CGS698C, Assignment 04

Himanshu Yadav

Part 1: A simple linear regression: Power posing and testosterone

Download the file `df_powerpose.csv` and load the following data set:

```
df_powerpose <- read.table("df_powerpose.csv",header=T,sep=",")
head(df_powerpose)
```

```
##   X id hptreat female age testm1  testm2
## 1 2 29    High   Male  19 38.725  62.375
## 2 3 30     Low Female  20 32.770  29.235
## 3 4 31    High Female  20 32.320  27.510
## 4 5 32     Low Female  18 17.995  28.655
## 5 7 34     Low Female  21 73.580  44.670
## 6 8 35    High Female  20 80.695 105.485
```

The data set, which was originally published in Carney, Cuddy, and Yap (2010) but released in modified form by Fosse (2016), shows the testosterone levels of 39 different individuals, before and after treatment, where treatment refers to each individual being assigned to a high power pose or a low power pose. In the original paper by Carney, Cuddy, and Yap (2010), the unit given for testosterone measurement (estimated from saliva samples) was picograms per milliliter (pg/ml). One picogram per milliliter is 0.001 nanogram per milliliter (ng/ml).

The research hypothesis is that on average, assigning a subject a high power pose vs. a low power pose will lead to higher testosterone levels after treatment.

Investigate this claim using a linear model and weakly informative priors[1] in brms. You'll need to estimate the effect of the variable that encodes the change in testosterone.

[1] You can also use default priors of brms assuming you know nothing about typical ranges of testosterone using salivary measurement. But I would suggest you try to come up with weakly informative priors. Feel free to do a prior predictive check if you are unsure.

Part 2: Poisson regression models and hypothesis testing

Human language has an interesting, empirical property: When dependency arcs are drawn between syntactically-related words, they rarely cross each other. You can ignore this sentence if you do not have any background in Linguistics. I am just setting a motivation for the problem.

Any sentence from a human language would contain some easy structures and some difficult structures. A crossing dependency is a type of structure that is arguably difficult to process and is rarely found in natural languages.

The number of crossing dependencies in a sentence can be given by a Poisson distribution

$$N_i \sim Poisson(\lambda_i) \tag{1}$$

where N_i is the number of crossing dependencies in the sentence i; λ_i is rate parameter indicating the expected rate of crossing dependencies in the sentence i, such that

$$\log \lambda_i = \alpha + \beta L_i \tag{2}$$

where L_i is the length of the sentence i, α is the expected rate of crossings in a sentence of **average length** (say 11) and β is the change in rate of crossings as a function of sentence length.

(Note: Sentence length means the number of words in a sentence. For simplicity, assume that sentence lengths can range from 2 to 20.)

Exercise 2.1 I*mplement the model in R or Python such that the function gives the number of crossings as the outcome, and takes sentence length, α, and β as its arguments.*

Exercise 2.2 G*enerate prior predictions of the model for sentences of length* 4 *under the following prior assumptions*

$$\alpha \sim Normal_{lb=0}(0.15, 0.1) \tag{3}$$

$$\beta \sim Normal_{lb=0}(0.25, 0.05) \tag{4}$$

Exercise 2.3 C*onsider a dataset of crossing dependencies from English and German corpora, "crossing.csv". This dataset contains number of crossings for each sentence from each language. Fit the following two models, $\mathcal{M}1$ and $\mathcal{M}2$, to the given data.*

Model $\mathcal{M}1$

Assumption: The rate of crossings is only a function of sentence length and it remains exactly the same in English and German.}

This assumption can be represented by the following regression.

$$N_{i,j} \sim Poisson(\lambda_{i,j}) \tag{5}$$

where $N_{i,j}$ is the number of crossing dependencies in sentence i in language j; $\lambda_{i,j}$ is rate parameter indicating the expected rate of crossing dependencies in sentence i in language j, such that

$$\log \lambda_{i,j} = \alpha + \beta L_{i,j} \tag{6}$$

where $L_{i,j}$ is the length of sentence i of language j.

The above model implies the average rate of crossings depends only on the sentence length.

Model $\mathcal{M}2$

Assumption: As sentence length increases, the number crossings grows at a different rate in English vs. German.}

$$N_{i,j} \sim Poisson(\lambda_{i,j}) \tag{7}$$

where $N_{i,j}$ is the number of crossing dependencies in sentence i in language j; $\lambda_{i,j}$ is rate parameter indicating the expected rate of crossing dependencies in sentence i in language j, such that

$$\log \lambda_{i,j} = \alpha + \beta L_{i,j} + \beta_{language} R_j + \beta_{interact} L_{i,j} * R_j \tag{8}$$

where $L_{i,j}$ is the length of sentence i of language j, R_j is the indicator variable such that $R_j = 0$ if the language is English and $R_j = 1$ if the language is German.

The above model implies that the average rate of crossings depends on the sentence length as well the language (English vs German).

Load the data file "crossings.csv" and fit the above two models to the data and estimate all the parameters. You can use "brms" to fit the above models, you will have to use Poisson family for the likelihood. Use the following priors:

$$\alpha \sim Normal(0.15, 0.1) \tag{9}$$

$$\beta \sim Normal(0, 0.15) \tag{10}$$

$$\beta_{language} \sim Normal(0, 0.15) \tag{11}$$

$$\beta_{interact} \sim Normal(0, 0.15) \tag{12}$$

Exercise 2.4 Quantify evidence for the models $\mathcal{M}1$ and $\mathcal{M}2$ using k-fold cross-validation.

Here is a sample code to do the same.

```
observed <- read.table("crossings.csv",sep=",",header=T)
# Visualize average rate of crossings
observed %>% group_by(Language,s.length) %>%
  summarise(mean.crossings=mean(nCross)) %>%
  ggplot(aes(x=s.length,y=mean.crossings,
             group=Language,color=Language))+
  geom_point()+geom_line()

# Code/center the predictors
observed$s.length <- observed$s.length - mean(observed$s.length)
observed$lang <- ifelse(observed$Language=="German",1,0)

# These two vectors will store log predictive desnities
# in each fold

lpds.m1 <- c()
lpds.m2 <- c()

untested <- observed
for(k in 1:5){
  # Prepare test data and training data
  ytest <- sample_n(untested,size=nrow(observed)/5)
  ytrain <- setdiff(observed,ytest)
```

```
  untested <- setdiff(untested,ytest)
  # Fit the models M1 and M2 on training data
  fit.m1 <-
    brm(nCross ~ 1 + s.length,data=ytrain,
        family = poisson(link = "log"),
        prior = c(prior(normal(0.15, 0.1), class = Intercept),
                  prior(normal(0, 0.15), class = b)),
        cores=4)

  fit.m2 <-
    brm(nCross ~ 1 + s.length + lang + s.length*lang,
        data=ytrain,
        family = poisson(link = "log"),
        prior = c(prior(normal(0.15, 0.1), class = Intercept),
                  prior(normal(0, 0.15), class = b)),
        cores=4)

  # retrieve posterior samples
  post.m1 <- posterior_samples(fit.m1)
  post.m2 <- posterior_samples(fit.m2)

  # Calculate log pointwise predcitive density using test data
  lppd.m1 <- 0
  lppd.m2 <- 0
  for(i in 1:nrow(ytest)){
    lpd_im1 <- log(mean(dpois(ytest[i,]$nCross,
                lambda=exp(post.m1[,1]+
                post.m1[,2]*ytest[i,]$s.length))))
    lppd.m1 <- lppd.m1 + lpd_im1
    lpd_im2 <- log(mean(dpois(ytest[i,]$nCross,
                lambda=exp(post.m2[,1]+
                post.m2[,2]*ytest[i,]$s.length+
                  post.m2[,3]*ytest[i,]$lang+
                  post.m2[,4]*ytest[i,]$s.length*ytest[i,]$lang)
                  )))
    lppd.m2 <- lppd.m2 + lpd_im2
  }
  lpds.m1 <- c(lpds.m1,lppd.m1)
  lpds.m2 <- c(lpds.m2,lppd.m2)
}

# Predictive accuracy of model M1
elpd.m1 <- sum(lpds.m1)
```

```r
# Predictive accuracy of model M2
elpd.m2 <- sum(lpds.m2)

# Evidence in favor of M2 over M1
difference_elpd <- elpd.m2-elpd.m1
```