Hierarchical Models with the rstanarm and brms Packages

Ben Goodrich

May 04, 2020

Final Projects

- Due by 11:59 PM on May 19th
- Can analyze data used in another class
- If you cannot share the data, let me know
- Can use rstanarm or brms or write your own Stan code (next week)
- I don't care very much what the previous literature says
- Go through the process of laying out a generative model, drawing from the prior predictive distribution, conditioning on the observed data (and making sure Stan samples well), looking at posterior predictive plots, comparing it to an alternative model, etc.
- Should be around ten pages as a PDF

`GLD_solver_LBFGS` in GLD_helpers.R

- Week2/GLD_helpers.R now has a `GLD_solver_LBFGS` function

```
source(file.path("..", "Week2", "GLD_helpers.R"))
args(GLD_solver_LBFGS)
```

```
## function (lower_quartile, median, upper_quartile, other_quantile,
##     alpha, check = TRUE)
## NULL
```

are the same as `GLD_solver`

- The `GLD_solver_LBFGS` function throws warnings rather than errors when it cannot find an exact solution for the asymmetry and steepness parameters

```
rstan::expose_stan_functions(file.path("..", "Week2", "quantile_functions.stan"))
a_s <- GLD_solver_LBFGS(lower_quartile = 3, median = 5, upper_quartile = 8,
                        other_quantile = 1200, alpha = 0.9) # warnings
```

```
## final  value 0.595725
## stopped after 4 iterations
```

What Are Hierarchical Models

- In Bayesian terms, a hierarchical model is nothing more than a model where the prior distribution of some parameter depends on another parameter
- In other words, it is just another application of the Multiplication Rule

$$f(\boldsymbol{\theta}) = \int f(\boldsymbol{\theta} \mid \boldsymbol{\phi}) f(\boldsymbol{\phi}) \, d\phi_1 \dots d\phi_K$$

- But most of the discussion of “hierarchical models” refers to the very narrow circumstances in which they can be estimated via frequentist methods
- From a frequentist perspective, a hierarchical model is appropriate for cluster random sampling designs, inappropriate for stratified random sample designs, and hard to justify for other sampling designs

Breakout Rooms

Suppose the coefficient on age is a linear function of the income of the person's zip code in a logit model for whether they vote, in the Oregon Medicaid experiment dataset from the homework. Draw from the prior predictive distribution in R using `replicate`. You can ignore the other predictors.

Cluster Sampling vs. Stratified Sampling

- For cluster random sampling, you
  - Sample J large units (such as schools) from their population
  - Sample N_j small units (such as students) from the j-th large unit
- If you replicate such a study, you get different realizations of the large units
- For stratified random sampling, you
  - Divide the population of large units into J mutually exclusive and exhaustive groups (like states)
  - Sample N_j small units (such as voters) from the j-th large unit
- If you replicate such a study, you would use the same large units and only get different realizations of the small units

Why Bayesians Should Use Hierarchical Models

- Suppose you estimated a Bayesian model on people in New York
- Next, you are going to collect data on people who live in Connecticut
- Intuitively, the New York posterior should influence the Connecticut prior
- But it is unlikely that the data-generating processes in Connecticut is exactly the same as in New York
- Hierarchical models apply when you have data from New York, Connecticut, and other states at the same time
- Posterior distribution in any one state is not independent of other states
- Posterior distribution in any one state are not the same as in other states
- McElreath argues hierarchical models should be the default and “flat” models should be the rare exception only when justified by the data
- With more data, there is always more heterogeneity in the data-generating processes that a generative model should be allowing for

Models with Group-Specific Intercepts

- Let α be the common intercept and $\boldsymbol{\beta}$ be the common coefficients while a_j is the deviation from the common intercept in the j-th group. Write a model as:

$$y_{ij} = \overbrace{\underbrace{\alpha + \sum_{k=1}^{K} \beta_k x_{ik}}_{\text{Frequentist } \boldsymbol{\mu}|\mathbf{x}} + a_j}^{\text{Bayesian } \boldsymbol{\mu}|\mathbf{x},j} + \boldsymbol{\epsilon} = \alpha + \sum_{k=1}^{K} \beta_k x_{ik} + \underbrace{a_j + \overbrace{\boldsymbol{\epsilon}}^{\text{Bayesian error}}}_{\text{Frequentist error}}$$

- The same holds in GLMs where $\eta_{ij} = \alpha + \sum_{k=1}^{K} \beta_k x_{ik} + a_j$ or $\eta_{ij} = \alpha + \sum_{k=1}^{K} \beta_k x_{ik}$ depending on whether you are Bayesian or Frequentist

Models with Group-Specific Slopes and Intercepts

- Let α be the common intercept and $\boldsymbol{\beta}$ be the common coefficients while a_j is the deviation from the common intercept in the j-th group and $\mathbf{b}_j$ is the deviation from the common coefficients. Write the model as:

$$y_{ij} = \overbrace{\underbrace{\alpha + \sum_{k=1}^{K} \beta_k x_{ik}}_{\text{Frequentist } \mu|\mathbf{x}} + a_j + \sum_{k=1}^{K} b_{jk} x_{ik}}^{\text{Bayesian } \mu|\mathbf{x},j} + \boldsymbol{\epsilon} =$$

$$\alpha + \sum_{k=1}^{K} \beta_k x_{ik} + \underbrace{a_j + \sum_{k=1}^{K} b_{jk} x_{ik} \qquad \overbrace{\boldsymbol{\epsilon}}^{\text{Bayesian error}}}_{\text{Frequentist error}}$$

- And similarly for GLMs

Frequentist Estimation of Multilevel Models

- Frequentists assume that a_j and b_j deviate from the common parameters according to a (multivariate) normal distribution, whose (co)variances are common parameters to be estimated
- To Frequentists, a_j and b_j are not parameters because parameters must remained fixed in repeated sampling of observations from some population
- Since a_j and b_j are not parameters, they can't be "estimated" only "predicted"
- Since a_j and b_j aren't estimated, they must be integrated out of the likelihood function, leaving an integrated likelihood function of the common parameters
- After obtaining maximum likelihood estimates of the common parameters, each a_j and b_j can be predicted from the residuals via a regression
- Estimated standard errors produced by frequentist software are too small
- There are no standard errors etc. for the a_j and b_j
- Maximum likelihood estimation often results in a corner solution

Table 2 from the lme4 Vignette (see also the FAQ)

Formula	Alternative	Meaning
`(1 | g)`	`1 + (1 | g)`	Random intercept with fixed mean
`0 + offset(o) + (1 | g)`	`-1 + offset(o) + (1 | g)`	Random intercept with *a priori* means
`(1 | g1/g2)`	`(1 | g1)+(1 | g1:g2)`	Intercept varying among `g1` and `g2` within `g1`
`(1 | g1)+(1 | g2)`	`1 + (1 | g1) + (1 | g2)`	Intercept varying among `g1` and `g2`
`x + (x | g)`	`1 + x + (1 + x | g)`	Correlated random intercept and slope
`x + (x || g)`	`1 + x + (1 | g) + (0 + x | g)`	Uncorrelated random intercept and slope

Table 2: Examples of the right-hand sides of mixed-effects model formulas. The names of grouping factors are denoted `g`, `g1`, and `g2`, and covariates and *a priori* known offsets as `x` and `o`.

lme4 syntax

Hierarchical Models in Psychology

- In political science and economics, the “big” units are often countries or sub-national political areas like states and the “small” units are people
- In psychology, the “big” units are often people and the “small” units are questions or outcomes on repeated tasks
- Hierarchical model syntax is like

```
y ~ x + (x | person) + (1 | question)
```

- Question of interest is how to predict `y` for a new “big” unit (person)

Hierarchical Models in rstanarm (from this paper)

```
dat <- read.csv("https://osf.io/5cg32/download")
library(rstanarm)
```

```
post <-  stan_glmer(valence ~ arousal + (1 + arousal | PID), data = dat,
                    prior = normal(0, 1, autoscale = FALSE),
                    prior_intercept = normal(50, 100, autoscale = FALSE))
```

```
post
```

```
...
##             Median MAD_SD
## (Intercept) 29.9    5.4
## arousal      0.5    0.1
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 9.3    0.4
```

```
##
## Error terms:
##  Groups   Name        Std.Dev. Corr
##  PID      (Intercept) 20.71
##           arousal      0.24    -0.66
##  Residual              9.27
## Num. levels: PID 20
##
...
```

Accessor Functions (based on the lme4 package)

```r
fixef(post)
```

```
## (Intercept)     arousal
##  29.9281178   0.5380694
```

```r
cbind(b = head(ranef(post)$PID), total = head(coef(post)$PID))
```

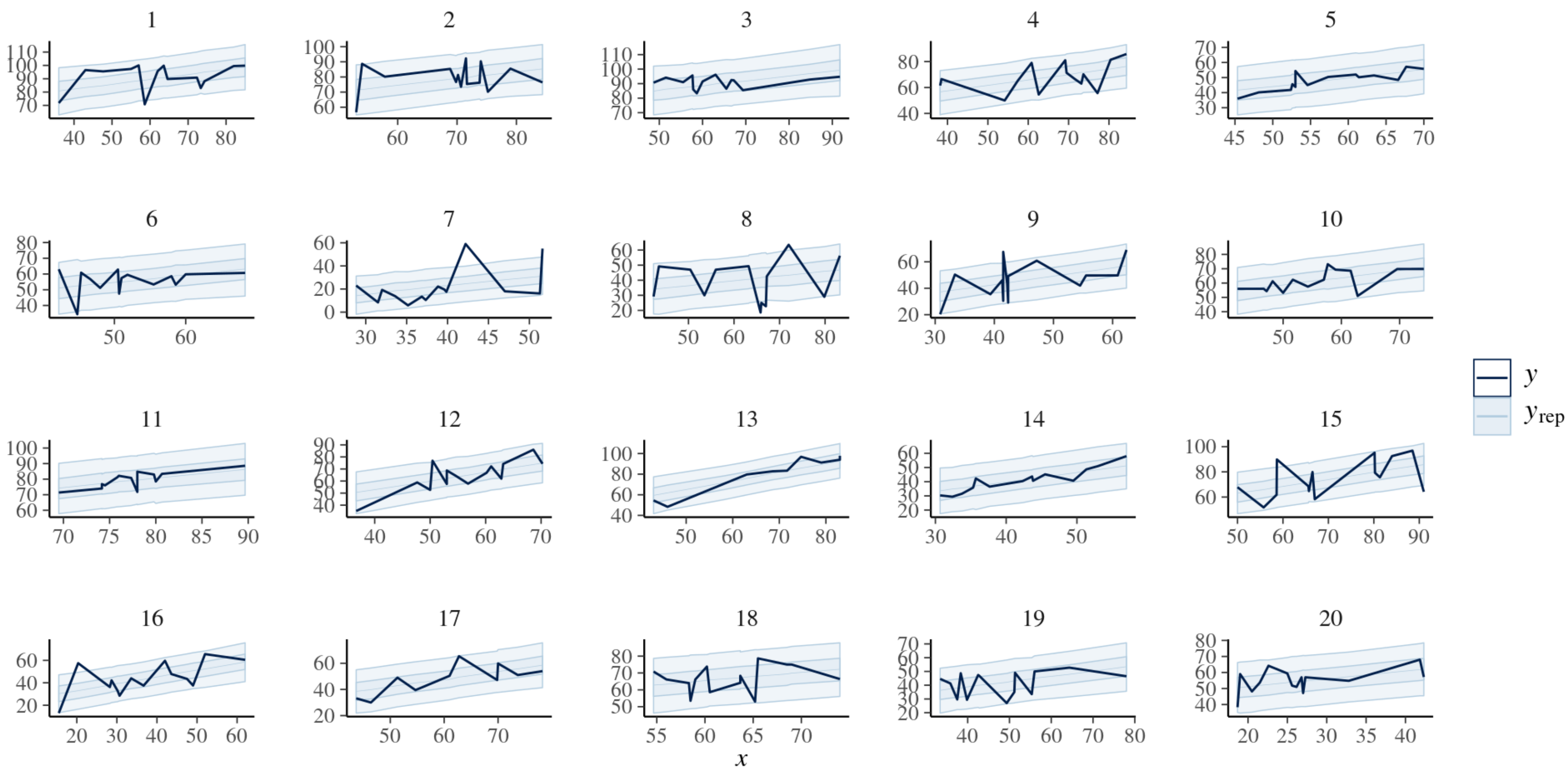
```
##   b.(Intercept)   b.arousal total.(Intercept) total.arousal
## 1     36.987540 -0.16446363          66.91566     0.3736058
## 2     17.187123 -0.08308324          47.11524     0.4549861
## 3     38.584861 -0.20028856          68.51298     0.3377808
## 4      9.849710 -0.10118385          39.77783     0.4368855
## 5    -13.972518  0.02890319          15.95560     0.5669726
## 6      1.199908 -0.06238785          31.12803     0.4756815
```

```r
dim(as.matrix(post)) # 4000 x 46
```

```
## [1] 4000   46
```

Posterior Predictive Checks

```
pp_check(post, plotfun = "ribbon_grouped", x = dat$arousal, group = dat$PID)
```



Frequentist Perspective

- Frequentists also assume $\mathbf{u}_j \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$, implying $\mathbf{b}_j \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{L}\mathbf{L}^\top\right)$
- But frequentists integrate each $\mathbf{u}_j$ out of the original likelihood function and then choose $\widehat{\boldsymbol{\beta}}, \widehat{\sigma}$, and $\widehat{\mathbf{L}}$ to maximize the integrated likelihood, which (pre-data) in the Gaussian case is N-dimensional multinormal with expectation $\mathbf{X}\boldsymbol{\beta}$ and variance-covariance matrix $\sigma^2 \left(\mathbf{Z}^\top \left(\mathbf{I} \otimes \mathbf{L}\mathbf{L}^\top\right) \mathbf{Z} + \mathbf{I}\right)$
- If the original likelihood is not Gaussian, then the integrated likelihood does not have an analytical form but it could be maximized numerically
- Frequentists refer to $\widehat{\boldsymbol{\beta}}, \widehat{\sigma}$, and $\widehat{\mathbf{L}}$ as (estimated) fixed effects because β, $\boldsymbol{\sigma}$, and $\mathbf{L}$ are parameters that are fixed across repeated samples of datasets from the population, whereas the $\mathbf{b}_j$ are referred to as random effects, which are not parameters because they vary from dataset to dataset

Frequentist Example

```r
poll <- readRDS("GooglePoll.rds") # WantToWin is coded as 1 for Romney and 0 for Obama
poll$Income[poll$Income == "150,000+"] <- "100,000-149,999"
library(dplyr)
collapsed <- filter(poll, !is.na(WantToWin)) %>%
             group_by(Region, Gender, Urban_Density, Age, Income) %>%
             summarize(Romney = sum(grepl("Romney", WantToWin)), Obama = n() - Romney) %>%
             na.omit
```

```r
lme4::glmer(cbind(Romney, Obama) ~ Gender + Urban_Density + Age + Income +
              (Gender + Urban_Density + Age + Income | Region),
            data = collapsed, family = binomial(link = "logit"))
```

```
## singular fit
```

- For models that are more complicated than `(1 + x | g)`, the MLE of $\boldsymbol{\Sigma}$ usually implies that $\widehat{\boldsymbol{\Sigma}}^{-1}$ does not exist

Bayesian Version of the “Same” Model

```r
post_h <- stan_glmer(cbind(Romney, Obama) ~ Gender + Urban_Density + Age + Income +
                       (Gender + Urban_Density + Age + Income | Region),
                     data = collapsed, family = binomial(link = "logit"),
                     QR = TRUE, adapt_delta = 0.98, seed = 12345)
```

```r
post_h
```

```
...
##  observations: 513
## ------
##                        Median MAD_SD
## (Intercept)            -0.5    0.2
## GenderMale              0.4    0.1
## Urban_DensitySuburban  -0.2    0.1
## Urban_DensityUrban     -0.5    0.1
## Age25-34                0.1    0.1
## Age35-44                0.5    0.1
## Age45-54                0.8    0.1
## Age55-64                0.8    0.1
## Age65+                  1.3    0.1
## Income25,000-49,999    -0.1    0.1
## Income50,000-74,999    -0.1    0.1
## Income75,000-99,999    -0.1    0.2
## Income100,000-149,999   0.2    0.3
##
## Error terms:
##  Groups Name                  Std.Dev. Corr
##  Region (Intercept)           0.169
##         GenderMale            0.093     0.00
##         Urban_DensitySuburban 0.096    -0.02  0.00
##         Urban_DensityUrban    0.098     0.02  0.00  0.07
##         Age25-34              0.107     0.07 -0.01  0.00  0.02
##         Age35-44              0.122     0.00  0.02  0.02
##         Age45-54              0.113     0.09 -0.02  0.00
##         Age55-64              0.100     0.02  0.00  0.01
##         Age65+                0.108     0.03 -0.01 -0.01
##         Income25,000-49,999   0.115    -0.08 -0.02 -0.03
##         Income50,000-74,999   0.115    -0.03 -0.01  0.01
##         Income75,000-99,999   0.129    -0.03  0.03 -0.01
##         Income100,000-149,999 0.130     0.03 -0.01  0.01
##
##
##
##
##
##
##
##
##
##
##   0.03
##   0.00  0.02
##  -0.03  0.00 -0.01
##   0.01 -0.01  0.00 -0.02
## Num. levels: Region 4
##
## Sample avg. posterior predictive distribution of y:
##          Median MAD_SD
## mean_PPD 5.5    0.1
##
## ------
```

Poststratification

```
mu <- posterior_linpred(post_h, transform = TRUE)
dim(mu)
```

```
## [1] 4000  513
```

- Assume `shares` is the proportion of voters for each level of `Gender`, `Urban_Density`, `Age`, and `Income` crossed with `Region`

```
mu_ <- mu %*% shares
```

- Now you have a posterior distribution for the proportion supporting Romney for the country as a whole

PSISLOOCV (of a group)

```
(loo_hier <- loo(post_h)) # 156 parameters
```

```
...
##          Estimate   SE
## elpd_loo   -845.9 20.5
## p_loo        27.6  2.3
## looic      1691.8 41.1
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
NA
NA
NA
NA
...
```

What Were the Priors?

```
prior_summary(post_h)
```

```
## Priors for model 'post_h'
## ------
## Intercept (after predictors centered)
##  ~ normal(location = 0, scale = 10)
##
## Coefficients (in Q-space)
##  ~ normal(location = [0,0,0,...], scale = [2.5,2.5,2.5,...])
##
## Covariance
##  ~ decov(reg. = 1, conc. = 1, shape = 1, scale = 1)
## ------
## See help('prior_summary.stanreg') for more details
```

What Is `decov(1, 1, 1, 1)`?

- `decov` = Decomposition of Covariance
- `reg.` is the regularization parameter in the LKJ prior on the correlation matrix
- `conc.` is the concentration parameter in the Dirichlet prior on the variance components
- `shape` and `scale` pertain to the Gamma prior on multiplier for the variance components
- You usually do not need to change these defaults to get good results

Cafes Example from McElreath

14.1.3. The varying slopes model. Now we're ready to play the process in reverse. We just generated data from a set of 20 cafés, and those cafés were themselves generated from a statistical population of cafés. Now we'll use that data to learn about the data-generating process, through a model.

The model is much like the varying intercepts models from the previous chapter. But now the joint population of intercepts and slopes appears, instead of just a distribution of varying intercepts. This is the varying slopes model, with explanation to follow. First we have the probability of the data and the linear model:

$$W_i \sim \text{Normal}(\mu_i, \sigma) \qquad \text{[likelihood]}$$

$$\mu_i = \alpha_{\text{CAFÉ}[i]} + \beta_{\text{CAFÉ}[i]} A_i \qquad \text{[linear model]}$$

Then comes the matrix of varying intercepts and slopes, with it's covariance matrix:

$$\begin{bmatrix} \alpha_{\text{CAFÉ}} \\ \beta_{\text{CAFÉ}} \end{bmatrix} \sim \text{MVNormal}\left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \mathbf{S} \right) \qquad \text{[population of varying effects]}$$

$$\mathbf{S} = \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \mathbf{R} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \qquad \text{[construct covariance matrix]}$$

McElreath (2020) chapter 14

Cafes Example with brms

```r
brm(wait ~ 1 + afternoon + (1 + afternoon | cafe), data = dataset,
    prior = c(prior(normal(0, 10), class = Intercept),
              prior(normal(0, 10), class = b),
              prior(cauchy(0, 2), class = sd),    # absolute value of Cauchy
              prior(cauchy(0, 2), class = sigma), # same
              prior(lkj(2), class = cor))         # just a correlation
```

Breakout Rooms: Prior Predictive Distribution

Write a Stan function that returns a matrix whose rows are drawn from the prior predictive distribution of the previous model. You will need `normal_rng`, `cauchy_rng`, `uniform_rng`, (for the correlation), and `bernoulli_logit_rng`.

```
functions {
  matrix prior_PD_rng(int S, vector A, int[] cafe) {
    int N = rows(A);
    int J = max(cafe);
    int N_j = N / J; # assume all cafes have the same N_j
    for (s in 1:S) {
      // draw population parameters here
      for (j in 1:J) {
        // draw cafe-specific parameters here
        // draw outcomes here
      }
    }
  }
}
```

McElreath / Kotz Example

```
library(brms)
funding <-
  tibble(
    discipline   = rep(c("Chemical sciences", "Physical sciences", "Physics", "Humanities",
                         "Technical sciences",  "Interdisciplinary", "Earth/life sciences",
                         "Social sciences", "Medical sciences"),
                     each = 2),
    gender       = rep(c("m", "f"), times = 9),
    applications = c(83, 39, 135, 39, 67, 9, 230, 166, 189, 62, 105, 78, 156, 126, 425, 409, 2
    awards       = c(22, 10, 26, 9, 18, 2, 33, 32, 30, 13, 12, 17, 38, 18, 65, 47, 46, 29) %>%
    rejects      = c(61, 29, 109, 30, 49, 7, 197, 134, 159, 49, 93, 61, 118, 108, 360, 362, 19
    male         = ifelse(gender == "f", 0, 1) %>% as.integer()
  )
```

```
b13.bonus_2 <-
  brm(awards | trials(applications) ~ 1 + male + (1 + male | discipline),
      data = funding, family = binomial, control = list(adapt_delta = 0.9),
      prior = c(prior(normal(0, 4), class = Intercept), prior(normal(0, 4), class = b),
                prior(cauchy(0, 1), class = sd), prior(lkj(4), class = cor)))
```

```
## Compiling the C++ model

## Start sampling
```

Results

```
b13.bonus_2
```

```
##  Family: binomial
##   Links: mu = logit
## Formula: awards | trials(applications) ~ 1 + male + (1 + male | discipline)
##    Data: funding (Number of observations: 18)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~discipline (Number of levels: 9)
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)            0.27      0.14     0.03     0.60 1.00      895      555
## sd(male)                 0.32      0.18     0.03     0.74 1.00     1014     1249
## cor(Intercept,male)     -0.17      0.31    -0.70     0.45 1.00     2957     3083
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -1.62      0.15    -1.88    -1.31 1.00     1887     2062
## male          0.15      0.17    -0.21     0.48 1.00     2063     1987
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Meta-Analysis

- “Meta-analysis” of previous studies is popular in some fields such as education and medicine
- Can be written as a multi-level model where each study is its own “group” with its own intercept that captures the difference between what each study is estimating and what it wants to estimate
- Outcome is the point estimate for each Frequentist study
- Estimated standard error from each Frequentist study is treated as an exogenous known

Meta-Analysis Example

```r
data("towels", package = "metaBMA")
prior_c <- prior(normal(0, 1), class = "Intercept") + prior(cauchy(0, 0.3), class = "sd")
towels_c <- brm(logOR | se(SE) ~ 1 + (1 | study), data = towels,
                prior = prior_c, control = list(adapt_delta = 0.97))
```

```
## Compiling the C++ model
```

```
## Start sampling
```

Meta-Analysis Results

```
towels_c
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: logOR | se(SE) ~ 1 + (1 | study)
##    Data: towels (Number of observations: 7)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~study (Number of levels: 7)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.12      0.11     0.01     0.41 1.00     1464     2343
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.21      0.11    -0.02     0.40 1.00     2160     1634
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Gaussian Processes

A simple Gaussian Process logit model with a squared exponential covariance function is

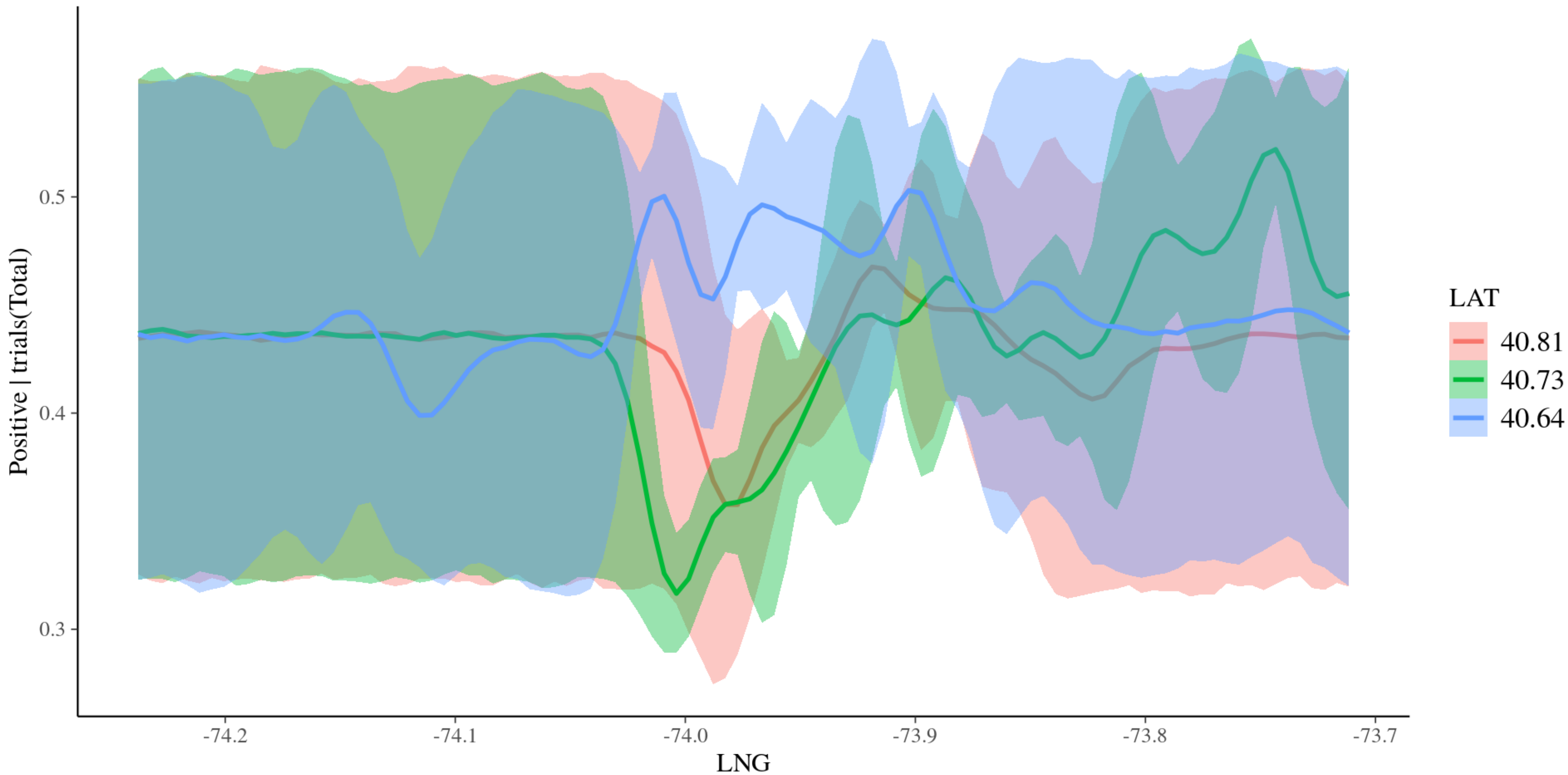
- $\frac{1}{\rho} \sim \mathcal{G}amma\left(a,b\right)$
- $\alpha \sim t_+\left(v,0,s\right)$
- $\Sigma_{ij} = \Sigma_{ji} = \alpha^2 e^{-\frac{1}{\rho}\sum_{d=1}^{D}\left(x_{id}-x_{jd}\right)^2}$
- $\gamma \sim \mathcal{N}\left(0,2\right)$
- $\boldsymbol{\eta} \sim \mathcal{N}\left(\gamma\mathbf{1},\boldsymbol{\Sigma}\right)$
- $\mu_j = \frac{1}{1+e^{-\eta_j}}$
- $y_j \sim \text{Binomial}\left(n_j,\mu_j\right)$

where, for example, n_j is the number of coronavirus tests in zipcode j and y_j is the number of positives

```
tests <- readr::read_csv("https://raw.githubusercontent.com/nychealth/coronavirus-data/master/tests-by-zcta.csv")[-
zipcodes <- readr::read_csv("https://gist.githubusercontent.com/erichurst/7882666/raw/5bdc46db47d9515269ab12ed6fb28
zipcodes$ZIP <- as.integer(zipcodes$ZIP)
tests <- dplyr::inner_join(tests, zipcodes, by = c("MODZCTA" = "ZIP"))
post <- brm(Positive | trials(Total) ~ 1 + gp(LAT, LNG), data = tests, family = binomial)
```

What Did the Gaussian Process Model Imply?

```
conditional_effects(post, effects = "LNG:LAT")
```



How Good Was the Model?

```
bayes_R2(post)
```

```
##      Estimate    Est.Error      Q2.5     Q97.5
## R2 0.9987536 0.0001502648 0.9984368 0.9990337
```

```
loo(post)
```

```
## Warning: Found 147 observations with a pareto_k > 0.7 in model 'post'. With this many
## problematic observations, it may be more appropriate to use 'kfold' with argument 'K = 10'
## to perform 10-fold cross-validation rather than LOO.
```

```
##
## Computed from 4000 by 176 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo   -910.7 11.5
## p_loo       149.5  7.9
## looic      1821.4 23.1
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
```