Unit 6: Final Report

Computer Modelling

1 Aims

- Write up your discoveries from unit 5 in a computational report of up to 2400 words.
- Extend your work from unit 5 in one of a few directions.
- Work without step-by-step instructions.

NOTE: do not use generative AI to write your report. You must write it yourself. You can continue to use it to write new code needed for the report.

---

2 Preparation

Ensure that your code successfully runs and produces the expected results: compare to other students' results to check, or ask TAs in person or on Piazza.

Set up a document for your report; you can either use LaTeX or a word processor. Your report should include both: - a write-up of your results from unit 5 - a write-up of one of the mini-projects below.

Choose **one** of the mini-projects below. If your unit 5 code did not work, you can only choose Project A.

---

3 Aims of the report

The goal of a your report is to communicate your results with enough detail and information that others can independently judge the merits of your work. It is important to include enough information so your results could be independently reproduced.

You should not typically include code or file descriptions in the report; you can discuss algorithms, since these matter for the results, but the specifics of the code itself are not interesting to the reader. Your results would have been the same if you had organized the code differently, so are not important to the physics.

Good reports will condense the main result in an abstract, set the background of the physics in the introduction in an attractive way, and have a clear conclusion.

---

4 Report Structure

The report should be no more than 2400 words. This is a strict limit; you will be penalized for going over. Shorter reports are fine, and will not be penalized as long as they cover all the required material.

The limit does not count equations or tables, but does count figure captions. Material in any appendices will not be marked or included in the word count or marking.

Typical reports will have the following structure:

4.1 Abstract

This should very briefly summarize the physics of what you have done, and the main result(s), written so it can stand almost alone, separate from the rest of the report. Write this last.

4.2 Introduction

Your report's introduction should include the core mathematical and physical background for the problem you are working on, including the basic physics equations you implemented and their meaning, the data you are working with, and a description of the methodology you used (though you do not need to include the full Metropolis algorithm).

4.3 Results & Discussion

Your results section should make clear in each part exactly which problem is being analyzed. In each case show your results and discuss their meaning carefully with respect to the programming and/or physics, as appropriate.

If any of your results do not make sense then state this clearly and discuss possible causes; don't try to hide or ignore them!

4.3.1 Unit 5 Results

Document your core results from unit 5, including 1D and 2D histograms of the parameters. There are tools to make more sophisticated versions of these like `corner` if you wish. Include a table of the main statistical results for each parameter, and discuss the physical meaning of the results.

Include the grid likelihood in your unit 5 write-up to show consistency with the Metropolis results, but you don't need to use it for the mini-project below/

4.3.2 Convergence Tests

Demonstrate the convergence of your results with respect to the length of the Metropolis chains; i.e. show that your results don't change significantly if you use a longer chain.

4.3.3 Mini-Project Results

Include results and discussion from the mini-project you choose below.

4.3.4 Conclusions

(Re-)State the main conclusions you have arrived at, and whether (and how you know) your analysis was satisfactory and why.

For best marks, compare your results to values from literature (you will have to search for these).

Finally, briefly reflect on project: are there changes you would make to the algorithms or methods to improve the simulation? What else could be improved?

absolute magnitude of the supernova, should be varied for each supernova

dark energy varies with redshift (integrand we used was an approximation)

Integral method, but did not contribute a lot, it was quite accurate.

5 Mini-Projects

Pick one of these mini-projects to include in your report.

5.1 Project A: Unsuccessful Codes

Only do this mini-project if your code has not successfully computed correct MCMCs and likelihoods, and you are unable to fix it in time. Your report can still get reasonable grades even if it is about a code that has not worked.

Document as carefully as you can which aspects of your code do work, and which bits don't. Illustrate both with plots and values. Where possible, explain what makes you think the values are incorrect.

Given these results, discuss the possible parts of the code that might be the source of the problem, and which can be assumed to be working.

5.2 Project B: Improved Likelihoods

When we developed the likelihood in unit 4 we actually ignored a potentially important component. We assumed that the errors on the data points in the supernova data were independent, but this is not actually correct: there is some correlation between them. In this mini-project you will upgrade your likelihood to include this correlation, and determine its effects.

5.2.1 Project B Information

A *covariance matrix* is a matrix that describes the correlation between the errors on the data points. It is a square matrix with the same number of rows and columns as the number of data points. Its diagonal is the variances σ_i^2 of the data points, and the off-diagonal elements are the covariances between them.

When you have a covariance matrix, the likelihood instead becomes:

$$\log \mathcal{L} = -\frac{1}{2} \sum_{i,j} (d_i - m_i(\theta)) C_{ij}^{-1} (d_j - m_j(\theta))$$

or equivalently in vector form:

$$\log \mathcal{L} = -\frac{1}{2} (\mathbf{d} - \mathbf{m}(\theta))^T \mathrm{C}^{-1} (\mathbf{d} - \mathbf{m}(\theta))$$

where $\mathbf{d}$ is the vector of data points, $\mathbf{m}$ is the vector of model values, and C is the covariance matrix.

Note that you can do matrix-vector multiplication in numpy with the `@` operator using numpy arrays. We don't recommend using the numpy `matrix` class.

5.2.2 Project B Task

You can find a covariance matrix for the Pantheon supernova data in a file in the Unit 6 folder on Learn.

Upgrade your likelihood to read and use this file. Compare likelihood and sampling results using it to the previous results, and discuss the differences. Quantify the resulting differences in the model parameters.

5.3 Project C: Varying the absolute magnitude

In Units 4 and 5 you fixed the M parameter which represents the absolute magnitude (brightness) of a supernova. In this mini-project you will free this parameter.

5.3.1 Project C Information

In the last part of this project you will need to use a *Prior*. This is an extra piece of external information that you include along with your likelihood.

In this project you will use a Gaussian prior on the M parameter. This is a bit like the likelihood for the supernova data, but instead of the x value being the distance modulus it is simply the (scalar) value of M:

$$\log \pi(M) = -\frac{1}{2}\frac{(M - M_0)^2}{2\sigma_M^2}$$

The log-prior is added into the log-likelihood to give the total log-probabilty to use in your sampling (the total is called the *posterior*):

$$\log \mathcal{P} = \log \mathcal{L}(\theta) + \log \pi(M)$$

5.3.2 Project C Task

Upgrade your likelihood and sampling to vary the M parameter with proposal size of 0.2 instead of fixing it to -19.3. To start with, let M vary between -20 and -19 (don't add the prior yet). Sample this likelihood and produce the same kinds of plots as you did for unit 5. You should find a notable result in one of the plots. Document it and explain it in terms of the equations of the system.

You may find that your chain does not settle down to a steady value of some parameters any more, but instead wanders around more broadly - this is not a problem here, but you should discuss what effect it has on the likelihood.

Next, add a Gaussian prior on the M parameter with mean $M_0 = -19.3$ and standard deviation $\sigma_M = 0.02$. Compare the results you get when including this to your previous results, and discuss carefully.

5.4 Project D: Varying the Equation of State

This project is **not** possible if you used the fallback cosmology class.

The equations in unit 1 assumed that dark energy was a constant, called the cosmological constant. In this project you will upgrade your code to allow dark energy behaviour to vary with redshift.

5.4.1 Project D Information

Dark energy could have a varying behaviour with time. We model this by allowing its *equation of state* $w(z)$ to vary; this is the ratio between its pressure and density.

The equation of state could have a complicated history, but a useful test model is one in which it is constant in time, $w(z) = w_0$. The effect of this on our equations here is that the integrand in your Cosmology class becomes:

$$\left[\Omega_m(1+z')^3 + \Omega_k(1+z')^2 + \Omega_\Lambda(1+z')^{3(1+w_0)}\right]^{-\frac{1}{2}}$$

where w_0 is a new parameter of the model. If $w_0 = -1$ then this reduces to the cosmological constant model that we used before.

5.4.2 Project D Task

Add the new parameter w_0 to your Cosmology class and Likelihood. For the proposal use $\sigma_{w0} = 0.3$.

Explore the effect of these new parameters, and sample the likelihood with them. Generate the same kinds of plots as you did for unit 5. Discuss your findings.

5.5 Project E: Metropolis Diagnostics

The Metropolis algorithm is noisy, and if your proposal is not very good it can take a long time before it converges to the correct distribution. In this project you will implement some diagnostics that we use to check if Metropolis is working correctly.

5.5.1 Project E Information

We will implement two statistics here.

The first is the *acceptance rate*, which is the fraction of proposed steps that are accepted. This should ideally be around 0.25 for a well-tuned Metropolis algorithm.

The second is the *Gelman-Rubin statistic*, which is a measure of how well the chains have converged. To use this statistic you need to run two or more separate chains. It compares how well they agree with each other.

To compute it, you run M separate chains, each of length N. Once they are done, you can work out the mean and variance of each parameter in each chain. For each parameter, you then have two vectors of length M, one vector $\underline{\mu}$ of the means, and one vector $\underline{v}$ of the variances. Then calculate the values B, which is the variance of the means $\underline{\mu}$, and W, which is the mean of the variances $\underline{v}$. The Gelman-Rubin statistic is then:

$$R = \sqrt{\frac{N-1}{N} + \frac{M+1}{M}\frac{B}{W}}$$

If the value of $R-1$ is small then the chain has converged. Common convergence criteria include $R-1 < 0.01$ or $R-1 < 0.03$

5.5.2 Project E Task

Add code to calculate and report the acceptance rate of your Metropolis algorithm. Discuss the results and what they mean for the efficiency of your chains.

Add code to calculate and report the Gelman-Rubin statistic of your Metropolis algorithm for each parameter, after running multiple chains. Choose a criterion, and use it to calculate how many samples are needed in the chains for convergence. Plot and discuss your results.

---

6 Submission and Marking Scheme

6.1 Mark Scheme

The report is marked out of sixty points, and is worth 30% of the course final grade.

The following categories are used to mark the report, each worth 10 points (i.e. evenly weighted):

- New code structure & style (except for project A)
- Abstract & Introduction
- Core results discussion
- Mini-project results & discussion
- Conclusions
- Report layout, language, and graphics

6.2 Submission

Submit your report as a PDF to the "Unit 6: Final report" TurnItIn submission box. Submit your complete final code, including all files we need to run it (apart from the Pantheon data files), to the "Unit 6: Final report code" submission box.