Applied Bayesian Modeling module 10:
Using (r)stan and prior choice

Leontine Alkema, lalkema@umass.edu
Fall 2022

Outline

- We are going to discuss how to fit your own models in (r)Stan!
- This slide set: Introduce main set up, and key references to more detailed materials
  - Along the way: choice of priors
- Intro to (r)Stan is a start only;
  get the main set-up, then learn by doing in your project/research (or stick to using brm)
- Next modules: Bayesian workflow (more on model checking and MCMC diagnostics)

Stan: recap

- Stan = programming language, commonly used as MCMC sampler for Bayesian analyses
- Various interfaces available
  - So far, we used `brms` (which takes in model formulas and has a lot of built-in defaults),
  - Next: use `rstan` and write our own model specs

How to fit a model using rstan: steps

- Fit model using `stan` function call. This needs
  - Model in Stan code
  - Data inputs that correspond to data used in Stan model
- Work with the `rstan`-output: samples, diagnostics related to sampling

Writing a model in Stan

- A Stan program is divided into coding blocks.
- Essential: data, parameters, model
- Optional: transformed parameters, generated quantities, functions

Example 1: estimate mean radon

- Estimate μ and σ when $y_i|\mu, \sigma \sim N(\mu, \sigma^2)$
- In R-studio, when you go to file, and open a new stan model file, you'll see a model, with parts copied here

```
// The input data is a vector 'y' of length 'N'.
data {
   int<lower=0> N;
   vector[N] y;
}

// The parameters accepted by the model. Our model
// accepts two parameters 'mu' and 'sigma'.
parameters {
   real mu;
   real<lower=0> sigma;
}
```

Example 1: estimate mean radon (ctd)

- Estimate μ and σ when $y_i|\mu, \sigma \sim N(\mu, \sigma^2)$

```
data {
    int<lower=0> N;
    vector[N] y;
}
parameters {
    real mu;
    real<lower=0> sigma;
}
// The model to be estimated. We model the output
// 'y' to be normally distributed with mean 'mu'
// and standard deviation 'sigma'.
model {
    y ~ normal(mu, sigma);
}
```

Choice of priors

- Note that in default stan model, no priors for μ or σ are specified (discussed next). We can add to the model block

$$\mu \sim N(0,1), \tag{1}$$
$$\sigma \sim N^{+}(0,1), \tag{2}$$

  where N^{+} refers to a normal density, truncated at 0.

```
model {
   mu ~ normal(0, 1); // adding a prior for mu
   sigma ~ normal(0, 1); // adding a prior for sigma
   y ~ normal(mu, sigma);
}
```

Model fitting

- Specify the data and call `stan`.
- See examples of what to do with stan fitted object in Rmd.

```
stan_dat <- list(y = dat$y, N = length(dat$y))
fit <- stan(file = 'module10_stan_mean_addpriors.stan',
  data = stan_dat)
```

Example 2: multilevel regression model for radon

- Model to fit:

$$
\begin{aligned}
y_i|\mu_i, \sigma &\sim N(\mu_i, \sigma_y^2), && (3)\\
\mu_i &= \mu_\alpha + \eta_{j[i]} + \beta x_i, && (4)\\
\eta_j &\sim N(0, \sigma_\alpha^2), && (5)\\
\mu_\alpha &\sim N(0, 1), && (6)\\
\beta &\sim N(0, 1), && (7)\\
\sigma_y &\sim N^+(0, 1), && (8)\\
\sigma_\alpha &\sim N^+(0, 1), && (9)
\end{aligned}
$$

where $j[i]$ refers to county and x_i to floor indicator.

Example 2: stan model

- Data and parameter block

```
data {
   int<lower=1> N;
   int<lower=1> J; // number of counties
   int<lower=1,upper=J> county_id[N];
   vector[N] x;
   vector[N] y;
}
parameters {
   vector[J] eta; // stan-preferred non-centered parametrizat
   real beta;
   real mu_alpha;
   real<lower=0> sigma_y;
   real<lower=0> sigma_alpha;
}
```

Example 2: stan model

- model block

```
model {
  vector[N] mu;
  //comment: vectorized statements are more efficient in stan
  // but using a loop here to make code easy to follow
  for (i in 1:N)
    mu[i] = mu_alpha + eta[county_id[i]] + x[i] * beta;
  eta ~ normal(0, sigma_alpha);
 // prior choice to be discussed
  mu_alpha ~ normal(0, 1);
  beta ~ normal(0, 1);
  sigma_y ~ normal(0, 1);
  sigma_alpha ~ normal(0, 1);
  y ~ normal(mu, sigma_y);
}
```

How to start building your own Stan models?

- Consider writing your own simple models
  - Start simple, build up a model yourself
  - Choose reasonable priors (discussed next) but don't worry too much about "final choices" until a later stage of your project
  - Worry about efficiency aspects later in the process, if/when you run into issues or when you get into a stage where you want to do lots of model fitting
- But also, learn from examples (and don't re-invent the wheel for actual analyses):
  - Use Stan examples from the user guide
  - `stan_demo` has 500+ examples from various references
  - Consider published code with papers
  - Check the model and stan inputs from `brm`-call (as starting point)
- And check the Stan reference manual and lots of other info on `mc-stan.org`

Example: brm-based Stan model

- (Part of) model for estimating μ and σ when $y_i|\mu, \sigma \sim N(\mu, \sigma^2)$

```
fit_brm <- brm(y ~ 1, data = dat)
stancode(fit_brm)
-----------------------
data {
  int<lower=1> N;  // total number of observations
  vector[N] Y;  // response variable
}
parameters {
  real Intercept;  // temporary intercept for centered predic
  real<lower=0> sigma;  // dispersion parameter
}
```

Example: brm-based Stan model

- (Part of) model for estimating μ and σ when $y_i|\mu, \sigma \sim N(\mu, \sigma^2)$

```
transformed parameters {
  real lprior = 0;  // prior contributions to the log posteri
  lprior += student_t_lpdf(Intercept | 3, 1.3, 2.5);
  lprior += student_t_lpdf(sigma | 3, 0, 2.5)
        - 1 * student_t_lccdf(0 | 3, 0, 2.5);
}
model {
  // likelihood including constants
  // initialize linear predictor term
  vector[N] mu = Intercept + rep_vector(0.0, N);
  target += normal_lpdf(Y | mu, sigma);
  // priors including constants
  target += lprior;
}
```

What's going on in `brm`-based Stan model?

- Target specifies the (addition to) log-posterior
  $\Rightarrow$ Alternative way of specifying the likelihood and prior densities
- For likelihood:
  - `target += normal_lpdf(Y | mu, sigma)`
  - is equivalent to `y` $\sim$ `normal(mu, sigma)`.
- For priors (choice of priors to be discussed next): combi of
  - `lprior += student_t_lpdf(Intercept | 3, 1.3, 2.5)` in `transformed parameters` and
  - `target += lprior` in `model`

  is equivalent to
  - `Intercept` $\sim$ `student_t(3, 1.3, 2.5)` in `model`;
- Some code added for additional functionality/generalizability:
  - "temporary intercept for centered predictors" makes (more) sense when we add covariates
  - data `prior_only` equals FALSE in our default fit (when TRUE, there is no fitting, just simulating from the prior; discussed later)

How to set prior distributions: big picture summary

- To start with the obvious: If you have actual prior information on parameters, include that information in the prior.
- For many settings, with sufficient data, the posterior is not sensitive to the choice of prior, i.e., various priors produce the same posterior, as long as the prior choosen does not conflict with the data
- But for some settings, the posterior can be sensitive to choice of prior, so more effort needed
  - Worst case: the prior introduces bias
  - Other options: computational issues with poorly chosen prior
- Good to keep in mind: A prior that reflects weak knowledge about the outcomes of interest may still have desirable statistical properties in terms of bias-variance trade off
  - you may introduce some bias by incoorporating some prior information
  - but if that prior information is reasonable within the context of the problem,
  - you can still end up with a Bayesian estimator with preferable properties due to bias-variance trade off (illustration in HW2).

How to set prior distributions: big picture summary (ctd)

- Default approaches that work for range of settings is an active area of research. In this course, we will (try to) follow BDA/Stan team prior choice recommendations.
  - Main wiki: `https://github.com/stan-dev/stan/wiki/Prior-Choice-Recommendations`
  - Unfortunately, currently the wiki is not the easiest place to get information and BDA may be outdated. We will consider `brm` defaults for regression models and the various papers cited on the wiki.
- Some terminology to discuss:
  - Improper priors
  - Conjugate and semi-conjugate priors
  - Non-informative priors et al
  - Jeffreys priors
  - Unit information priors
  - Weakly informative priors

Improper priors

- A probability density $p(\theta)$ is called improper if it does not integrate to one, $\int p(\theta)d\theta \neq 1$.
- Example: flat density, $p(\theta) \propto 1$.
- When using an improper prior, the posterior may or may not be proper.
- Not usually recommended across the board for all model parameters in a complex model (explained later).

Conjugate priors

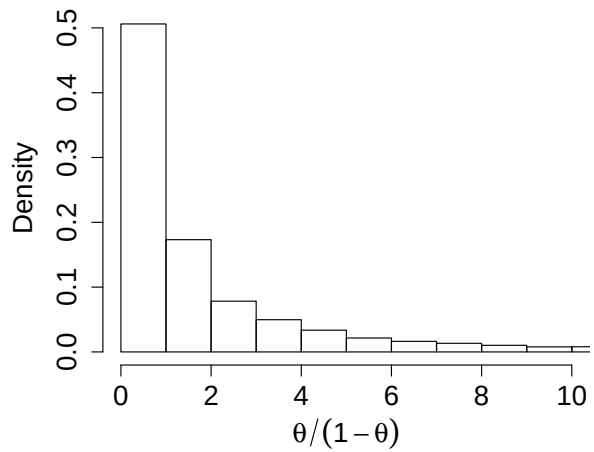
- Conjugate prior: For a certain likelihood, a prior distribution $p(\theta)$ for unknown parameter θ is called a *conjugate* prior distribution if it results in a posterior distribution $p(\theta|\boldsymbol{y})$ of the same form.
- Examples:
  - Normal prior when estimating population mean μ when $y_i|\mu \sim N(\mu, s^2)$, where s^2 is known:
    If $\mu \sim N(m_0, s^2_{\mu 0})$, then $\mu|\boldsymbol{y} \sim N(..., ...)$
  - (not discussed) Estimating probability θ when $y \sim Bin(n, \theta)$:
    Combining a Beta(a, b) prior on θ with a binomial likelihood results in the posterior $\theta|y \sim Beta(y + a, n - y + b)$.

Semi-conjugate priors

- Semi-conjugate prior: For a certain likelihood, a prior distribution $p(\theta)$ for unknown parameter θ is called a *semi-conjugate* prior distribution if it results in a full conditional distribution $p(\theta|\boldsymbol{y}, \text{other model parameters})$ of the same form.
- Example: estimating mean μ when $y_i|\mu, \sigma^2 \sim N(\mu, \sigma^2)$:
  - If $\mu \sim N(m_0, s_{\mu 0}^2)$, then $\mu|\boldsymbol{y}, \sigma^2 \sim N(..., ...)$.
  - (not discussed) If $1/\sigma^2 \sim Gamma(\nu_0/2, \nu_0/2 \cdot \sigma_0^2)$, then $1/\sigma^2|\boldsymbol{y}, \mu \sim Gamma(..., ...)$.
- For earlier MCMC algorithms that were based on Gibbs-sampling, semi-conjugate priors were often a default option for computational efficiency.
- This computational gain is not relevant for Stan's default sampling algorithm.

Noninformative prior distributions

- Noninformative prior is (ideally) a prior that does not represent any prior information but instead, "lets the data speak".
- Problem: how can we set a noninformative prior?
- Example: for $y|\theta \sim Bern(\theta)$, is $\theta \sim U(0,1)$ a noninformative prior?
- Maybe, when you consider inference for θ because the prior is flat...
- but not if you look at some transformed version of θ, e.g. odds $\psi = \theta/(1-\theta)$, the prior on θ imposes a non-flat prior on ψ.
- So which scale to choose for the flat prior?



- Conclusion: it is not necessarily clear what the "best noninformative" prior is.
- Attempts to find a way to specify a prior that does not depend on the scale of the parameters include Jeffreys priors.

Jeffreys priors (suppl. material)

- Approach: for a given likelihood $p(\theta|\boldsymbol{y})$, use prior distribution $p(\theta)$ that is proportional to the square root of the Fisher information $I(\theta)$,

$$p(\theta) \propto \sqrt{I(\theta)},$$

  with Fisher information $I(\theta) = -E(\partial^2 \log(p(\boldsymbol{y}|\theta))/\partial\theta^2)$.
- Hence, the prior depends on the likelihood function.
- Result for a given likelihood $p(\boldsymbol{y}|\theta)$:
  - Suppose we construct a Jeffreys prior $p(\theta)$
  - Suppose we also construct a Jeffreys prior $p(\psi)$ for a transformation $\psi = h(\theta)$ with $\theta = h^{-1}(\psi)$.
  - The prior induced on ψ through $p(\theta)$ equals the Jeffreys prior on ψ:

$$p^*(\psi) = p(\theta) \cdot |\partial h^{-1}(\psi)/\partial\psi| = p(\psi).$$

- Example when $y \sim Bern(\theta)$: Beta(0.5,0.5) is Jeffrey's prior for θ.
- Some problems:
  - J's prior may result in posteriors that do not make sense substantively
  - Jeffreys priors may be improper.

Noninformative prior distributions (ctd)

- Priors always provides some information, hence terms like vague/diffuse priors are more appropriate.
- For many settings, with sufficient data, the posterior is not sensitive to the choice of prior, so vague priors work fine.
  - We can compare prior and posterior to check that prior is roughly "locally flat" for regions with high likelihood
  - We can change priors to check sensitivity of model results to prior specification.
- But for some settings, the posterior can be sensitive to choice of prior, then consider using a more context-based prior.

Unit information priors (suppl. material)

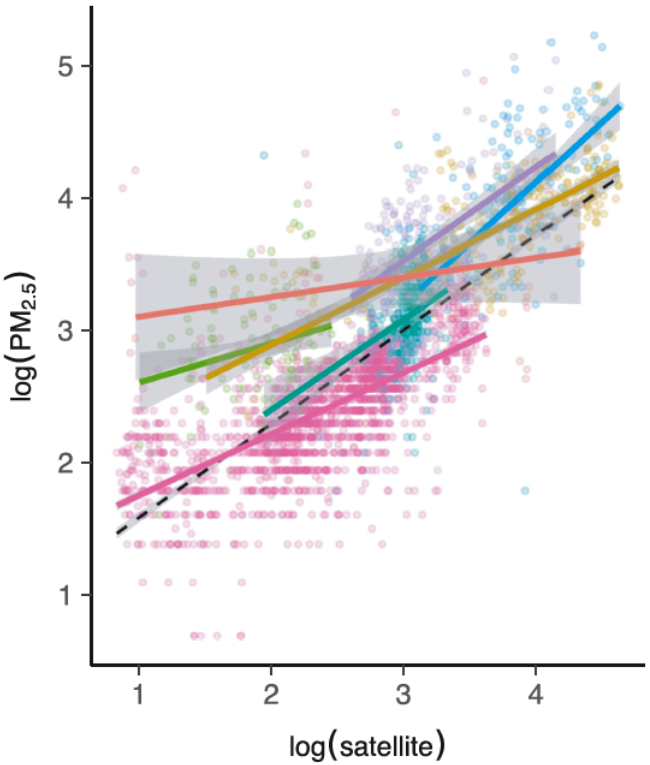
- A standardized approach to setting a prior is using a unit information prior (UIP, Kass and Wasserman 1995).
- Goal: set a prior that is informed by the data and carries as much information as one observation.
- Approach (Hoff p.231 or LL p.276, extra):
  - Set $\theta \sim N(\hat{\theta}, nI(\hat{\theta})^{-1})$, with $\hat{\theta}$ the MLE and $I(\hat{\theta})$ the observed Fisher information $I(\hat{\theta}) = -(\partial^2 \log(p(\boldsymbol{y}|\theta))/\partial\theta^2)|_{\theta=\hat{\theta}}$.
- Example: $y_i|\mu, \sigma \sim N(\mu, \sigma^2)$ with σ^2 known:
  If $\mu|\sigma \sim N(\mu_0, \sigma^2_{\mu 0})$, with $\mu_0 = \bar{y}$ and $\sigma^2_{\mu 0} = \sigma^2$, then
  $\mu|\boldsymbol{y} \sim N\left(\frac{\mu_0/\sigma^2_{\mu 0} + n\cdot\bar{y}/\sigma^2}{1/\sigma^2_{\mu 0} + n/\sigma^2}, \frac{1}{1/\sigma^2_{\mu 0} + n/\sigma^2}\right) \Rightarrow \mu|\boldsymbol{y} \sim N(\bar{y}, \sigma^2/(n+1))$.
- UIP is not really a prior because it is informed by the data but can be considered as the prior info of someone with weak but accurate prior information.

Weakly informative priors

- A weakly informative prior contains some information – enough to regularize the posterior distribution, that is, to keep it roughly within reasonable bounds–but without attempting to fully capture one's scientific knowledge about the underlying parameter (BDA 2.8/2.9).
- Main idea:
  - quite often there's at least some knowledge about the scale
  - useful also if there's more information from previous observations, but not certain how well that information is applicable in a new case uncertainty
- Construction:
  - Start with some version of a noninformative prior distribution and then add enough information so that inferences are constrained to be reasonable, OR
  - Start with a strong, highly informative prior and broaden it to account for uncertainty in one's prior beliefs and in the applicability of any historically based prior distribution to new data.

Example of weakly informative prior

Taken from Gabry et al (2019).Visualization in Bayesian workflow (discussed next module)

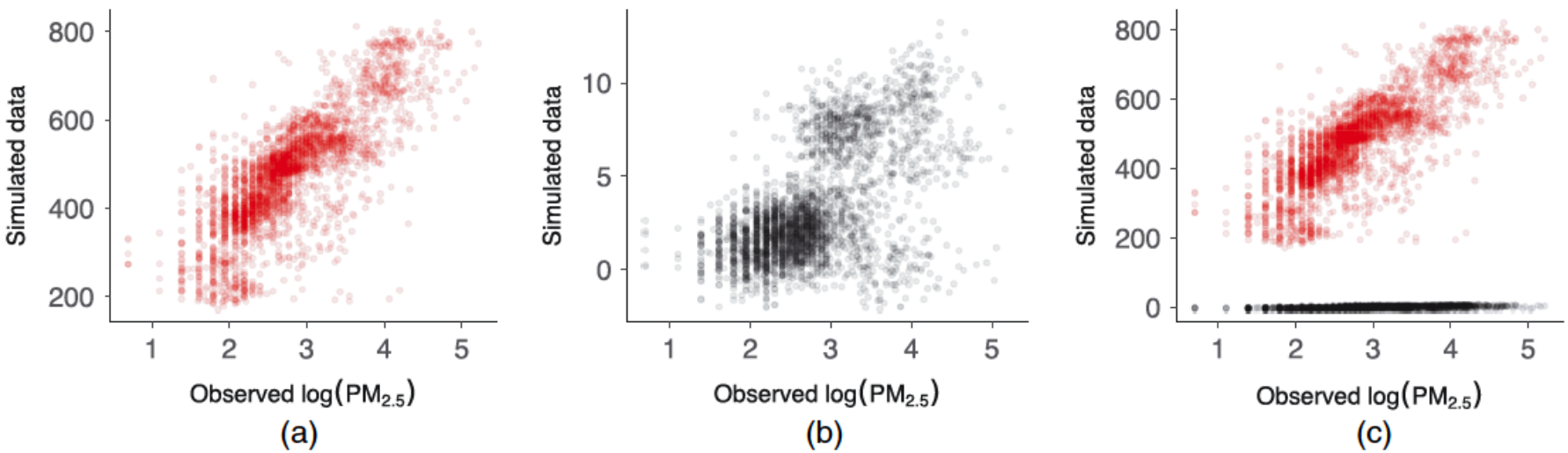


- A recent report estimated that $PM_{2.5}$ (particulate matter measuring less than 2.5 microns in diameter) is responsible for three million deaths worldwide each year (Shaddick et al, 2017)
- Goal: Estimation of $PM_{2.5}$, using satellite data, in different regions of the world

Model for log($PM_{2.5}$): $y_i|\mu_i, \sigma \sim N(\mu_i, \sigma_y^2)$, with $\mu_i = \alpha_{j[i]} + \beta_{j[i]} x_i$, where x_i is log(satelite estimate), and region-specific coefficients α_j, β_j are estimated hierarchically

Example of weakly informative prior (ctd)

- Plot below show simulated data (on y-axis) against real data (on x-axis) for two sets of priors:
  which prior setting (a or b) do you think is referred to as weakly informative, as opposed to vague?
- Weakly informative priors are used in setting (b): creating simulated data that is more extreme than the observed data but excludes implausible results



Practical summary advice on prior choice

- For many settings, with sufficient data, the posterior is not sensitive to the choice of prior, as long as that prior does not conflict with the data.
  - Check that priors are vague relative to posteriors, do sensitivity checks
  - Note that results may be much more sensitive to other model assumptions, e.g. choice of likelihood function, regression function, ... (next module)
- But for some settings, the posterior can be sensitive to choice of prior. In those cases, consider
  - prior choice recommendations from the literature/stan team
  - prior predictive checks (discussed in a later module)
  - a different model set-up such that you don't have to specify a prior, e.g. through a hierarchical model.