Assignment 5

Louis Nix II

4/14/2020

1. American Family Survey

1.1 Adjusting for Party Identification or Related Variables

First, I do some simple recoding to all of the variable to ensure they are ordered where appropriate and lines with "Not Sure" or "Other" are removed.

```
AFS$app_dtrmp[AFS$app_dtrmp == "Not sure"] <- NA
AFS$app_dtrmp <- factor(AFS$app_dtrmp, ordered = TRUE,
                        levels = c("Strongly disapprove",
                                   "Somewhat disapprove",
                                   "Somewhat approve",
                                   "Strongly approve") )
AFS$pid3[AFS$pid3 == "Not sure" | AFS$pid3 =="Other"] <- NA
AFS$pid3 <- factor(AFS$pid3, ordered = TRUE,
                        levels = c("Democrat",
                                   "Independent",
                                   "Republican"))
AFS$pid7[AFS$pid7 == "Not sure"] <- NA
AFS$pid7 <- factor(AFS$pid7, ordered = TRUE,
                        levels = c("Strong Democrat",
                                   "Not very strong Democrat",
                                   "Lean Democrat",
                                   "Independent",
                                   "Lean Republican",
                                   "Not very strong Republican",
                                   "Strong Republican"))
AFS$ideo5[AFS$ideo5 == "Not sure"] <- NA
AFS$ideo5 <- factor(AFS$ideo5, ordered = TRUE,
                        levels = c("Very liberal",
                                   "Liberal",
                                   "Moderate",
                                   "Conservative",
                                   "Very conservative"))
AFS$presvote16post[AFS$presvote16post == "Other"] <- NA
AFS$presvote16post <- factor(AFS$presvote16post, ordered = FALSE,
                        levels = c("Did not vote for President",
                                   "Donald Trump",
                                   "Evan McMullin",
                                   "Gary Johnson",
```

```
                                    "Hillary Clinton",
                                    "Jill Stein"))
```

Now I run all five models (the base and then four others with each of the potential predictors added and removed iteratively).

```
base_prior <- prior(normal(3, 1), class = "Intercept") +
              prior(exponential(.75), class = "sd")

model_init <- brm(formula = app_dtrmp ~ 1 + (1 | inputstate),
                  data = AFS,
                  family = cumulative(),
                  prior = base_prior)
```

```
## Compiling the C++ model

## Trying to compile a simple C file

## Start sampling
```

```
model2 <- brm(formula = app_dtrmp ~ 1 + pid3 + (1 + pid3| inputstate),
                  data = AFS,
                  family = cumulative(),
                  prior = base_prior,
                  control = list(adapt_delta = 0.90))
```

```
## Compiling the C++ model

## Trying to compile a simple C file

## Start sampling
```

```
model3 <- brm(formula = app_dtrmp ~1 + pid7 + (1 + pid7| inputstate),
                  data = AFS,
                  family = cumulative(),
                  prior = base_prior)
```

```
## Compiling the C++ model

## Trying to compile a simple C file

## Start sampling
```

```
model4 <- brm(formula = app_dtrmp ~ 1 + ideo5 + (1 + ideo5| inputstate),
                  data = AFS,
                  family = cumulative(),
                  prior = base_prior)
```

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

```
## Trying to compile a simple C file

## Start sampling
```

```
model5 <- brm(formula = app_dtrmp ~ 1 + presvote16post + (1 + presvote16post| inputstate),
                  data = AFS,
                  family = cumulative(),
                  prior = base_prior)
```

```
## Compiling the C++ model

## Trying to compile a simple C file

## Start sampling
```

```
model_init_loo <- loo(model_init)
model_2_loo <- loo(model2)
model_3_loo <- loo(model3)
model_4_loo <- loo(model4)
model_5_loo <- loo(model5)
```

```
model_init_loo
```

```
##
## Computed from 4000 by 2782 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -3411.4 29.2
## p_loo        14.7  0.2
## looic      6822.8 58.3
## ------
## 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.
```

```
model_2_loo
```

```
##
## Computed from 4000 by 2523 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -2438.4 41.1
## p_loo        17.1  0.3
## looic      4876.7 82.2
## ------
## 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.
```

```
model_3_loo
```

```
##
## Computed from 4000 by 2702 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -2334.5 46.2
## p_loo        35.3  0.9
## looic      4669.1 92.3
## ------
## 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.
```

```
model_4_loo
```

```
##
## Computed from 4000 by 2585 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -2395.3 44.2
## p_loo        24.0  0.7
## looic      4790.5 88.5
## ------
## 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.
```

```
model_5_loo
```

```
##
## Computed from 4000 by 2725 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -2346.3 44.8
## p_loo        29.6  1.5
## looic      4692.6 89.6
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     2712  99.5%   1641
##  (0.5, 0.7]   (ok)         12   0.4%   1552
##    (0.7, 1]   (bad)         1   0.0%   2505
##    (1, Inf)   (very bad)    0   0.0%   <NA>
## See help('pareto-k-diagnostic') for details.
```

```
model5
```

```
##  Family: cumulative
##   Links: mu = logit; disc = identity
## Formula: app_dtrmp ~ 1 + presvote16post + (1 + presvote16post | inputstate)
##    Data: AFS (Number of observations: 2725)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~inputstate (Number of levels: 51)
##                                                                Estimate Est.Error
## sd(Intercept)                                                      0.11      0.08
## sd(presvote16postDonaldTrump)                                      0.10      0.08
## sd(presvote16postEvanMcMullin)                                     0.57      0.51
## sd(presvote16postGaryJohnson)                                      0.44      0.29
## sd(presvote16postHillaryClinton)                                   0.28      0.18
## sd(presvote16postJillStein)                                        0.58      0.51
## cor(Intercept,presvote16postDonaldTrump)                          -0.15      0.38
## cor(Intercept,presvote16postEvanMcMullin)                         -0.01      0.37
## cor(presvote16postDonaldTrump,presvote16postEvanMcMullin)         -0.00      0.38
## cor(Intercept,presvote16postGaryJohnson)                          -0.00      0.37
## cor(presvote16postDonaldTrump,presvote16postGaryJohnson)           0.03      0.38
## cor(presvote16postEvanMcMullin,presvote16postGaryJohnson)          0.00      0.37
## cor(Intercept,presvote16postHillaryClinton)                        0.05      0.37
## cor(presvote16postDonaldTrump,presvote16postHillaryClinton)       -0.05      0.37
## cor(presvote16postEvanMcMullin,presvote16postHillaryClinton)      -0.00      0.38
## cor(presvote16postGaryJohnson,presvote16postHillaryClinton)        0.03      0.37
## cor(Intercept,presvote16postJillStein)                             0.02      0.37
## cor(presvote16postDonaldTrump,presvote16postJillStein)            -0.04      0.38
## cor(presvote16postEvanMcMullin,presvote16postJillStein)           -0.01      0.37
## cor(presvote16postGaryJohnson,presvote16postJillStein)            -0.04      0.38
## cor(presvote16postHillaryClinton,presvote16postJillStein)          0.04      0.37
##                                                                l-95% CI u-95% CI
## sd(Intercept)                                                      0.01     0.28
## sd(presvote16postDonaldTrump)                                      0.00     0.28
## sd(presvote16postEvanMcMullin)                                     0.02     1.89
## sd(presvote16postGaryJohnson)                                      0.02     1.07
## sd(presvote16postHillaryClinton)                                   0.02     0.69
## sd(presvote16postJillStein)                                        0.02     1.88
## cor(Intercept,presvote16postDonaldTrump)                          -0.82     0.61
## cor(Intercept,presvote16postEvanMcMullin)                         -0.70     0.70
## cor(presvote16postDonaldTrump,presvote16postEvanMcMullin)         -0.70     0.72
## cor(Intercept,presvote16postGaryJohnson)                          -0.70     0.71
## cor(presvote16postDonaldTrump,presvote16postGaryJohnson)          -0.67     0.73
## cor(presvote16postEvanMcMullin,presvote16postGaryJohnson)         -0.72     0.71
## cor(Intercept,presvote16postHillaryClinton)                       -0.66     0.73
## cor(presvote16postDonaldTrump,presvote16postHillaryClinton)       -0.72     0.68
## cor(presvote16postEvanMcMullin,presvote16postHillaryClinton)      -0.71     0.71
## cor(presvote16postGaryJohnson,presvote16postHillaryClinton)       -0.68     0.70
## cor(Intercept,presvote16postJillStein)                            -0.70     0.72
## cor(presvote16postDonaldTrump,presvote16postJillStein)            -0.73     0.68
## cor(presvote16postEvanMcMullin,presvote16postJillStein)           -0.70     0.69
## cor(presvote16postGaryJohnson,presvote16postJillStein)            -0.74     0.69
```

```
## cor(presvote16postHillaryClinton,presvote16postJillStein)       -0.68     0.73
##                                                                Rhat Bulk_ESS
## sd(Intercept)                                                  1.00     1386
## sd(presvote16postDonaldTrump)                                  1.00     2367
## sd(presvote16postEvanMcMullin)                                 1.00     2490
## sd(presvote16postGaryJohnson)                                  1.00     1913
## sd(presvote16postHillaryClinton)                               1.00     1031
## sd(presvote16postJillStein)                                    1.00     2166
## cor(Intercept,presvote16postDonaldTrump)                       1.00     5418
## cor(Intercept,presvote16postEvanMcMullin)                      1.00     7589
## cor(presvote16postDonaldTrump,presvote16postEvanMcMullin)      1.00     5693
## cor(Intercept,presvote16postGaryJohnson)                       1.00     5083
## cor(presvote16postDonaldTrump,presvote16postGaryJohnson)       1.00     3944
## cor(presvote16postEvanMcMullin,presvote16postGaryJohnson)      1.00     3604
## cor(Intercept,presvote16postHillaryClinton)                    1.00     3959
## cor(presvote16postDonaldTrump,presvote16postHillaryClinton)    1.00     2959
## cor(presvote16postEvanMcMullin,presvote16postHillaryClinton)   1.00     2875
## cor(presvote16postGaryJohnson,presvote16postHillaryClinton)    1.00     3112
## cor(Intercept,presvote16postJillStein)                         1.00     7275
## cor(presvote16postDonaldTrump,presvote16postJillStein)         1.00     5255
## cor(presvote16postEvanMcMullin,presvote16postJillStein)        1.00     4495
## cor(presvote16postGaryJohnson,presvote16postJillStein)         1.00     3493
## cor(presvote16postHillaryClinton,presvote16postJillStein)      1.00     2940
##                                                                Tail_ESS
## sd(Intercept)                                                      2001
## sd(presvote16postDonaldTrump)                                      2345
## sd(presvote16postEvanMcMullin)                                     2312
## sd(presvote16postGaryJohnson)                                      1959
## sd(presvote16postHillaryClinton)                                   2087
## sd(presvote16postJillStein)                                        2272
## cor(Intercept,presvote16postDonaldTrump)                           3131
## cor(Intercept,presvote16postEvanMcMullin)                          2588
## cor(presvote16postDonaldTrump,presvote16postEvanMcMullin)          3440
## cor(Intercept,presvote16postGaryJohnson)                           2944
## cor(presvote16postDonaldTrump,presvote16postGaryJohnson)           3079
## cor(presvote16postEvanMcMullin,presvote16postGaryJohnson)          3233
## cor(Intercept,presvote16postHillaryClinton)                        2808
## cor(presvote16postDonaldTrump,presvote16postHillaryClinton)        2957
## cor(presvote16postEvanMcMullin,presvote16postHillaryClinton)       3412
## cor(presvote16postGaryJohnson,presvote16postHillaryClinton)        3204
## cor(Intercept,presvote16postJillStein)                             3017
## cor(presvote16postDonaldTrump,presvote16postJillStein)             3462
## cor(presvote16postEvanMcMullin,presvote16postJillStein)            2898
## cor(presvote16postGaryJohnson,presvote16postJillStein)             3120
## cor(presvote16postHillaryClinton,presvote16postJillStein)          3529
##
## Population-Level Effects:
##                             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept[1]                   -0.28      0.08    -0.44    -0.11 1.00     6990
## Intercept[2]                    0.59      0.08     0.43     0.76 1.00     7127
## Intercept[3]                    2.07      0.10     1.88     2.27 1.00     5744
## presvote16postDonaldTrump       2.68      0.12     2.46     2.91 1.00     5663
## presvote16postEvanMcMullin      0.25      0.57    -0.84     1.38 1.00     6832
## presvote16postGaryJohnson       0.12      0.25    -0.39     0.61 1.00     5921
```

```
## presvote16postHillaryClinton    -2.30      0.16    -2.63    -2.02 1.00     3419
## presvote16postJillStein         -2.21      0.53    -3.39    -1.30 1.00     4598
##                              Tail_ESS
## Intercept[1]                     2732
## Intercept[2]                     3651
## Intercept[3]                     3490
## presvote16postDonaldTrump        3244
## presvote16postEvanMcMullin       3098
## presvote16postGaryJohnson        3027
## presvote16postHillaryClinton     2674
## presvote16postJillStein          2568
##
## 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).
```

Simply using loo_compare would suggest that the fifth model is best, though the standard error on the difference with the third model would suggest that the decision is slim. The fifth model would indicate that there is quite a bit of state level heterogeneity. Since the predictive variable is an unordered factor (i.e. factored by who the respondent voted for in 2016 but with no order enforced on candidates), we can look to the standard deviation in the estimates for each level of the factor. For those who voted for President Trump in 2016, there seems to be very little state-state variation in estimates (sd = 0.1), while those who voted for Hillary Clinton seem to have varied a bit from state to state. This could indicate that voter profile for respondents who voted for Clinton is liked to more state-by-state variation in approval (i.e. the profile could be less homogenous), while those who voted for Trump are more consistent state-to-state. For those who voted for the other candidates, the sample numbers ar much smaller, but the standard deviation in estimates appears to be quite large compared to the previous two factor levels, with the estimation in state level heterogeneity reaching above .4 for all of the other three factor levels.

However, we know that using elpd for the comparison here is somewhat inappropriate since elpd is monotonically related the model's fit to the current data (i.e. maximizing the peak in the simulated data is closely related to maximizing the fit to the likelihood) and hierarchical models are not built to maximize that very fit. Instead, hierarchical models are built to maximize the fit to potential future data and do not prioritize the expected lpd value. As a result, basing the deicision on just elpd would be ill advised.

1.2 Binary or Ordinal

```
AFS$app_dtrmp_binary <- AFS$app_dtrmp %in% c("Strongly approve", "Somewhat approve")

Pr <- pp_expect(model5)

ll <- dbinom(x = AFS$app_dtrmp_binary, size = 1, log = TRUE,
             prob = apply(Pr, MARGIN = 1:2, FUN = function(p) p[1] + p[2]))
ll_loo <- loo(ll)

model5_2 <- brm(formula = app_dtrmp_binary ~ 1 + presvote16post +
                  (1 + presvote16post| inputstate),
                  data = AFS,
                  family = bernoulli(),
                  prior = base_prior)
```

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

```
## Trying to compile a simple C file

## Start sampling
```

```
model5_2_loo <- loo(model5_2)
```

```
ll_loo
```

```
##
## Computed from 4000 by 2725 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -4679.0 37.7
## p_loo      2627.4 32.9
## looic      9358.0 75.3
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     2681  98.4%   1212
##  (0.5, 0.7]   (ok)         12   0.4%   865
##    (0.7, 1]   (bad)        16   0.6%   28
##    (1, Inf)   (very bad)   16   0.6%   11
## See help('pareto-k-diagnostic') for details.
```

```
model5_2_loo
```

```
##
## Computed from 4000 by 2939 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -1100.3 33.0
## p_loo        26.5  2.8
## looic      2200.6 66.0
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     2901  98.7%   1219
##  (0.5, 0.7]   (ok)         33   1.1%   1369
##    (0.7, 1]   (bad)         5   0.2%   77
##    (1, Inf)   (very bad)    0   0.0%   <NA>
## See help('pareto-k-diagnostic') for details.
```

Comparing the two elpd's it is clear that the binary encoding is far better for fit (i.e. the elpd is over 3000 larger, many multiples of the standard error value). This would clearly lend support to political scientists' preference for the binary encoded response variable.

1.3 Predicting States

First, we generate the predictions:

```
predict <- posterior_predict(model5_2)
```

Next, to calculate the posterior probability that Trump will win a state, we need to use the predictions to determine the percentage of simulations in which Trump won a majority of votes for that state. We can do this with a set of for loops extracting the appropriate simulations for each state and determining the winner (under the given assumptions) in each simulation:

```
states = sort(unique(AFS$inputstate))
states_prob = c()

for (i in states){
  obs = c()
  for (j in 1:nrow(AFS)){
    if (AFS[j,"inputstate"]==i){
      obs = cbind(obs, predict[,j])
      }
    else{
      next
      }
    }

  trump_wins_sim = c()
  for (k in 1:nrow(obs)){
    trump_votes = sum(obs[k,])/ncol(obs)
    outcome = ifelse(trump_votes>=0.5,1,0)
  trump_wins_sim = c(trump_wins_sim, outcome)
  }
  states_prob = c(states_prob,round(mean(trump_wins_sim),3))
}
trump_win = data.frame(states,states_prob)
```

```
trump_win
```

```
##                   states states_prob
## 1                Alabama       0.034
## 2                 Alaska       0.862
## 3                Arizona       0.005
## 4               Arkansas       0.889
## 5             California       0.000
## 6               Colorado       0.002
## 7            Connecticut       0.034
## 8               Delaware       0.335
## 9  District of Columbia       0.174
## 10               Florida       0.002
## 11               Georgia       0.001
## 12                Hawaii       0.057
## 13                 Idaho       0.416
## 14              Illinois       0.000
## 15               Indiana       0.001
```

```
## 16                Iowa       0.043
## 17              Kansas       0.165
## 18            Kentucky       0.398
## 19           Louisiana       0.018
## 20               Maine       0.482
## 21            Maryland       0.030
## 22       Massachusetts       0.000
## 23            Michigan       0.055
## 24           Minnesota       0.083
## 25         Mississippi       0.094
## 26            Missouri       0.368
## 27             Montana       0.015
## 28            Nebraska       0.182
## 29              Nevada       0.010
## 30       New Hampshire       0.175
## 31          New Jersey       0.011
## 32          New Mexico       0.382
## 33            New York       0.000
## 34      North Carolina       0.004
## 35        North Dakota       0.885
## 36                Ohio       0.002
## 37            Oklahoma       0.641
## 38              Oregon       0.000
## 39        Pennsylvania       0.004
## 40        Rhode Island       0.177
## 41      South Carolina       0.006
## 42        South Dakota       0.710
## 43           Tennessee       0.209
## 44               Texas       0.000
## 45                Utah       0.777
## 46             Vermont       0.038
## 47            Virginia       0.001
## 48          Washington       0.023
## 49       West Virginia       0.374
## 50           Wisconsin       0.022
## 51             Wyoming       0.215
```

Using these predictions is likely to be better than just using the observed data since the latter method would overfit. In other words, using the observed approvals and not accounting for state-level heterogeneity would produce a model fit solely and tightly around the approval ratings, which is likely to overfit to the observed data. Instead, the hierarchical nature (i.e. pooling of estimates) allows for better fit to predicting future data, which would include the election.

2. Discrimination in Police Stops

2.1 Prior Predictive Distribution

After exposing the function, PPD is a produced set of tuple estimates for number of searches and hits:

```
north_carolina <- readRDS("north_carolina.rds")

rstan::expose_stan_functions("NC_rng.stan")
```

```
## Trying to compile a simple C file
```

```
PPD <- NC_rng(D = nrow(north_carolina), R = ncol(north_carolina),
              Asian = north_carolina[ , 1], Black = north_carolina[ , 2],
              Hispanic = north_carolina[ , 3], White = north_carolina[ , 4])
```

2.2 Legal Analysis

From the state level perspective, the attorney's argument would have no merit since the model would give an accurate estimation of trends across the state and to take issue with pooling the data of precincts would be to assert that departments operate completely independent of one another (a seemingly absurb assumption considering the constant collaboration needed to effectively police a state).

From the perspective of a particular precinct, the merits of the attorney's argument are less straightforward. First, from a statistical perspective, the data available are sufficient for the model to be fully specified and to produce accurate threshold and signal distribution estimates. As a result, it is specifically the pooling of information that allows for the accurate estimation of thresholds for a given department/race pair and the judge should not grant the attorney's request.

However, while the estimation of the distribution of thresholds is possible through the hierarchical models, the estimation is better for judging thresholds for department-race pairs relative to other pairs. To be more concrete, the estimation is a somewhat better for a relative assessment of racial discrimination among department-race pairs, but will somwhat bias individual estimates towards the mean of the posterior distribution. From the perspective of the attorney, requesting that the hierarchical model not be admissable could be good if they are representing a department free of racial discrimination (relative to other departments) since the distribution of that department's estimations for thresholds will be pulled somewhat by the information of dpeartment will more prevalent racial discrimination. However, making this argument could be bad for the attorney if they are representing a racist deparment, in which they would benefit from the slight bias. Regardless of the specifics of the department, the attorney might find merit in the eyes of a judge for throwing the model out since the U.S. legal system is designed to try people on their actions alone and not in the context of the influence of others, an assumption not held for the precincts analyzed using the hierarchical model.