# Bayesian Analysis - Predicting Hubble Constant from Supernova Data

Joshua Gowdy

May 15, 2025

# Contents

## 0.1 Abstract

This report will describe an classic application of **Bayesian methods to supernova data and cosmological model building and parameter estimation.** We wish to estimate the Hubble constant, $h$, and density of matter in the universe, $\Omega_m$ from supernova data. We will build a model of supernova observations in a probabilistic Bayesian framework evaluating likelihood, prior and posterior distributions against the data. In order to estimate parameters and build summary statistics we must sample from our model posterior which we do so using **Markov Chain Monte Carlo sampling via Metropolis-Hastings algorithm and Hamiltonian Monte Carlo algorithm.**

# Chapter 1

# Model Setup and Background

## 1.1 The Goal

**We wish to estimate the Hubble constant, $h$, and density of matter in the universe, $\Omega_m$ from supernova data.** Together these parameters determine the observed rate of expansion of the universe. Hubble constant drives expansion while a higher density of matter slows this expansion down via gravity.

We will use supernova as 'standard candles' i.e. objects whose true brightness is constant (nearly so) and known[1]. This standard candle property allow us to judge the distance to supernova since their apparent brightness in the night sky should be a simple function of the distance to the object. Further away = fainter brightness. In turn we may measure their distance a second way using their observed redshift i.e. the stretching of the wavelength of the light coming from the object (as compared to their standard spectrum). This stretching is due to their recessional velocity thanks to the expansion of space and increases with distance in a manner which depends key parameters we wish to estimate: $h$ and $\Omega_m$. These parameters are known to be between zero and one.

In short we expect a relation between the redshift $z$ of a supernova and its apparent brightness as measured by the distance modulus $\mu$. The data collected for these quantities is described below.

## 1.2 The Data

The dataset used in this analysis is derived from the 'JLA' (Joint Lightcurve Analysis) sample, which includes approximately 700 type Ia supernovae. These supernovae are grouped into 31 narrow bins of redshift, with each bin providing a single value of the distance modulus $\mu$. The corresponding sample file, `supernova_data_file_jla_mub.txt`,

[1]This is not strictly true and is a simplification. In fact the true brightness of supernovae is extrapolated from observations of the brightness of nearby supernova whose distance is ascertained by other means (sometimes even using Hubble's law). We gloss over this fact for this toy study.

contains $n = 31$ pairs of data points $(z_i, \mu_i)$, representing supernovae with redshifts $z < 1.3$. Each pair corresponds to a redshift bin and the average observed distance modulus for that bin.

In addition to the binned measurements, the dataset includes a covariance matrix quantifying the variance uncertainties in the binned distance moduli, as well as the correlations between different redshift bins. This matrix is provided in a separate file named `jla_mub_covmatrix.txt`, which contains the elements of $C$.

## 1.3 The Model

We will construct a theoretical model of the relationship between redshift $z$ of a supernova and its apparent brightness as measured by the distance modulus $\mu$. The model will involve parameters $h$ and $\Omega_m$. Both the parameter space and the sample space of the data are hence two-dimensional. If the reader is not familiar with the physics they may simply think of this report as an investigation of the following model

$$\mathcal{M}: \quad \mu - \mu_0(z; h, \Omega_m) \ \sim \ \text{Noise} \tag{1.1}$$

for some distribution of $z$ values.

We won't say anything about the underlying distribution of $z$ values themselves. That is we haven't modelled the pair $(z, \mu)$ but instead modelled the distribution of $\mu$ given $z$. I.e. in effect we include the $z$ values as parameters of the model. This is a valid method since we are interested in the relationship between the two not their overall joint distribution. It should still be noted that our model inherently assumes a 'prior' for these $z$ values namely that they sit within the range observed in the data. Outside this range our model may not extrapolate well!

In short our model dictates that the $\mu$ values should follow a specific theoretical curve $\mu_0$ up to some noise. The modelling of this noise must also be contained with our probabilistic model which we describe shortly. First we describe the Bayesian framework for handling probabilistic models.

## 1.4 Bayesian Methods

In a probabilistic framework we can define a random variable

$$X = (z, \mu)$$

encoding the data from an observation of a supernova. When we collect data from multiple supernova we are sampling from the joint distribution of this random variable.

We can construct a probabilistic model of this data, parametrising this model by parameters $\theta$. In which case the likelihood is the probability that we observe the data given the model and its chosen parameters

$$\mathcal{L}(X, \theta) = P(X|\theta) \tag{1.2}$$

in our case the parameters are

$$\theta = (h, \Omega_m).$$

Given a model and some parameters we can predict the distribution of the supernova samples in the future. As well as a prediction engine such a model also gives a method to evaluate uncertainties.

**What about finding the best estimates for the parameters $\theta$ in the first place?**

This relates to our goal of parameter estimation and fitting the model. To do so we need to find the implied probability distribution in parameter space i.e. over the values of $h$ and $\Omega_m$, *given the observed data* $X$.

We can use Bayes theorem

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}. \tag{1.3}$$

We rarely know the model independent distribution for the data i.e. the total probability $P(X)$ so we work with

$$P(\theta|X) \sim P(X|\theta)P(\theta) \tag{1.4}$$

this is the un-normalised posterior probability written in terms of our model likelihood $\mathcal{L}(X, \theta) = P(X|\theta)$ and the prior 'belief' for the distribution of the parameters before we observed any data $P(\theta)$.

Once we have this distribution we can devise a method to sample from it and numerically compute best estimates for the parameters and properties of the distributions spread describing uncertainty in our estimates.

### 1.4.1 Why use Bayesian methods in this case?

We use Bayesian methods for this problem for a variety of reasons.

- We wish to propagate all sources of uncertainty in our model—both measurement error and prior uncertainty– returning full distributions rather than point estimates for parameter.
- We have an analytic, physics-informed model at hand and strong priors for our parameters. This aids both predictive power and interpretability over say a ML model.
- a ML model relies purely on the data to learn relationships and would struggle in this low data environment. Bayesian methods on the other hand can incorporate uncertainty in the data directly.
- priors in Bayesian methods act as a regulariser limiting the (over)influence of the data on predictions. A ML model in this scenario would likely over fit.

We now finally describe the full Bayesian model.

## 1.5 The Model Continued

A first guess at a model is simply

$$\mathcal{M}: \quad \mu - \mu_0(z; h, \Omega_m)) \ \sim \ \mathcal{N}(0, \sigma^2) \tag{1.5}$$

Where we have assumed that the noise is given by independent and identically distributed errors. For arguments sake they could be Gaussian with mean 0 (i.e. no bias in the model). The assumption of no bias in $\mu$ is a statement of intent that our model should capture the underlying relationships present in the data.

The errors are unlikely to be identically distributed so the variance $\sigma^2$ should more properly be given as $\sigma^2(z)$ since there is no reason to assume that the variance of the error stays the same across the population of supernova at a range of red shifts. Furthermore, in the above we have also assumed that the errors are uncorrelated between observations. This is unlikely since there may be systematic errors in the observation method which means that if one observation has a larger than predicted $\mu$ it is likely the next observation will also be larger even at a different $z$.

Instead we should model all the observations in the data set as a joint multivariate distribution $a = 1, ..., 700$

$$\mathcal{M}: \quad \mu_a - \mu_0(z_a; h, \Omega_m)) \ \sim \ \mathcal{N}(0, \mathcal{C}) \tag{1.6}$$

which could be Gaussian and were the covariance matrix of the errors across the observations is $\mathcal{C}$. Component $\mathcal{C}_{aa}$ is the variance of the error in the observation of $\mu_a$ made at redshift $z_a$. The off diagonal elements $\mathcal{C}_{ab}$ describe the correlations between errors made at different observations e.g. at different red shifts.

**How should we deal with these covariance parameters of the errors? Indeed why assume they are Gaussian?** It seems as though we need a theoretical model of the error profile of the observations made at different $z$ and how these errors may be correlated. The covariance matrix then becomes another collection of parameters of the model. However in this study we do not wish to model them in order to get their posterior distributions (sometimes called a nuisance or intermediary parameter in the Bayesian literature). Rather than including these covariance parameters alongside $h$ and $\Omega_m$ in defining prior and likelihoods for the supernova population, we will instead fix these parameters immediately to the sample covariances observed within the supernova data that was collected. I.e we use the sample estimate of these quantities right of the bat. This is similar to how we handled the $z$ values which we fixed in the model to their observed values rather than considering their underlying distribution.

So to access the covariance information of the observed data, which included 700 supernova, we will resample the data and define a set 31 bins. These are defined by the red shift values z, i.e. a bin captures all supernova with redshifts within the bin's

bound. For each bin we compute the average $z$ value and the average $\mu$. These form our new dataset $(z_i, \mu_i)$ for $1 = 1, .., 31$. Then as advertised, we will also compute directly from the data the within-bin variances and the between-bin covariances to form a sample covariance matrix $C_{ij}$. Then our model is for $1 = 1, .., 31$

$$\mathcal{M}: \quad \mu_i - \mu_0(z_i; h, \Omega_m)) \ \sim \ \mathcal{N}(0, C). \tag{1.7}$$

Now we have good reason to believe that the errors are indeed Gaussian since, unlike the error profiles of the raw values, the errors about the mean values should be normally distributed by the central limit theorem.

With this model we may immediately write down the likelihood $\mathcal{L} = P(\{\mu_i\} \mid h, \Omega_m, \{z_i\})$

$$\mathcal{L} \sim \exp\left[ -\frac{1}{2} \sum_{i,j} (\mu_i - \mu_0(z_i; h, \Omega_m)) C_{ij}^{-1} (\mu_j - \mu_0(z_j; h, \Omega_m)) \right]. \tag{1.8}$$

We will mostly be working with flat priors for the parameters $P(h), P(\Omega_m)$ equal to 1 on the range $[0, 1]$, justified by our prior physics informed belief about these parameters. So the posterior and likelihoods are identical. We will often also work with log-probabilities to improve stability of the code in handling the often very small probabilities. So the log-posterior (= log-likelihood) is just

$$\log \mathcal{L} = -\frac{1}{2} \sum_{i,j} (\mu_i - \mu_0(z_i; h, \Omega_m)) C_{ij}^{-1} (\mu_j - \mu_0(z_j; h, \Omega_m)). \tag{1.9}$$

We have reached our goal for this chapter: our model's distribution function from which we will sample (using MCMC methods) in order to find the estimates of the parameters (e.g. the mean and median of the distribution) and also the uncertainty in those estimates (spread of the distribution).

To find the above distribution we also need the theoretical $(z, \mu)$ relation given by $\mu_0(z; h, \Omega_m)$. In figure 1.1, is a plot of the data against some example theoretical curves with different values of the Hubble constant and matter density. Looks like these constants are close to $h = 0.7, \Omega_m = 0.3$ but how to be sure? That's what the next chapter does.

Below we briefly describe the physics derivation for the theoretical curves in $(z, \mu)$ space given by $\mu_0(z; h, \Omega_m)$.

### 1.5.1 Theoretical $(z, \mu)$ Curves

**This can be skipped for a none physics audience.**

The flux from a supernova of luminosity $L$ is given by

$$f = \frac{L}{4\pi D_L^2} \tag{1.10}$$

where $D_L$ is the luminosity distance. In Big Bang cosmology, it is given by

$$D_L = \frac{(1+z)c}{H_0 \sqrt{|1 - \Omega|}} \, S_k(r) \tag{1.11}$$

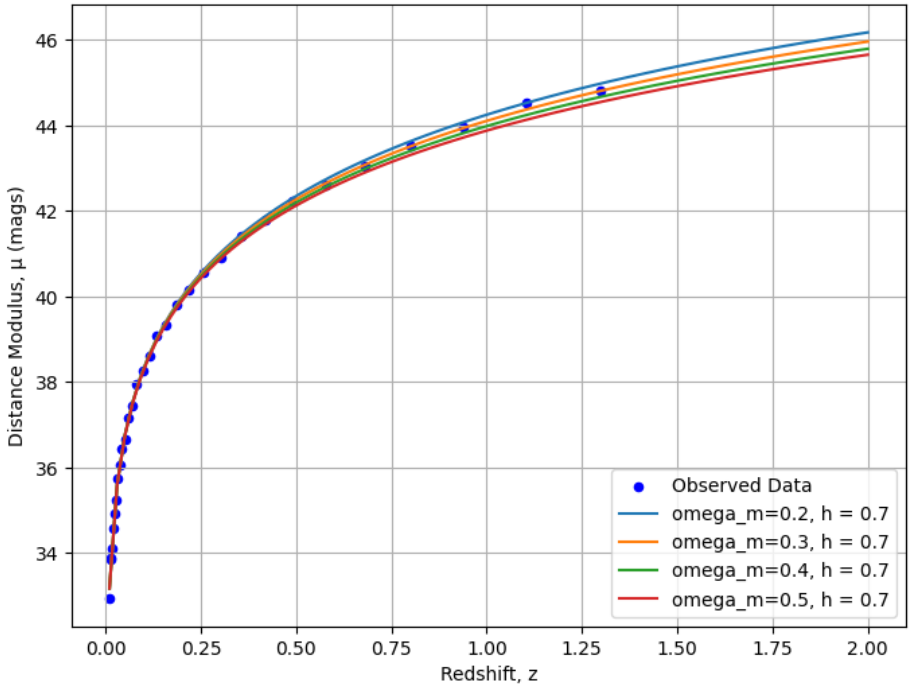


Figure 1.1: Plot of the data against some example theoretical curves with different values of Hubble constant and matter density

where

$$r(z) = \sqrt{|1-\Omega|}\int_0^z \frac{dz'}{\sqrt{\Omega_m(1+z')^3+\Omega_v+(1-\Omega)(1+z')^2}} \tag{1.12}$$

and

$$S_k(r) = \begin{cases} \sin r & \text{if } \Omega > 1 \\ r & \text{if } \Omega = 1 \\ \sinh r & \text{if } \Omega < 1 \end{cases}$$

Here, $\Omega = \Omega_m + \Omega_v$, and $z$ is the observed redshift of the supernova. $\Omega_m$, $\Omega_v$, and $H_0$ are the current density parameters for matter, vacuum energy, and the Hubble constant respectively.

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

$$D_L(z) = 3000\,h^{-1}(1+z)\int_0^z \frac{dz'}{\sqrt{\Omega_m(1+z')^3+1-\Omega_m}} \quad \text{(in Mpc)} \tag{1.13}$$

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

To avoid evaluating integrals for each $D_L$, an accurate fitting formula valid for flat universes (from U.-L. Pen, ApJS, 120, 49 (1999)) is given by:

$$D_L(z) = \frac{c}{H_0}(1+z)\left[\eta(1,\Omega_m) - \eta\left(\frac{1}{1+z},\Omega_m\right)\right] \tag{1.14}$$

where

$$\eta(a,\Omega_m) = \frac{2}{\sqrt{s^3+1}}\left[\frac{1}{a^4} - 0.1540\frac{s}{a^3} + 0.4304\frac{s^2}{a^2} + 0.19097\frac{s^3}{a} + 0.066941 s^4\right]^{-1/8} \tag{1.15}$$

and

$$s^3 \equiv \frac{1-\Omega_m}{\Omega_m} \tag{1.16}$$

Fluxes are usually expressed in magnitudes:

$$m = -2.5\log_{10} F + \text{constant} \tag{1.17}$$

The distance modulus is

$$\mu = m - M \tag{1.18}$$

where $M$ is the absolute magnitude (value of $m$ if the source is at 10 parsecs). With $D_L$ in Mpc and factoring out $h$, we define $D_L^* = D_L(h=1)$, so that

$$\mu_0 = 25 - 5\log_{10} h + 5\log_{10}\left(\frac{D_L^*}{\text{Mpc}}\right) \tag{1.19}$$

# Chapter 2

# Sampling the Model

We have set up the model and found its log-posterior distribution function (1.9) as a (quite complicated) analytic function. We need a method to sample this distribution, i.e. generate data points in parameter space $(h, \Omega_m)$ which follow this distribution function. Once we have these data points we can compute estimates (e.g. the mean and median) of the parameters and also the uncertainty/covariance matrix in these parameters.

Were the distribution function simpler we could possibly try to analytically find its mean and median. This is not feasible even for this quite simple model so we resort to simulation. Namely, we turn to Monte-Carlo Markov Chain sampling methods.

## 2.1 Monte Carlo Markov Chain methods

Monte Carlo Markov Chain (MCMC) methods are a class of algorithms that rely on random sampling to estimate numerical results e.g. to estimate a distribution or expectation which is analytically intractable. The samples are drawn according to a Markov chain. This is a sequence of random variables $X_1, X_2, X_3, \ldots$ with the *Markov property*: the distribution of the next state depends only on the current state, not the full history:

$$P(X_{n+1} = x \mid X_n = x_n, X_{n-1}, \ldots, X_1) = P(X_{n+1} = x \mid X_n = x_n).$$

This property enables the construction of a memory-efficient and tractable model for sequential stochastic processes. The 'memoryless' property means that, after running the chain for a sufficient number of steps it 'forgets' its initial condition and can reach an equilibrium behaviour consisting of random 'oscillations' according to its stationary distribution. We generate samples from a target probability distribution $\pi(x)$ by constructing a Markov chain whose stationary distribution is $\pi(x)$. After accounting for the initial 'burn in' phase the generated samples can be treated as approximately drawn from $\pi(x)$.

## 2.2 Metropolis-Hastings Algorithm

One of the most common MCMC algorithms is the Metropolis-Hastings algorithm. This algorithm works by taking a random walk in the variable space which is weighted by the target distribution and so in effect samples this distribution. The random walk proposes a new sample according to a proposal distribution and accepts the sample and moves to the new point if the target distribution is greater at the new point. If the new point is lower it may still accept the new point with a probability proportional to the ratio of the probabilities at the two points.

This random walk spends more time/samples in areas of oh higher probability according to the target distribution versus those with lower probabilities. It occasionally visits a lower probability region due to its random nature and non zero acceptance probability for such regions.

It works as follows:

1. Start at an initial point $x_0$.
2. At a point $x_k = x$:
   (a) Propose a new point $x'$ based on a proposal distribution $q(x' \mid x)$. E.g. a Gaussian centred at $x$.
   (b) Compute the acceptance ratio:
   $$r = \frac{\pi(x')q(x \mid x')}{\pi(x)q(x' \mid x)}.$$
   For symmetric proposal distributions (E.g. Gaussian centred at $x$ or Uniform) we have $q(x \mid x') = q(x' \mid x)$ then
   $$r = \frac{\pi(x')}{\pi(x)}$$
   compares the value of the distribution at the two points
   (c) Accept $x'$ with probability $\min(1, r)$ and set $x_{k+1} = x'$. Otherwise, stay at $x_k = x$.
   If $r > 1$ acceptance is guaranteed otherwise acceptance probability is proportional to $r$.
3. Repeat the process to build a chain $\{x_0, x_1, x_2, \ldots\}$.

The results of this sampling algorithm for our model are presented in Figure 2.1 with a bivariate Gaussian proposal distribution with uncorrelated variances = 0.01, an initial point $(0.5, 0.5)$ and 10000 sample points.

We can discard the burn in phase giving the distribution in Figure 2.2.

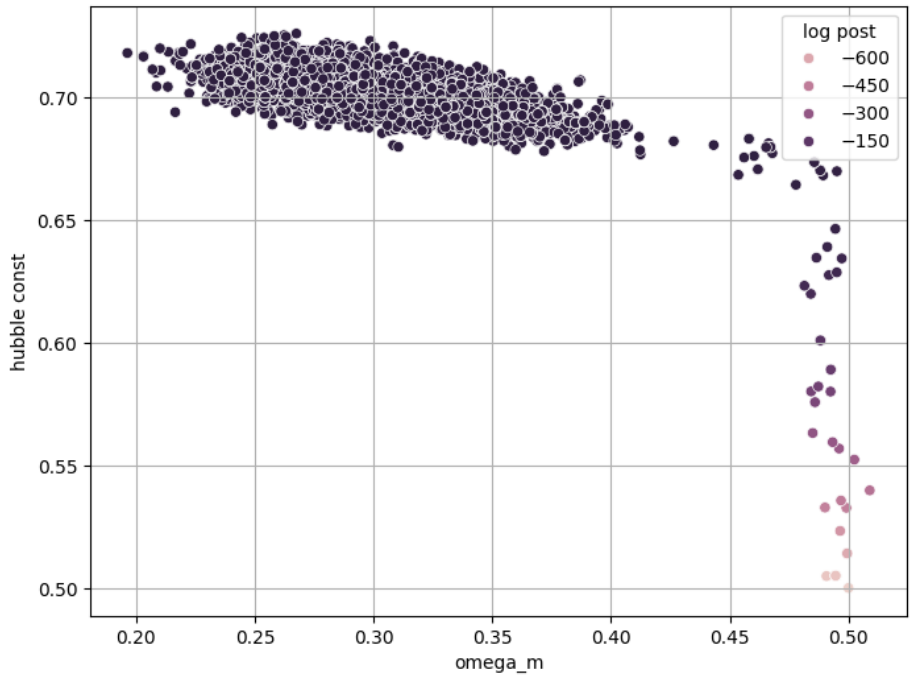


Figure 2.1: Sampled points from our model distribution under Metropolis Hastings Algorithm

We can then compute estimates for the Hubble constant $h$ and Matter density $\Omega_m$ of the universe estimates. These are:

$$\text{Sample Mean: } (h, \Omega_m) = (0.702948, 0.296688) \tag{2.1}$$

$$\text{Sample Median: } (h, \Omega_m) = (0.703071, 0.294928) \tag{2.2}$$

$$\text{Sample Mode: } ((h, \Omega_m) = (0.703354, 0.294342) \tag{2.3}$$

The covariance matrix is:

$$c = \begin{pmatrix} 0.000049 & -0.000127 \\ -0.000127 & 0.000959 \end{pmatrix} \tag{2.4}$$

Achieving what we set out to do!

### 2.2.1 Effect of choosing a different prior

**How is the posterior distribution affected by choosing a different prior for model parameters? How does the prior then influence our estimates for these parameters?**

Instead of using flat priors we try using a Gaussian prior for the Hubble parameter $h$ (we keep flat priors for $\Omega_m$). We try a good Gaussian prior for $h$ with mean 0.738 and standard deviation 0.024 and a bad Gaussian prior with mean 0.2 and standard deviation 0.01. The plots for the sampling of the resulting posterior distribution are

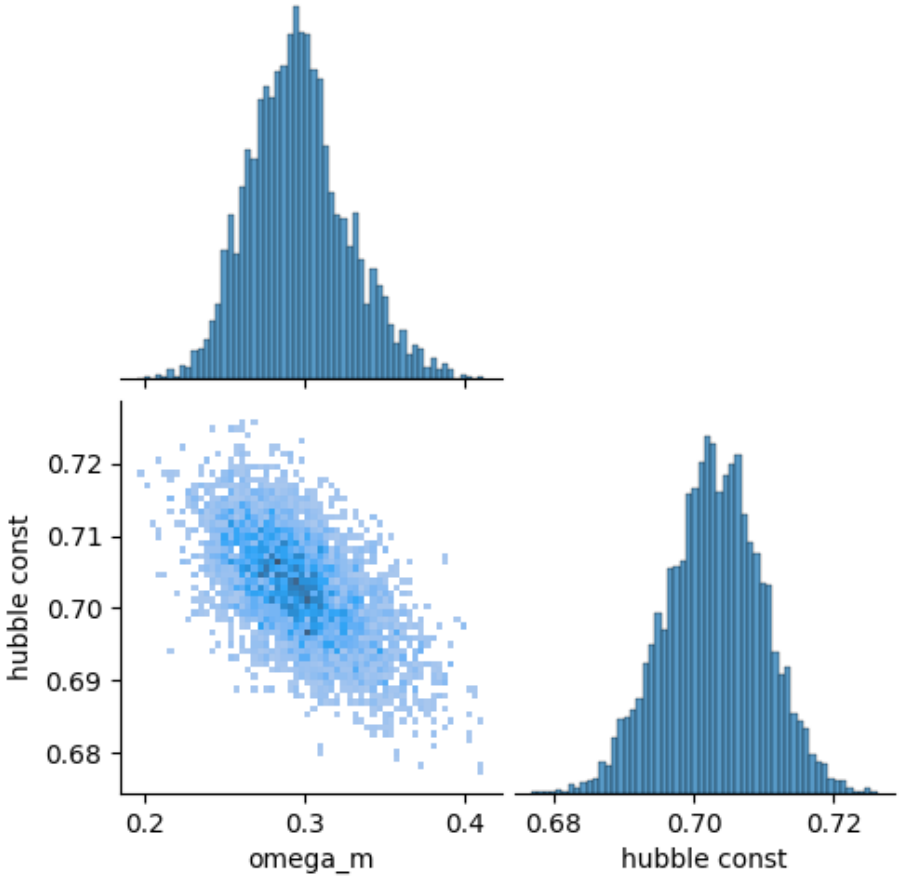


Figure 2.2: Model distribution after discarding burn in.

given in Figure 2.3. We see that a good Gaussian prior works in tandem with the data and increases the speed of convergence and leads to a resulting stationary distribution with lower covariance i.e. lower uncertainty in our estimates. On the other hand, a bad prior increases the length of the burn in phase and even when the data finally prevails leads to lasting bias. The resulting distribution is no longer peaked at the true values.

## 2.3 Hamiltonian Monte Carlo Algorithm

We also investigated a different sampling algorithm: the Hamiltonian Monte Carlo Algorithm. We won't go into the details of this algorithm but the reader can refer to the companion notebook. It works by defining a Hamiltonian consisting of the standard kinetic term and a potential term given by the posterior 'landscape' e.g. $U = -\log \mathcal{L}$. The random walk's behaviour is then stochastic dynamical and follows the posterior landscape. (the random walk is like a thermally excited particle (bead of water say) jostling around in a potential well (a glass). It spends most its time in the low potential (high probability) areas but can some times jump in and out when

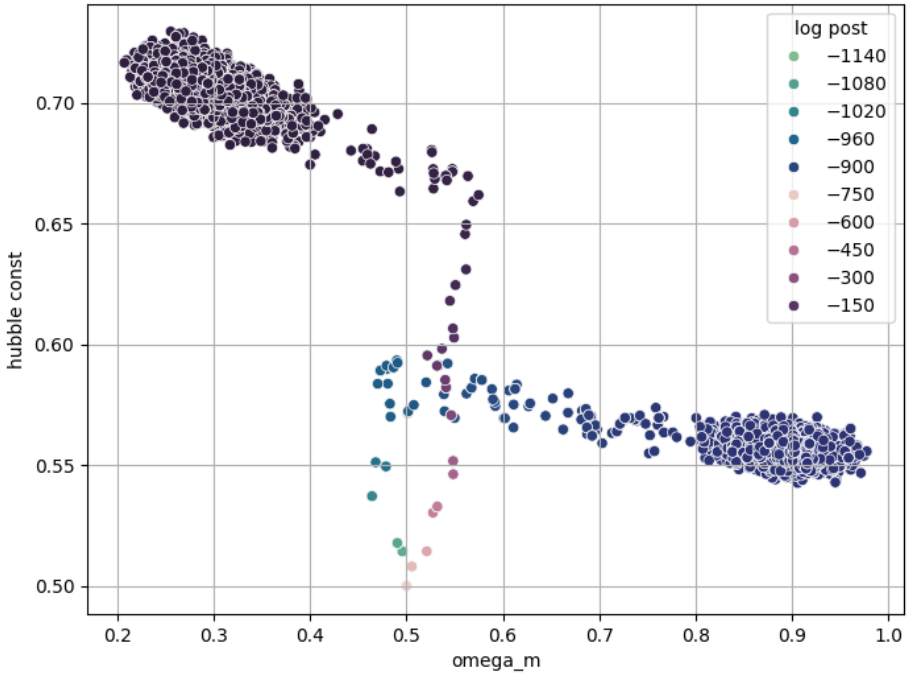


Figure 2.3: Sampled points from our model distribution with a good and bad Gaussian prior.

it receives enough kinetic energy.

rather than implement this model using the original model posterior 1.9 and taking derivatives of this complicated function of the parameters. Instead we form a new estimate of the posterior distribution from the results of the Metropolis Hastings Sampling. This leverages the previous work we have done and is an example of Bayesian Hierarchical Modelling. The new posterior distribution in parameter space is a bivariate normal distribution with means and covariance given my our previous estimates

$$U = \frac{1}{2}(\theta - \theta^*)_\alpha c^{-1}_{\alpha\beta}(\theta - \theta^*)_\beta \tag{2.5}$$

which can be easily differentiated symbolically to find the stochastic Hamiltonian dynamics.

The resulting plot of the HMC sample points is presented in Figure 2.4.

The resulting estimates are:

$$\text{Sample Mean: } (h, \Omega_m) = (0.703177, 0.297000) \tag{2.6}$$

$$\text{Sample Median: } (h, \Omega_m) = (0.703320, 0.298236) \tag{2.7}$$

The covariance matrix is:

$$c = \begin{pmatrix} 0.000044 & -0.000088 \\ -0.000088 & 0.000672 \end{pmatrix} \tag{2.8}$$

We note that the HMC algorithm has a longer burn in and is very sensitive to the internal parameters associated to the 'leapfrog integration step' used to compute the dynamics.

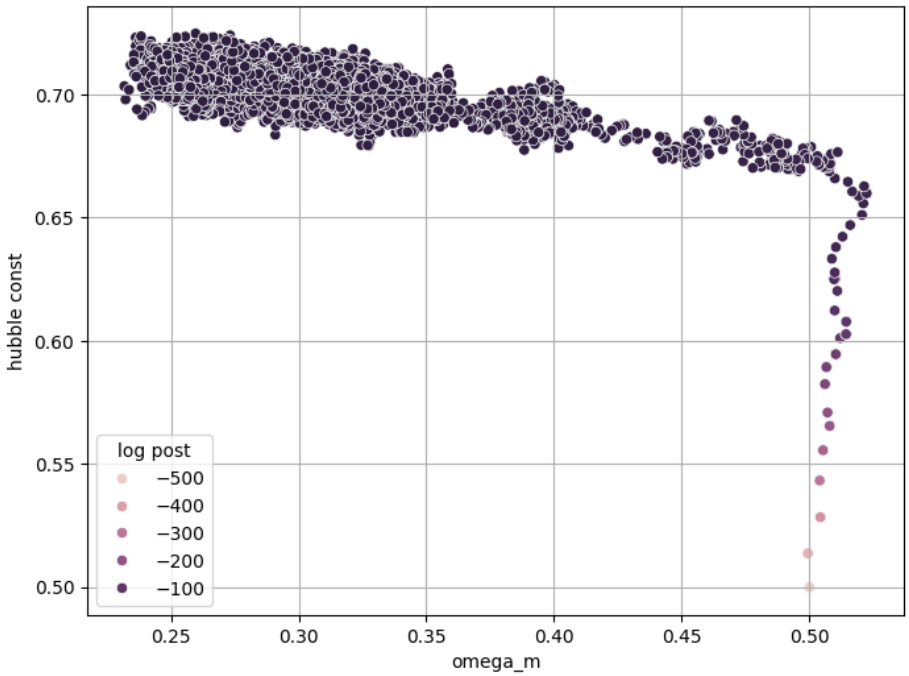


Figure 2.4: Sampled points from our hierarchical model using Hamiltonian Monte Carlo sampling.

# Chapter 3

# Conclusions

We used Bayesian methods and theoretical laws to model observations of the brightness and colour of supernovae. We then estimate the Hubble parameter and matter content of the universe.

Bayesian methods are well suited to cosmological modelling due to the importance of correctly propagating uncertainties and the at times small datasets. They lead to interpretable models which are physically motivated.

To convert analytic models into numerical results and best estimates we use simulation in the form of Markov Chain Monte Carlo Methods used to sample our probabilistic model. We used the Metropolis Hastings algorithm and the Hamiltonian Monte Carlo algorithm comparing the two and the influence of different priors on our estimates.