Prof. Carles Sánchez
Departament de Física, UAB, & IFAE
carles.sanchez@uab.cat

Barcelona Institute of Science and Technology

Statistics and Data Analysis (2024/25)
Assignment on Bayesian Statistics

Instructions

1. The assignment has to be done as a jupyter `python3` notebook, which should be uploaded to Classroom **along with** the PDF of the same notebook.
2. The deadline for the assignment is **Oct 23, 2024**.
3. The notebook has to be commented and all steps explained/justified, similar to the notebooks we have seen during the classes.

Assignment

A cosmological experiment[1] provides a measurement of the expansion history of the Universe, $H(z)$, obtained from the ages of passively-evolving galaxies in galaxy clusters at various redshifts, z. These measurements of $H(z)$ can be used to constrain the Hubble parameter H_0, quantifying the local expansion rate, and the matter density in the Universe, Ω_m, using the following model:

$$H(z) = H_0\sqrt{\Omega_m(1+z)^3 + (1-\Omega_m)}. \tag{1}$$

The measurements can be read from the data file in this assignment using this `python` code:

```python
import numpy as np
z, h, herr = np.loadtxt('Hz_BC03_all.dat',unpack=True)
```

If D denotes the data provided by the measurements of $H(z)$, and H_0, Ω_m are the parameters of the model of Equation (1), we want to use Bayesian parameter inference to compute the posterior probability

$$p(H_0, \Omega_m|D) \propto p(D|H_0, \Omega_m)\, p(H_0, \Omega_m)$$

using Monte Carlo Markov Chains (with the Metropolis implementation). For the likelihood, $p(D|H_0, \Omega_m)$, you can assume a Gaussian likelihood, given the measurement uncertainties that are provided in the data file (the column named `herr` above). In the case of this example, the data is composed of H_i measurements at different z_i, with uncertainties σ_i. The Gaussian likelihood will then be computed as:

$$p(D|H_0, \Omega_m) = \prod_i p(D_i|H_0, \Omega_m) = \prod_i \exp -(H_i - H(z_i, H_0, \Omega_m))^2/\sigma_i^2,$$

where $H(z_i, H_0, \Omega_m)$ is the evaluation of the model at z_i with parameters H_0, Ω_m. This can also be written as:

$$p(D|H_0, \Omega_m) = \exp -\frac{\chi^2}{2}, \text{ with } \chi^2 = \sum_i (H_i - H(z_i, H_0, \Omega_m))^2/\sigma_i^2$$

[1] https://arxiv.org/abs/0907.3149

To successfully complete this assignment, please follow the steps below:

1. Plot the measurements with their error bars along a handful of parameter combinations for the $H(z)$ model, some similar to the data, some different.
2. Define `python` functions for the model, the posterior (likelihood times prior) and the proposal distribution. For the prior distribution on the parameters, you can start by using a uniform prior over $50 < H_0 < 100$ and $0 < \Omega_m < 1$.
3. Write your MCMC code and run 4 or 5 chains, initializing them at different locations in the two parameters. Study them, plotting the parameters against the chain step, and cut out the burn-in regions.
4. Study the efficiency of the chains (steps accepted / steps proposed) and the convergence, running several independent chains starting from different points and checking that you get consistent posteriors.
5. From the samples of the chains, make a 2D density plot of H_0, Ω_m, and also make 1D plots of the histogram of each parameter (do they look Gaussian?). Calculate the 68% confidence interval around the mode of the distribution on each of the two parameters (please do it by counting on the histogram of the samples and not computing the standard deviation, as it could be non-symmetric).
6. Repeat the plot of point 1, but now with parameter combinations drawn directly from the posterior.
7. Finally, repeat the whole exercise but now using a Gaussian prior on $\Omega_m = 0.315 \pm 0.007$, given by measurements of the Planck Satellite[2]. This is a good example on how to use prior information in our analysis. What is the new 68% confidence interval on H_0? Comment on why is it larger or smaller. And what is the new 68% confidence interval on Ω_m, and why? Please do all the above checks on the sanity of the chains in this case.

Final remark: Please note that, even if this looks like a very specific example, the exact same procedure could be used on any problem concerning some data and a model with various parameters, where you want to estimate the parameters based on the data and your prior knowledge about the parameters. It is likely that all of you will encounter that sort of problem at some point (or often!), and now you have coded a very elegant and robust way to perform that parameter inference, and it can be ran in your laptop in just a minute or two. I hope it will be useful for you in the future!

[2]https://arxiv.org/abs/1807.06209