

# Question Set 4 — Metropolis and Nested Sampling Solutions

## Question 1

My version of `straightline.py` is now given in the `NSwMCMC` repository at <https://github.com/eggplantbren/NSwMCMC>, so get the latest version of that. The dataset is the distance from which people could read a road sign, as a function of age. Age is associated with worse eyesight, as you'd expect.

In the code, I parameterised the noise standard deviation using  $\ln \sigma$  instead of  $\sigma$ , as  $\ln \sigma$  has a uniform prior, and this is a little neater than parameterising it as  $\sigma$  and using  $p(\sigma) \propto \sigma^{-1}$ .

To run `plain_metropolis.py` on this problem, edit the import statement which by default imports things from `transit_model`, so that it imports the functions from `straight_line`. Then execute `plain_metropolis.py`. When `plain_metropolis` is imported, it will plot the data, and you'll have to close that plot before the rest of the code will run.

To compute posterior summaries, such as the posterior means and standard deviations, just load `posterior_samples.txt`. After excluding burn-in, I got

$$m = -3.00 \pm 0.42 \text{ feet per year} \quad (1)$$

$$b = 576 \pm 24 \text{ feet} \quad (2)$$

$$\sigma = 51.0 \pm 7.0 \text{ feet.} \quad (3)$$

I had to re-exponentiate the third column of output to get  $\sigma$  from  $\ln \sigma$ .

## Question 2

The version with the  $t$ -distribution for the sampling distribution is given in `straightline2.py`. The posterior distributions for  $m$ ,  $b$ , and  $\sigma$  aren't affected very much...

## Question 3

I ran `nested_sampling.py` on the `straightline` model, with 100 particles (that's the default in the `NSwMCMC` code). You can monitor the progress by looking at the image in `progress_plot.png`. The marginal likelihood was

$$\ln p(D|\text{straightline}) = -176.23 \pm 0.40 \quad (4)$$

For the modified version of the model, I got

$$\ln p(D|\text{straightline2}) = -175.85 \pm 0.39 \quad (5)$$

These are very close, taking into account the numerical uncertainties. Therefore, the data does not imply much about whether the  $t$  or the normal distribution model is better. That's because there weren't any discrepant points, and both the  $t$  and normal distributions are consistent with that. The posterior for  $\nu$  in the `straightline2` model would show that the data only rules out very low values.

Finally, it is usually not recommended to use Bayesian model comparison with naive broad priors, like I did here. However, I used the same broad priors for most of the parameters that the two models had in common, so I wasn't going to unintentionally favour one of the models solely due to the priors.

By the way, I blame Python (or at least, my inability to use Python well) for the slow speed of these computations. Try implementing NS in C/C++/Fortran or using whatever Python tricks I have failed to use, and it should work a lot faster!