Jesse Geman
Data 2020
May 3rd, 2018

Data 2020: Final Project

Introduction

Given the dataset, there are many ways to build multilevel models. I experimented with a few different models, mostly the results were discouraging — but I am confident there is a lot of room for improvement. In the following paragraphs I will describe my process and my findings.

Selecting a level

After the discussions last class I decided it made sense to use County and State as levels. There were 47 states in our dataset with recorded police killings and 238 counties. Intuitively, it seemed that the states would yield more informative results simply because there were more recorded killings in each state. With that said, I could imagine there being more variance in the factors related to police killings across a state as opposed to a county. This could be a potential downside.

It would also be interesting to use the race of the deceased as a level and train the model to predict something like census tract per capita income, or unemployment. Or, we could go the other way, and split the tracts into quintiles based on per capita income or unemployment and, using the quintiles as levels, try to predict the race of the deceased. In my exploratory analysis report I mentioned that I was worried about arriving at any conclusions in my inference of the data. Specifically, I thought that income could be a confounding factor. This could be a way to control for that (of course there are possible routes to explore this that would not involve multilevel models).

Another interesting level to consider would be the “armed” variable. It is easy to imagine there would be different slopes or intercepts based on this value of this category.

As of now, I have made models with state and county as levels.

Pre-Processing The Data

This proved more time consuming that anticipated. No surprise there. More than anything, I had to figure out a way to aggregate the census data for each state and county. Below is the code for how I did this for each state. The process for aggregating the county level data was the same.

```r

toytableState = acs %>%
  group_by(State) %>%
  summarise(TotalState = sum(TotalPop),
            StateMen = sum(Men)/TotalState,
            #TotalWomenState = sum(Women),
            StateUnemployment =  (sum(TotalPop*(Unemployment*.01)))/TotalState,
            StateIncomePerCap = sum(TotalPop*IncomePerCap)/TotalState,
            StatePoverty = (sum(TotalPop*(Poverty*.01)))/TotalState,
            StateFracHispanic = (sum(TotalPop*(Hispanic*.01)))/TotalState,
            StateFracBlack = (sum(TotalPop*(Black*.01)))/TotalState,
            StateFracWhite = (sum(TotalPop*(White*.01)))/TotalState)

```

<u>Building a Model</u>

This was fairly trivial. One thing to note was that the there weren't enough observations to build a county level varying slope model, which is unfortunate, because it would have been interesting to compare the county level varying slope model to the state level varying slope model. It's also validating, remember I voiced my concern about the number of counties with recorded police killings in the <u>selecting a level</u> section. In addition, the model failed to converge when I tried to give the armed feature a varying slope in the state level model.

```r
InterceptStateModel <-glmer(Minority ~ age + gender + armed + (1|State),
                    family = binomial("logit"), data = datastateuse)

#unfortunately the model fails to converge for feautres other than age and
#gender having varying coefficients
OneSlopeStateModel <-glmer(Minority ~ age + gender + armed +(1+age|State),
                    family = binomial(link = "logit"), data = datastateuse)
TwoSlopeStateModel <-glmer(Minority ~ age + gender + armed +(1+age+gender|State),
                       family = binomial(link = "logit"), data = datastateuse)

InterceptCountyModel <- glmer(Minority ~ age + gender + armed + (1|County),
                              family = binomial(link = "logit"), data = datacountyuse)

#Unfortunately, we don't have enough observations to run a varying slope model for the county
#SlopeCountyModel <- glmer(Minority ~ age + gender + (1+age+gender|County),
#                              family = binomial("logit"), data = datacountyuse)
```

<u>Model Assessment</u>

The results were nothing to write home about. When I compared the three state models together the AIC and BIC scores varied marginally corresponding to the model degrees of freedom. The likelihood values were all the same as were the deviances. Below you can see the results just described, obtained using the “anova” function. The chisq value for the varying slope model where the age feature had a varying coefficient was significant.

```
> ## Compare models
> kable(anova(InterceptStateModel, OneSlopeStateModel,TwoSlopeStateModel), type = "pandoc", caption = "Table 1: Comparison of varying/intercept slope and varying intercept on State")


|                    | Df|      AIC|      BIC|    logLik| deviance|     Chisq| Chi Df| Pr(>Chisq)|
|:-------------------|--:|--------:|--------:|---------:|--------:|---------:|------:|----------:|
|InterceptStateModel |  5| 551.3337| 571.5469| -270.6668| 541.3337|        NA|     NA|         NA|
|OneSlopeStateModel  |  7| 555.3030| 583.6015| -270.6515| 541.3030| 0.0306642|      2|  0.9847848|
|TwoSlopeStateModel  | 10| 560.9640| 601.3903| -270.4820| 540.9640| 0.3390739|      3|  0.9525197|
```

Below are some diagnostics obtained from the county level model. They are similar to the ones above though the AIC and BIC numbers are smaller (there are 5 degrees of freedom, hence df.resid = 416).

```
   AIC      BIC   logLik deviance df.resid
 547.5    567.7   -268.7    537.5      416
```

<u>Conclusions</u>

As I mentioned there is still much to do! Obviously, my results leave something to be desired. Additionally, I would like to run more diagnostics to get a better sense of just how bad the models are. Nonetheless, I successfully built a few multilevel models, something that had eluded me up until this point and I feel like I have the start of a foundation to build on. I look forward to building different multilevel models and improving the ones I have.[1]

[1] I have not included my code for this section since it was all fairly straightforward and boilerplate.