Supernova data analysis exercise

Florent Leclercq[a)]
CNRS & Sorbonne Université, UMR 7095, Institut d'Astrophysique de Paris, 98 bis boulevard Arago, F-75014 Paris, France

(Dated: 27 March 2025)

The goal of this hands-on exercise is to get familiar with the basics of Markov Chain Monte Carlo (MCMC) methods, by to writing an MCMC code to infer cosmological parameters from type Ia supernovæ data. It covers:

- simulating from a given data model,
- writing a basic likelihood for cosmological data,
- writing a Metropolis-Hastings sampler,
- diagnosing a Markov Chain (trace plots, tuning the proposal distribution),
- checking convergence and efficiency (Gelman-Rubin test, effective sample size),
- changing the prior for existing chains via importance sampling,
- writing and tuning a basic Hamiltonian Markov Chain Monte Carlo sampler,
- replacing part of a data model by an emulator.

I. BACKGROUND: COSMOLOGY WITH SUPERNOVÆ IA

This exercise is based around the relationship between the apparent brightness and distance of a special type of astronomical objects that have a known intrinsic brightness (so-called 'standard candles'). This relationship is useful for cosmological physics, as the theoretical relation depends on the amount of matter, the amount of dark energy, and the curvature of the Universe. Type Ia supernovæ are thought to arise in binary systems in which one of the stars is a white dwarf that collapses when reaching the Chandrasekhar limit, and are thus standard candles (after calibration). The resulting 'supernova Hubble diagram' (Figure 1) was famously used to infer the acceleration of the expansion of the Universe in 1998, culminating in the award of the 2011 Nobel prize in physics to Schmidt, Perlmutter and Riess.

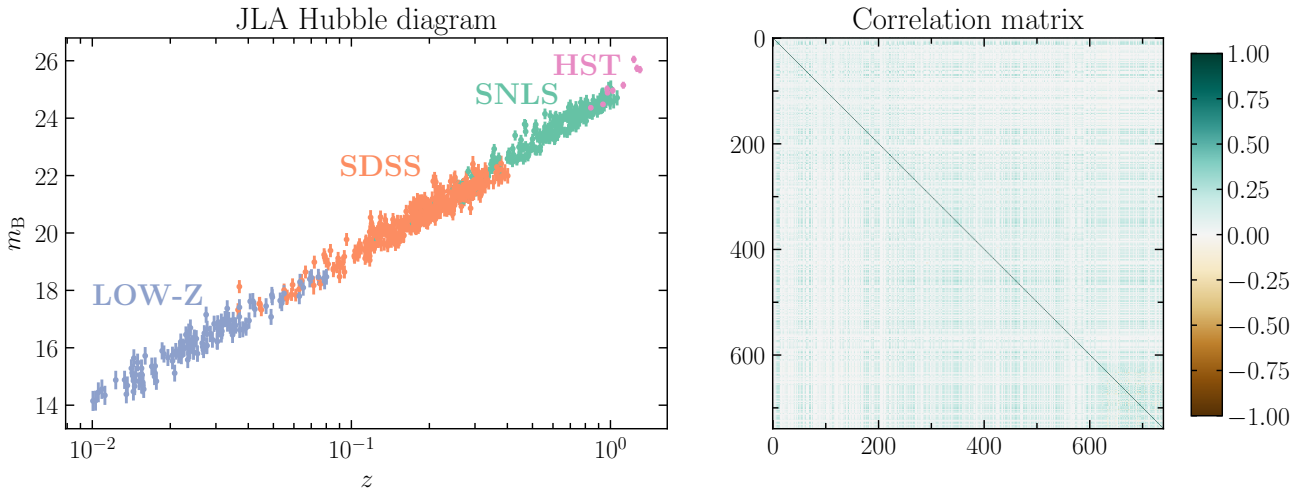

FIG. 1. *Left panel.* JLA Hubble diagram, representing the observed B-band apparent magnitudes m_B of 740 type Ia supernovae as a function of their redshift. The error bars represented correspond to $2\Delta m_B$, where Δm_B is included in the JLA catalogue. The different colours correspond to the different observational programmes used in the compilation. *Right panel.* Correlation matrix of the observed apparent magnitudes, taking into account statistical and various systematic uncertainties (see Betoule *et al.*, 2014, section 5.5 for details on the construction of the covariance matrix). Figure taken from Leclercq (2018).

[a)]Electronic mail: florent.leclercq@iap.fr; https://www.florent-leclercq.eu/; ORCID: 0000-0002-9339-1404

II. THEORY AND PARAMETERS

The flux-luminosity relationship for a supernova of luminosity L is given by

$$F = \frac{L}{4\pi D_L^2}, \tag{1}$$

which defines the cosmological *luminosity distance* D_L. In the standard cosmological model, D_L is given as a function of redshift z by (Hogg, 1999)

$$D_L = \frac{(1+z)\mathrm{c}}{H_0\sqrt{|\Omega_k|}} S_k(r), \tag{2}$$

where

$$r(z) = \sqrt{|\Omega_k|}\int_0^z \frac{\mathrm{d}z'}{\sqrt{\Omega_\Lambda + \Omega_k(1+z')^2 + \Omega_\mathrm{m}(1+z')^3}}, \tag{3}$$

and

$$S_k(r) = \begin{cases} \sinh\left[r(z)\right] & \text{if } \Omega_k > 0, \\ r(z) & \text{if } \Omega_k = 0, \\ \sin\left[r(z)\right] & \text{if } \Omega_k < 0, \end{cases} \tag{4}$$

depending on the sign of the Universe's curvature density parameter $\Omega_k \equiv \Omega_\mathrm{m} + \Omega_\Lambda - 1$. There are three free parameters in the problem: the matter density parameter Ω_m, the dark energy (or cosmological constant) density parameter Ω_Λ, and the Hubble parameter H_0. Instead of H_0, it is convenient to work with the dimensionless parameter h, defined such that $H_0 \equiv 100h$ km s^{-1} Mpc^{-1}. For anyone unfamiliar with cosmology, Ω_m, Ω_Λ, and h are somewhere between 0 and 1.

For a flat Universe ($\Omega_k = 0$), there are only two free parameters, and the expression for D_L simplifies to

$$D_L(z) = \frac{\mathrm{c}}{H_0}(1+z)\int_0^z \frac{\mathrm{d}z'}{\sqrt{1-\Omega_\mathrm{m}+\Omega_\mathrm{m}(1+z')^3}}. \tag{5}$$

To avoid evaluating the integral, it is possible to use an accurate fitting formula (valid for a flat Universe only), given by (Pen, 1999)

$$D_L(z) = \frac{\mathrm{c}}{H_0}(1+z)\left[\eta(1,\Omega_\mathrm{m}) - \eta\left(\frac{1}{1+z},\Omega_\mathrm{m}\right)\right], \tag{6}$$

where

$$\eta(a,\Omega_\mathrm{m}) \equiv 2\sqrt{s^3+1}\left(\frac{1}{a^4} - 0.1540\,\frac{s}{a^3} + 0.4304\,\frac{s^2}{a^2} + 0.19097\,\frac{s^3}{a} + 0.066941\,s^4\right)^{-1/8} \tag{7}$$

and $s^3 \equiv (1-\Omega_\mathrm{m})/\Omega_\mathrm{m}$. This is accurate to better than 0.4% for $0.2 \le \Omega_\mathrm{m} \le 1$.

Fluxes of astronomical objects are usually expressed in terms of apparent magnitudes m, where $m \equiv -2.5\log_{10} F + \text{constant}$ (such that a difference of 5 magnitudes corresponding to a brightness factor of 100). The absolute magnitude M is the value of m if the source is at a luminosity distance of 10 pc, i.e.

$$M \equiv m - 5\log_{10}\frac{D_L}{10\text{ pc}}. \tag{8}$$

In the case of standard candles, both the apparent magnitude and absolute magnitude are known;[1] it is therefore possible to estimate the distance using the *distance modulus* $\mu \equiv m - M$. With D_L in Mpc, this is

$$\mu = 25 - 5\log_{10} h + 5\log_{10}\left(\frac{D_L^*}{\text{Mpc}}\right), \tag{9}$$

where the Hubble parameter has been factored out of D_L, i.e. $D_L^* \equiv D_L(h=1)$.

---

[1] There is here a simplification in the exercise: we assume that we know the absolute magnitude M (or intrinsic luminosity L) of type Ia supernovæ, but in fact, unless we have at least one supernova with a known distance, we don't. More realistically, M and h are degenerate, since M is set from low-redshift supernovæ for which we assume Hubble's law to give the distance. For the purpose of this exercise, we will ignore this complication.

III. DATA

The full data file from the 'JLA' sample (Betoule *et al.*, 2014) consists of data from 740 supernovæ.[2]

For the purpose of this exercise, we will use a pre-processed file where the data are averaged together in 31 narrow bins of redshift containing supernovæ with $z < 1.3$, to give a measurement of the distance modulus μ for each bin. The data file `jla_mub.txt` contains $N = 31$ pairs of (z, μ). The 31×31 covariance matrix C of the data, provided as a list of numbers in an obvious order, is in the file `jla_mub_covmatrix.txt`.

IV. PRELIMINARY EXERCISE: SUPERNOVA SIMULATIONS

1. Setup your environment. You need to have access to some programming language and be comfortable using it. A convenient choice is python as provided by Jupyter Notebooks (off-line or in Google Colab, for example).
2. Plot the theoretical distance modulus formula for a flat Universe, for parameter values $\Omega_{\rm m} \in \{0.2, 0.3, 0.4, 0.5\}$, for the range $0 < z < 2$. Set $h = 0.7$.
3. Plot the observational data (in the file `jla_mub.txt`) on the graph.
4. Generate some simulated data: make a random set of 20 supernovæ with redshifts drawn uniformly in the range $0 < z < 2$. Assuming a flat Universe with $\Omega_{\rm m} = 0.3$ and $h = 0.7$, compute the expected μ for each of them.
5. Using a Gaussian random number generator routine, add an error to μ for each supernova, drawn from a Gaussian distribution with mean 0 and standard deviation $\sigma = 0.1$ (magnitudes).
6. Plot these data on a graph, with error bars, and overplot theory curves for $\Omega_{\rm m} = 0.3$ and $h \in \{0.6, 0.7, 0.8\}$. Data points should scatter around the correct theoretical curve.

V. METROPOLIS-HASTINGS SAMPLING

1. First assume a flat Universe and use the fitting formula for D_L^*. Assume that the observational errors are Gaussian, i.e. that the likelihood of the problem is

$$L \propto \exp\left\{-\frac{1}{2}\sum_{i,j=1}^{N}\left[\mu_i - \mu_{\rm th}(z_i)\right] C_{ij}^{-1}\left[\mu_j - \mu_{\rm th}(z_j)\right]\right\}, \tag{10}$$

   where $\mu_{\rm th}$ is the theoretical value of the distance modulus. Write a routine giving the log-likelihood as a function of parameters $\Omega_{\rm m}$ and h. What is the log-likelihood for $(\Omega_{\rm m}, h) = (0.3, 0.7)$?
2. Assume uniform priors on the parameters and write a Metropolis-Hastings sampler for the likelihood. For the proposal distribution, you may like to start with a very simple 'top-hat' distribution, where the new point is selected from a rectangular region centred on the old point. You may also use a Gaussian proposal distribution for each parameter.
   Reminder: the Metropolis-Hastings algorithm is given below. The *Hastings ratio* is defined by

Algorithm 1: Metropolis-Hastings sampler

```
begin
    initialise x_(0);
    for i = 1 to n do
        x* ↶ q(x*|x) (proposal distribution);
        α ↶ U(0,1) (uniform distribution);
        if α < min[1, r(x, x*)] then
            x_(i) = x*;
        else
            x_(i) = x_(i−1);
        end
    end
    return (x_(0), ..., x_(n));
end
```

[2] See `http://supernovae.in2p3.fr/sdss_snls_jla/ReadMe.html` for more details.

$$r(\boldsymbol{x}, \boldsymbol{x}^*) \equiv \frac{p(\boldsymbol{x}^*)}{p(\boldsymbol{x})} \frac{q(\boldsymbol{x}|\boldsymbol{x}^*)}{q(\boldsymbol{x}^*|\boldsymbol{x})}. \tag{11}$$

In the case of a symmetric proposal distribution (i.e. $q(\boldsymbol{x}^*|\boldsymbol{x}) = q(\boldsymbol{x}|\boldsymbol{x}^*)$, a *Metropolis update*), it simplifies to

$$r(\boldsymbol{x}, \boldsymbol{x}^*) = \frac{p(\boldsymbol{x}^*)}{p(\boldsymbol{x})}. \tag{12}$$

3. Produce a 'trace plot' for each parameter (i.e. plot the value of Ω_m and h as a function of the position in the chain), for a maximum of 1000 trials. Compute the acceptance ratio of your chain. Estimate visually the length of the burn-in phase.

4. Explore visually the chain when you have (a) a very small proposal distribution, (b) an intermediate proposal distribution and (c) a very large proposal distribution. Compute the acceptance ratio for each situation. What do you conclude?

5. Once you have settled on a reasonable proposal distribution, run five different chains with different starting points. Produce the trace plots of the different chains on the same graph, for each parameter. Can you make a more precise statement about the burn-in phase?

6. Compute the Gelman-Rubin statistic $\widehat{R}$ (Gelman & Rubin, 1992) using your five chains. Deduce roughly how long the chains should be for convergence.
Reminder (see Gelman *et al.*, *Bayesian Data Analysis* (third edition), p. 284-285): if m is the number of chains and n is the length of chains, one can define a "between" chains variance as

$$B \equiv \frac{n}{m-1} \sum_{j=1}^{m} \left(\bar{\psi}_{.j} - \bar{\psi}_{..}\right)^2 \quad \text{where} \quad \bar{\psi}_{.j} = \frac{1}{n} \sum_{i=1}^{n} \psi_{ij} \quad \text{and} \quad \bar{\psi}_{..} = \frac{1}{m} \sum_{j=1}^{m} \bar{\psi}_{.j}, \tag{13}$$

and a "within" chains variance:

$$W \equiv \frac{1}{m} \sum_{j=1}^{m} s_j^2 \quad \text{where} \quad s_j^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left(\psi_{ij} - \bar{\psi}_{.j}\right)^2. \tag{14}$$

Estimators of the marginal posterior variance of the estimand are

$$\widehat{\mathrm{var}}^- \equiv W, \tag{15}$$

which underestimates the variance, and

$$\widehat{\mathrm{var}}^+ \equiv \frac{n}{n-1} W + \frac{1}{n} B, \tag{16}$$

which overestimates the variance. The potential scale reduction factor (PSRF) is defined as

$$\widehat{R} \equiv \sqrt{\frac{\widehat{\mathrm{var}}^+}{\widehat{\mathrm{var}}^-}}, \tag{17}$$

and the Gelman-Rubin convergence test consists in checking that $\widehat{R} \to 1$ as $n \to \infty$.

7. Compute the mean value of the parameters under the likelihood distribution, and their variances and covariance. On a 2D plot, show your samples and overplot 68%, 95%, and 99% likelihood contours.

VI. IMPORTANCE SAMPLING

Consider a non-flat prior, so that the target posterior distribution now differs from the likelihood. It is still possible to sample from the likelihood (as you have been doing in the previous section) and construct the posterior by weighting the samples with the prior to get the target posterior distribution. This is an example of *importance sampling*, where we sample from a different distribution from the one we eventually want.

1. Apply a Gaussian prior on the Hubble parameter h to your chain, with mean 0.738 and standard deviation 0.024, to get the weights for your existing samples.

2. On the same 2D plot, show your samples coloured according to their weight, likelihood contours, and posterior contours.

3. Compute the mean Ω_m and h with and without the prior.

VII. HAMILTONIAN MONTE CARLO SAMPLING

Extend to perform Hamiltonian Monte Carlo (HMC). The algorithm is (e.g. Hajian, 2007)

Algorithm 2: Hamiltonian Monte Carlo sampler

begin
  initialise $\boldsymbol{x}_{(0)}$;
  for $i = 1$ *to* n **do**
    $\boldsymbol{p} \curvearrowleft \mathcal{N}(0,1)$ (normal distribution);
    $(\boldsymbol{x}^*_{(0)}, \boldsymbol{p}^*_{(0)}) = (\boldsymbol{x}_{(i-1)}, \boldsymbol{p})$;
    for $j = 1$ *to* $N_{\rm steps}$ **do**
      make a leapfrog move: $(\boldsymbol{x}^*_{(j-1)}, \boldsymbol{p}^*_{(j-1)}) \to (\boldsymbol{x}^*_{(j)}, \boldsymbol{p}^*_{(j)})$;
    end
    $(\boldsymbol{x}^*, \boldsymbol{p}^*) = (\boldsymbol{x}_{(N_{\rm steps})}, \boldsymbol{p}_{(N_{\rm steps})})$;
    $\alpha \curvearrowleft \mathcal{U}(0,1)$ (uniform distribution);
    if $\alpha < \min\left(1, \exp\left\{-\left[H(\boldsymbol{x}^*, \boldsymbol{p}^*) - H(\boldsymbol{x}^*_{(0)}, \boldsymbol{p}^*_{(0)})\right]\right\}\right)$ **then**
      $\boldsymbol{x}_{(i)} = \boldsymbol{x}^*$;
    else
      $\boldsymbol{x}_{(i)} = \boldsymbol{x}_{(i-1)}$;
    end
  end
  return $(\boldsymbol{x}_{(0)}, \ldots, \boldsymbol{x}_{(n)})$;
end

The Hamiltonian of the system is

$$H = U + K, \tag{18}$$

with a kinetic energy $K \equiv \boldsymbol{p}^\intercal \mathbf{M}^{-1} \boldsymbol{p}/2$ (for a given mass matrix $\mathbf{M}$) and potential energy $U \equiv -\ln P - \ln L$, where P is the prior. The leapfrog or kick-drift-kick algorithm (which is forward-backward symmetric, as required to preserve detailed balance) is, for any time step t,

$$p_k\left(t + \frac{\epsilon}{2}\right) = p_k(t) - \frac{\epsilon}{2} \left.\frac{\partial U}{\partial x_k}\right|_{\boldsymbol{x}(t)} \tag{19}$$

$$x_k(t+\epsilon) = x_k(t) + \epsilon\, p_k\left(t + \frac{\epsilon}{2}\right) \tag{20}$$

$$p_k\left(t+\epsilon\right) = p_k\left(t + \frac{\epsilon}{2}\right) - \frac{\epsilon}{2} \left.\frac{\partial U}{\partial x_k}\right|_{\boldsymbol{x}(t+\epsilon)}. \tag{21}$$

1. The partial derivative of U with respect to h is easy to compute, but the partial derivative with respect to $\Omega_{\rm m}$ is complicated. Decide if you want to use exact or approximate derivatives.

1a. Use an automatic differentiation package to obtain the exact derivatives of U. Sympy (`https://www.sympy.org/`) is one way to differentiate U automatically and produce the python code. Other options include Jax and autograd.

1b. Alternatively, it is possible to approximate U by a bivariate Gaussian distribution, with a mean $\bar{\boldsymbol{x}}$ and covariance matrix $\boldsymbol{\Sigma}$ estimated from the chains obtained with the Metropolis-Hastings MCMC code:

$$U \approx \frac{1}{2} \sum_{\alpha,\beta} (x - \bar{x})_\alpha \Sigma^{-1}_{\alpha\beta} (x - \bar{x})_\beta, \tag{22}$$

from which analytical derivatives $\partial U/\partial x_\alpha$ are easily obtained. Since it is approximate, the Hamiltonian H will not be exactly conserved, but the Metropolis update step sorts everything out. Implement routines to get the exact potential, the approximate potential, and its derivatives. Plot 2D surfaces of these quantities and visually check the quality of the approximation.

2. Implement a Hamiltonian Monte Carlo sampler and run a few chains. Try changing the starting point and visualising trajectories in parameter space. Diagnose the chains using trace plots and compute their acceptance ratio.

3. For each parameter, compute the correlation length of the chain and the effective sample size. Definitions and a python code to compute autocorrelation lengths can be found, for example, at `https://emcee.readthedocs.io/en/stable/tutorials/autocorr/`.

4. Try changing the step size ϵ and the mass matrix $\mathbf{M}$. What happens if one parameter is much 'heavier' than the other? Try changing the number of leapfrog steps $N_{\rm steps}$ and randomising it (for example by drawing $N_{\rm steps}$ uniformly in $[\![1, N_{\rm steps,max}]\!]$). For each case, diagnose the chains using trace plots, acceptance ratio, and correlation length. What do you conclude?

5. Compare the performance of Metropolis-Hastings MCMC and HMC by plotting the effective sample size versus the number of function evaluations. 'Function' here refers to either the posterior (or potential) or one of its derivatives (we assume for simplicity that evaluating the potential or one of its derivatives has the same numerical cost). You will have to modify your samplers to keep track of the number of calls to each function.

VIII. THREE-PARAMETER PROBLEM AND USE OF AN EMULATOR

Generalise to non-flat universes and include Ω_Λ as a free parameter:

1. You will need to perform the integral for $D_L^*(\Omega_{\rm m}, \Omega_\Lambda)$ numerically. What is the value of $D_L^*(h = 1, \Omega_{\rm m} = 0.3)$ (using the fitting formula), $D_L^*(h = 1, \Omega_{\rm m} = 0.3, \Omega_\Lambda = 0.7)$ (computing the integral), and $D_L^*(h = 1, \Omega_{\rm m} = 0.3, \Omega_\Lambda = 0.71)$?

2. You may find that computing the integral is too expensive to be put in a MCMC sampler. Write an emulator of the data model: pre-compute $D_L^*(\Omega_{\rm m}, \Omega_\Lambda)$ on a grid for $(\Omega_{\rm m}, \Omega_\Lambda)$ spanning $[0, 1]^2$, use a 2D bilinear interpolation routine to predict $D_L^*(\Omega_{\rm m}, \Omega_\Lambda)$, and replace the numerical integrator by the interpolator in your routine giving μ. You only need to do this at the N redshifts of the data vector.

3. Write 3D routines for your log-prior and log-likelihood. Write a 3D Metropolis-Hasting sampler.

4. Run and diagnose some Markov chains using trace plots. Produce a 'triangle plot' showing samples and 2D contours for each pair of parameters.

5. Compute the posterior covariance matrix and try to feed it back to the sampler as covariance of a Gaussian proposal distribution.

ACKNOWLEDGEMENTS

I thank Alan Heavens and colleagues of the SOC of the ICIC Data Analysis workshop 2021 for their own lectures, from which this document is largely inspired.

REFERENCES

(Betoule *et al.*, 2014) M. Betoule, R. Kessler, J. Guy, J. Mosher, D. Hardin, R. Biswas, P. Astier, P. El-Hage, M. Konig, S. Kuhlmann, *et al.*, *Improved cosmological constraints from a joint analysis of the SDSS-II and SNLS supernova samples*, Astron. & Astrophys. **568**, A22 (2014), arXiv:1401.4064.

(Gelman & Rubin, 1992) A. Gelman, D. B. Rubin, *Inference from Iterative Simulation Using Multiple Sequences*, Statistical Science **7**, 457 (1992).

(Hajian, 2007) A. Hajian, *Efficient cosmological parameter estimation with Hamiltonian MonteCarlo technique*, Phys. Rev. D **75**, 083525 (2007), arXiv:astro-ph/0608679.

(Hogg, 1999) D. W. Hogg, *Distance measures in cosmology*, ArXiv Astrophysics e-prints (1999), astro-ph/9905116.

(Leclercq, 2018) F. Leclercq, *Bayesian optimization for likelihood-free cosmological inference*, Phys. Rev. D **98**, 063511 (2018), arXiv:1805.07152.

(Pen, 1999) U.-L. Pen, *Analytical Fit to the Luminosity Distance for Flat Cosmologies with a Cosmological Constant*, Astrophys. J. Supp. **120**, 49 (1999), arXiv:astro-ph/9904172 [astro-ph].