*, \mathbf{u}_{(N)}^*)$ 
9:   draw  $\alpha \sim \text{Uniform}(0, 1)$ 
10:  if  $\alpha < \min\{1, e^{-(H(\mathbf{x}^*, \mathbf{u}^*) - H(\mathbf{x}, \mathbf{u}))}\}$ 
11:     $\mathbf{x}_{(i)} = \mathbf{x}^*$ 
12:  else
13:     $\mathbf{x}_{(i)} = \mathbf{x}_{(i-1)}$ 
14: end for
```

$H = -\ln L + K$, where $K = \mathbf{u} \cdot \mathbf{u}/2$. One approach is to approximate U by a bivariate gaussian with covariances estimated from the MCMC code (or you can try computing the derivative exactly):

$$U = \frac{1}{2}(\theta - \theta_0)_\alpha C_{\alpha\beta}^{-1}(\theta - \theta_0)_\beta.$$

You can evolve the system with a naïve Euler method (not recommended in general for stability reasons), or use the leapfrog algorithm:

$$\begin{aligned}
u_i\left(t + \frac{\epsilon}{2}\right) &= u_i(t) - \frac{\epsilon}{2} \left(\frac{\partial U}{\partial x_i} \right)_{\mathbf{x}(t)} \\
x_i(t + \epsilon) &= x_i(t) + \epsilon u_i\left(t + \frac{\epsilon}{2}\right) \\
u_i(t + \epsilon) &= u_i\left(t + \frac{\epsilon}{2}\right) - \frac{\epsilon}{2} \left(\frac{\partial U}{\partial x_i} \right)_{\mathbf{x}(t+\epsilon)}.
\end{aligned} \tag{1}$$

Issues to consider are how many integration steps per point in the chain, and how big those steps are. For some discussion, see Hajian (2006), [astro-ph/0608679](#).