Bayesian Models for Astrophysical Data

Using R, JAGS, Python, and Stan

JOSEPH M. HILBE

Arizona State University and Jet Propulsion Laboratory, California Institute of Technology

RAFAEL S. DE SOUZA

Eötvös Loránd University, Budapest

EMILLE E. O. ISHIDA

Université Clermont-Auvergne (Université Blaise Pascal), France

CAMBRIDGE
UNIVERSITY PRESS

10.11 Gaussian Model, ODEs, and Type Ia Supernova Cosmology

Type Ia supernovae (SNe Ia) are extremely bright transient events which can be used as standardizable candles for distance measurements in cosmological scales. In the late 1990s they provided the first evidence for the current accelerated expansion of the Universe (Perlmutter *et al.*, 1999; Riess *et al.*, 1998) and consequently the existence of dark energy. Since then they have been central to every large-scale astronomical survey aiming to shed light on the dark-energy mystery.

In the last few years, a considerable effort has been applied by the astronomical community in an attempt to popularize Bayesian methods for cosmological parameter inference, especially when dealing with type Ia supernovae data (e.g. Andreon, 2011; Ma *et al.*, 2016; Mandel *et al.*, 2011; Rubin *et al.*, 2015; Shariff *et al.*, 2015). Thus, we will not refrain from

[26] https://groups.google.com/forum/#!topic/stan-users/hn4W_p8j3fs

tackling this problem and showing how Stan can be a powerful tool to deal with such complex model.

At maximum brightness the observed magnitude of an SN Ia can be connected to its *distance modulus* μ through the expression

$$m_{\text{obs}} = \mu + M - \alpha x_1 + \beta c, \tag{10.21}$$

where m_{obs} is the observed magnitude, M is the magnitude, and x_1 and c are the stretch and color corrections derived from the SALT2 standardization (Guy *et al.*, 2007), respectively. To take into account the effect of the host stellar mass $M_\star$ on M and β, we use the correction proposed by Conley *et al.* (2011):

$$M = \begin{cases} M & \text{if } M_\star < 10^{10} M_\odot \\ M + \Delta M & \text{otherwise} \end{cases} \tag{10.22}$$

Considering a flat Universe, $\Omega_k = 0$, containing dark energy and dark matter, the cosmological connection can be expressed as

$$\mu = 5 \log_{10}\left(\frac{d_L}{10pc}\right), \tag{10.23}$$

$$d_L(z) = (1+z)\frac{c}{H_0}\int_0^z \frac{dz}{E(z)}, \tag{10.24}$$

$$E(z) = \sqrt{\Omega_m (1+z)^3 + (1-\Omega_m)(1+z)^{3(1+w)}}, \tag{10.25}$$

where d_L is the luminosity distance, c the speed of light, H_0 the Hubble constant, Ω_m the dark-energy density, and w the dark-energy equation of state parameter.

In what follows we will begin with a simplified version of this problem and, subsequently, guide the reader through implementations of further complexity.

10.11.1 Data

We used data provided by Betoule *et al.* (2014), known as the joint light-curve analysis (JLA) sample.[27] This is a compilation of data from different surveys which contains 740 high quality spectroscopically confirmed SNe Ia up to redshift $z \sim 1.0$.

10.11.2 The Statistical Model Formulation

Our statistical model will thus have one response variable (the observer magnitude, m_{obs}) and four explanatory variables (the redshift z, the stretch x_1, the color c, and the host galaxy mass M_{host}).

[27] The complete data set is available at http://supernovae.in2p3.fr/sdss_snlsz_jla/ReadMe.html

The complete statistical model can be expressed as

$$
\begin{aligned}
& m_{\text{obs};i} \sim Normal(\eta_i, \varepsilon) \\
& \eta_i = 25 + 5\log_{10}(dl_i(H_0, w, \Omega_m)) + M(\Delta M) - \alpha x_{1;i} + \beta c_i \\
& \epsilon \sim Gamma(10^{-3}, 10^{-3}) \\
& M \sim Normal(-20, 5) \\
& \alpha \sim Normal(0, 1) \\
& \beta \sim Normal(0, 10) \\
& \Delta M \sim Normal(0, 1) \\
& \Omega_m \sim Uniform(0, 1) \\
& H_0 \sim Normal(70, 5) \\
& i = 1, \ldots, N
\end{aligned}
\tag{10.26}
$$

where d_L is given by Equation 10.24 and M by Equation 10.22. We use conservative priors over the model parameters. These are not completely non-informative but they do allow a large range of values to be searched without putting a significant probability on non-physical values.

10.11.3 Running the Model in R using Stan

After carefully designing the model we must face the extra challenge posed by the integral in the luminosity–distance definition. Fortunately, Stan has a built-in ordinary differential equation (ODE) solver which provides a user-friendly solution to this problem (the book Andreon and Weaver, 2015, Section 8.12, shows how this can be accomplished using JAGS). However, it is currently not possible to access this feature using pystan,[28] so we take this opportunity to show how Stan can also be handled from R using the package `rstan`.

Stan's ODE solver is explained in detail in the manual Team Stan (2016, Section 44). We advise the reader to go through that section before using it in your research. Here we would merely like to call attention for a couple of important points.

- The ODE takes as input a function which returns the derivatives at a given time, or, as in our case, a given redshift. This function must have, as input, time (or redshift), system state (the solution at $t = 0$), parameters (input a list if you have more than one parameter), real data, and integer data, in that order. In cases where your problem does not require integer or real data, you must still input an empty list (see Code 10.26 below).
- You must define a zero point for time (redshift, `z0 = 0`) and the state of the system at that point (in our case, `E0 = 0`). These must be given as input data in the dictionary to be passed to Stan. We strongly advise the reader to play a little with a set of simulated toy data before jumping into a real scientific scenario, so that you can develop an intuition about the behavior of the function and about the initial parameters of your model.

[28] This might not be a problem if you are using a pystan version higher than 2.9.0.

Our goal with this example is to provide a clean environment where the ODE solver role is highlighted. A more complex model is presented subsequently.

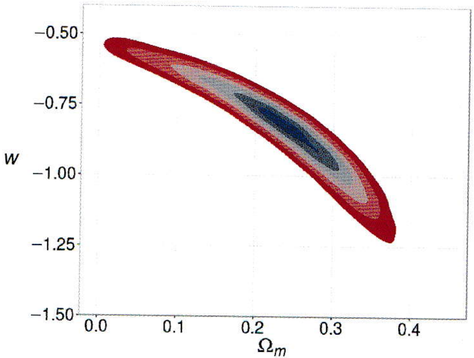
Code 10.26 Bayesian normal model for cosmological parameter inference from type Ia supernova data in R using Stan.

```
==========================================================
library(rstan)

# Preparation
# Set initial conditions
z0 = 0                                  # initial redshift
E0 = 0                                  # integral(1/E) at z0

# physical constants
c = 3e5                                 # speed of light
H0 = 70                                 # Hubble constant

# Data
# Read data
data <- read.csv("~/data/Section_10p11/jla_lcparams.txt",header=T)

# Remove repeated redshift
data2 <- data[!duplicated(data$zcmb),]

# Prepare data for Stan
nobs <- nrow(data2)                             # number of SNe
index <- order(data2$zcmb)                      # sort according to redshift
ObsMag <- data2$mb[index]                       # apparent magnitude
redshift <- data2$zcmb[index]                   # redshift
color <- data2$color[index]                     # color
x1 <- data2$x1[index]                           # stretch
hmass <- data2$m3rdvar[index]                   # host mass

stan_data  <- list(nobs = nobs,
                   E0 = array(E0,dim=1),
                   z0 = z0,
                   c = c,
                   H0 = H0,
                   obs_mag = ObsMag,
                   redshift = redshift,
                   x1 = x1,
                   color = color,
                   hmass = hmass)
 # Fit
stan_model="
functions {
     /**
     * ODE for the inverse Hubble parameter.
     * System State E is 1 dimensional.
     * The system has 2 parameters theta = (om, w)
     *
     * where
     *
     *   om:        dark matter energy density
     *   w:         dark energy equation of state parameter
     *
```

```
     * The system redshift derivative is
     *
     * d.E[1] / d.z  =
     *  1.0/sqrt(om * pow(1+z,3) + (1-om) * (1+z)^(3 * (1+w)))
     *
     * @param z redshift at which derivatives are evaluated.
     * @param E system state at which derivatives are evaluated.
     * @param params parameters for system.
     * @param x_r real constants for system (empty).
     * @param x_i integer constants for system (empty).
     */
     real[] Ez(real z,
               real[] H,
               real[] params,
               real[] x_r,
               int[] x_i) {
           real dEdz[1];

           dEdz[1] = 1.0/sqrt(params[1]*(1+z)^3
                     +(1-params[1])*(1+z)^(3*(1+params[2])));

           return dEdz;
     }
}
data {
    int<lower=1> nobs;                // number of data points
    real E0[1];                       // integral(1/H) at z=0
    real z0;                          // initial redshift, 0
    real c;                           // speed of light
    real H0;                          // Hubble parameter
    vector[nobs] obs_mag;             // observed magnitude at B max
    real x1[nobs];                    // stretch
    real color[nobs];                 // color
    real redshift[nobs];              // redshift
    real hmass[nobs];                 // host mass
}
transformed data {
      real x_r[0];                    // required by ODE (empty)
      int x_i[0];
}
parameters{
      real<lower=0, upper=1> om;      // dark matter energy density
      real alpha;                     // stretch coefficient
      real beta;                      // color coefficient
      real Mint;                      // intrinsic magnitude
      real deltaM;                    // shift due to host galaxy mass
      real<lower=0> sigint;           // magnitude dispersion
      real<lower=-2, upper=0> w;      // dark matter equation of state
                                         parameter
}
transformed parameters{
      real DC[nobs,1];               // co-moving distance
      real pars[2];                  // ODE input = (om, w)
      vector[nobs] mag;              // apparent magnitude
      real dl[nobs];                 // luminosity distance
      real DH;                       // Hubble distance = c/H0
```

```
        pars[1] = om;
        pars[2] = w;
        DH = (c/H0);

        # Integral of 1/E(z)
        DC = integrate_ode_rk45(Ez, E0, z0, redshift, pars,  x_r, x_i);

        for (i in 1:nobs) {
             dl[i] = DH * (1 + redshift[i]) * DC[i, 1];
             if (hmass[i] < 10) mag[i] = 25 + 5 * log10(dl[i])
                                         + Mint - alpha * x1[i] + beta
                                         * color[i];

             else
                 mag[i] = 25 + 5 * log10(dl[i])
                            + Mint + deltaM - alpha * x1[i] + beta * color[i];
        }
}
model {
        # Priors and likelihood
        sigint ~ gamma(0.001, 0.001);
        Mint ~ normal(-20, 5.);
        beta ~ normal(0, 10);
        alpha ~ normal(0, 1);
        deltaM ~ normal(0, 1);

        obs_mag ~ normal(mag, sigint);
}
"

# Run MCMC
fit <- stan(model_code = stan_model,
                data = stan_data,
                seed = 42,
                chains = 3,
                iter = 15000,
                cores = 3,
                warmup = 7500
)

# Output

# Summary on screen
print(fit,pars=c("om", "Mint","alpha","beta","deltaM", "sigint"),
                 intervals=c(0.025, 0.975), digits=3)
==================================================
          mean se_mean    sd    2.5%      25%     50%     75%   97.5%  n_eff Rhat
om       0.232   0.001 0.091   0.036    0.172   0.243   0.300   0.380   7423    1
Mint   -19.059   0.000 0.017 -19.094  -19.071 -19.059 -19.048  19.027   8483    1
w       -0.845   0.002 0.180  -1.237   -0.960  -0.829  -0.708  -0.556   7457    1
alpha    0.119   0.000 0.006   0.106    0.114   0.119   0.123   0.131  16443    1
beta     2.432   0.001 0.071   2.292    2.384   2.432   2.480   2.572  16062    1
deltaM  -0.031   0.000 0.013  -0.055   -0.039  -0.031  -0.022  -0.006  11938    1
sigint   0.159   0.000 0.004   0.151    0.156   0.159   0.162   0.168  16456    1
```

The results are consistent with those reported by Ma *et al.* (2016, Section 4), who applied Bayesian graphs to the same data. A visual representation of posteriors over Ω_m and w is shown in Figure 10.23, left-hand panel.



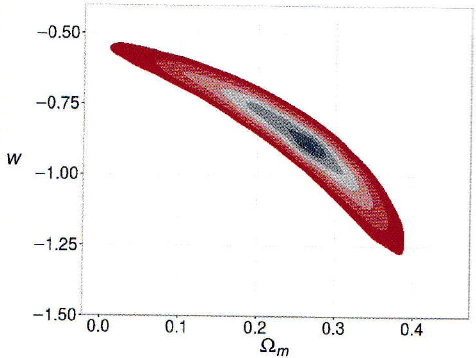


Figure 10.23 Joint posterior distributions over the dark-matter energy density Ω_w and equation of state parameter w obtained from a Bayesian Gaussian model applied to the JLA sample. Left: the results without taking into account errors in measurements. Right: the results taking into account measurement errors in color, stretch, and observed magnitude.