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 $\Omega = \Omega_m + \Omega_v > 1, = 1, < 1$, and $z$ is the observed redshift of the supernova. $\Omega_m, \Omega_v$ 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

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

where $H_0 = 100h\text{ km s}^{-1}\text{ Mpc}^{-1}$. 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 Mpc, this is

$$\mu = 25 - 5 \log_{10} h + 5 \log_{10}(D_L^*)$$

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

If we have measurements of $\mu$, then we can use Bayesian arguments to estimate the parameters $\Omega_m, \Omega_v, h$. 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 $\mu_i$, with associated errors $\sigma_i$. The sample file (SN.dat) contains one header line and there are $n = 291$ supernovae in total, with $z < 1.8$. 

Supernova MCMC and HMC exercise

Alan Heavens, University of Edinburgh, UK
SCMA V workshops, June 2011
<table>
<thead>
<tr>
<th>SN</th>
<th>z</th>
<th>mu</th>
<th>sigma</th>
<th>quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN90O</td>
<td>0.030</td>
<td>35.90</td>
<td>0.21</td>
<td>Gold</td>
</tr>
<tr>
<td>SN90T</td>
<td>0.040</td>
<td>36.38</td>
<td>0.20</td>
<td>Gold</td>
</tr>
<tr>
<td>SN90af</td>
<td>0.050</td>
<td>36.84</td>
<td>0.22</td>
<td>Gold</td>
</tr>
</tbody>
</table>

In R, you can read this into a $291 \times 5$ array `SNdat` with

```r
SNdat = read.table("SN.dat",header=T,skip=1)
```

and you can extract the columns into 1D arrays `SNname,z,...` with

```r
SNname = SNdat[,1]
z = SNdat[,2]...
```

4 Exercise

Write an MCMC code to estimate $h$ and $\Omega_m$ 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 $\mu_{th}$ is the theoretical value of the distance modulus, for which you will need to compute the integral for $D_L$ numerically (suggestion: do this by a crude method).

- Assume uniform priors on the parameters (so you estimate the likelihood)
- Optionally, generalise to non-flat Universes and include $\Omega_v$ as an independent parameter.

4.1 Tips

If you are estimating only $h$ and $\Omega_m$, you can precalculate $D_L$ for $h = 1$ as a function of $\Omega_m$, and interpolate when you are running the chains (and divide by $h$). This will be much faster than computing $D_L$ every time you change $h$ and $\Omega_m$.

The R routine `runif` will give random numbers with a uniform distribution - useful for a simple proposal distribution.

If you store the values of $h, \Omega_m, \Omega_v$ in a chain called `mchain[j,k]`, where $j$ takes the values 1,2,(3) and $k$ labels the position in the chain, then you can plot the chain in projection with commands like

```r
plot(mchain[1,],mchain[2,],xlab="h",ylab="Omega_m")
```

You might also like to plot histograms using `hist` for marginal distributions of each parameter.

Other functions you may find useful are `mean`, `sd`, `var` and `cov(mchain[1,],mchain[2,])` to compute means, standard deviations, variances, and the covariance of the estimates of $h$ and $\Omega_m$.

5 Extensions

- 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 $x(0)$
2: for $i = 1$ to $N_{\text{samples}}$
3:   $u \sim \mathcal{N}(0, 1)$ (Normal distribution)
4:   $(x^*_0, u^*_0) = (x_{(i-1)}, u)$
5:   for $j = 1$ to $N$
6:     make a leapfrog move: $(x^*_j, u^*_j) \rightarrow (x^*_j, u^*_j)$
7:   end for
8:   $(x^*, u^*) = (x_{(N)}, u_{(N)})$
9:   draw $\alpha \sim \text{Uniform}(0, 1)$
10:  if $\alpha < \min\{1, e^{-(H(x^*, u^*) - H(x, u))}\}$
11:    $x_{(i)} = x^*$
12:  else
13:    $x_{(i)} = x_{(i-1)}$
14: end for

$H = -\ln L + K$, where $K = u \cdot u / 2$. Approximate $U$ by a bivariate gaussian with covariances estimated from the MCMC code:

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

You can evolve the system with a naive Euler method, or use the leapfrog algorithm:

$$u_i(t + \frac{\epsilon}{2}) = u_i(t) - \frac{\epsilon}{2} \left( \frac{\partial U}{\partial x_i} \right)_{x(t)}$$

$$x_i(t + \epsilon) = x_i(t) + \epsilon u_i(t + \frac{\epsilon}{2})$$

$$u_i(t + \frac{\epsilon}{2}) = u_i(t) - \frac{\epsilon}{2} \left( \frac{\partial U}{\partial x_i} \right)_{x(t+\frac{\epsilon}{2})}.$$ 

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), astroph/0608679.