CSC 578B Assignment 1

Derek Robinson

Date: 2021-10-05 12:48:55

Contents

First we load our dataset and clean it up a little

```
#load the data
df <- read.csv('data.csv')
# Clean up an entry which had "OT" instead of "OT"
df["technique"][df["technique"] == "OT"] <- "OT"
# convert category and technique (factor) to numeric
df$category <- as.factor(df$category)
df$technique <- as.factor(df$technique)
# Take a peak at the data
head(df)
```

```
##   subject category technique tp
## 1       1       LE        NT  5
## 2       1       LE        OT  6
## 3       2       LE        NT  3
## 4       2       LE        OT  3
## 5       3       LE        NT  7
## 6       3       LE        OT  3
```

Step 1: Data Description and Descriptive Statistics

In the following table are the descriptions of each column in the dataset:

Column Name	Description
Subject	A unique indentifier for each subject (participant) of the study
Category	Describes if the subject was more experience (ME) or less experienced (LE)
Technique	Which technique was being used, either new technique (NT) or old technique (OT)

Column Name	Description
tp (True Positives)	The number of faults classified as true faults found by the subject

The most basic descriptive statistics are the mean and the variance. We will calculate both the mean and the variance of the true positives (`tp`).

The mean of the true positives is equal to 4.7785714 and the variance of the true positives is equal to 4.0873073.

Step 2: A Defense of the likelihoods

First we have to decide on a likelihood for `tp`. Since `tp` takes on positive natural numbers ($\mathbb{N}^+$), we will use the $\text{Poisson}(\lambda)$ distribution as this is a commonly used likelihood for this kind of data.

We will create a series of models $\mathbf{M} = \{\mathcal{M}_0, \ldots, \mathcal{M}_n\}$ and see how well the compare. After we have compared them we will choose our final model.

Before we start creating models we need to choose our prior for the lambda parameter of the poisson function. We already know the mean and variance of `tp`, so lets use those to choose a prior. We can guess that the tp should probably be below 10 most of time. First we will take a look at the default prior $\text{Normal}(0, 10)$.

```
max(rnorm(70, 0, 10))
```

```
## [1] 25.81959
```

That seems a little high, lets try $\text{Normal}(0, 2.5)$

```
max(rnorm(70, 0, 2.5))
```

```
## [1] 6.114207
```

That looks pretty good. Lets now make our first model which contains no predictor variables.

Model 0

```
m0 <- ulam(
  alist(
    tp ~ poisson(lambda),
    log(lambda) <- alpha, # log link
    alpha ~ normal(0, 2.5)
  ), data = df, cores = 2, chains = 4, cmdstan = TRUE, log_lik = TRUE, iter = 5e3
)
```

and lets check the diagnostics for model 0

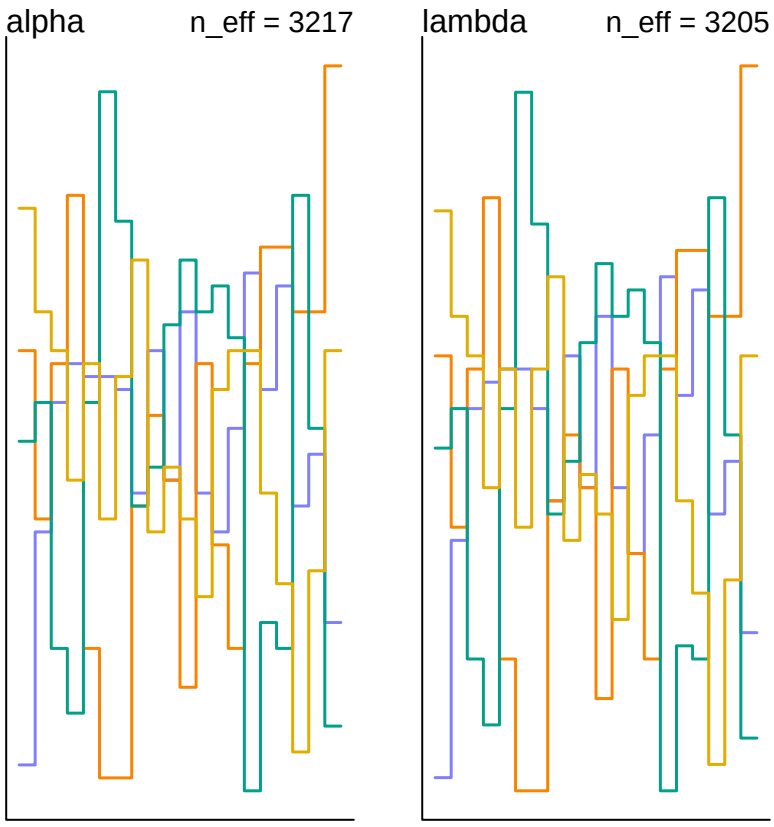
```
precis(m0)
```

```
##           mean         sd     5.5%    94.5%    n_eff    Rhat4
## alpha 1.563181 0.03836779 1.500848 1.625693 3217.161 1.001416
```

Okay, so `n_eff` is in the thousands and $\widehat{R} < 1.01$, that's good.

Let's also have a look at the trankplots for model 0.

```
trankplot(m0)
```



That's what we like to see, all the chains mixing well after the initial phase.

Now lets create several more models, one with category as the predictor, another with technique as the predictor, and then finally using both category and technique as the predictor.

Model 1

```
m1 <- ulam(
  alist(
    tp ~ poisson(lambda),
    log(lambda) <- alpha + beta_category*category, # log link
    alpha ~ normal(0, 2.5),
    beta_category ~ normal(0, 1)
  ), data = df, cores = 2, chains = 4, cmdstan = TRUE, log_lik = TRUE, iter = 5e3
)
```

again, check the diagnositcs

```
precis(m1, depth = 2)
```

```
##                      mean         sd       5.5%      94.5%    n_eff    Rhat4
## alpha          1.69253890 0.11555634  1.5044667 1.87571165 2244.100 1.000952
## beta_category -0.09770484 0.08253647 -0.2305323 0.03621894 2214.618 1.001303
```

Again, `n_eff` is in the thousands and $\widehat{R} < 1.01$

Model 2

```
m2 <- ulam(
  alist(
    tp ~ poisson(lambda),
    log(lambda) <- alpha + beta_technique*technique, # log link
    alpha ~ normal(0, 2.5),
    beta_technique ~ normal(0, 1)
  ), data = df, cores = 2, chains = 4, cmdstan = TRUE, log_lik = TRUE, iter = 5e3
)
```

Again, check the diagnostics

```
precis(m2, depth = 3)
```

```
##                       mean         sd       5.5%        94.5%    n_eff    Rhat4
## alpha           1.7517017 0.11763290  1.5648912  1.939892200 2213.589 1.000017
## beta_technique -0.1276549 0.07604374 -0.2514773 -0.006846689 2197.856 1.000071
```

Again, `n_eff` is in the thousands and $\widehat{R} < 1.01$

Model 3 (Final Model)

```
m3 <- ulam(
  alist(
    tp ~ poisson(lambda),
    log(lambda) <- alpha + beta_category*category + beta_technique*technique, # log link
    alpha ~ normal(0, 2.5),
    beta_category ~ normal(0, 1),
    beta_technique ~ normal(0, 1)
  ), data = df, cores = 2, chains = 4, cmdstan = TRUE, log_lik = TRUE, iter = 5e3
)
```

For the final time, lets check the diagnostics

```
precis(m3, depth = 4)
```

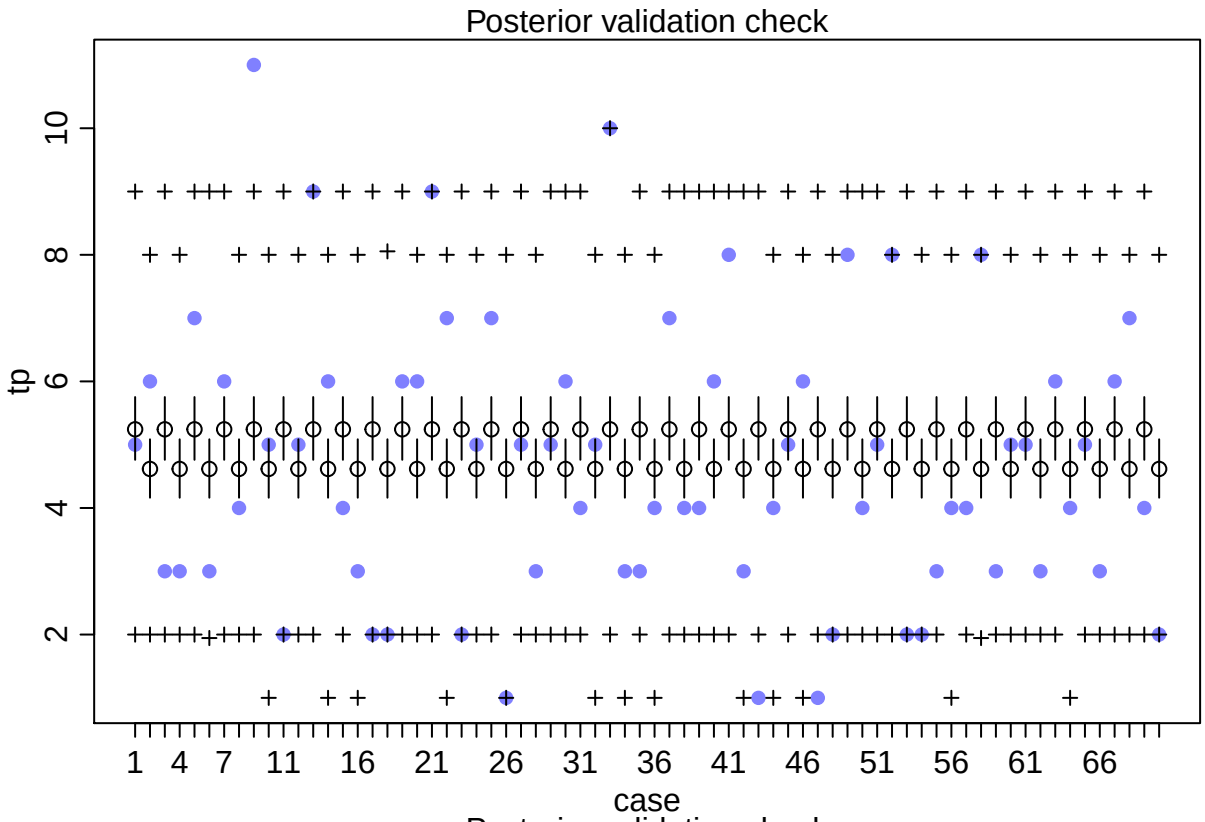
```
##                       mean         sd       5.5%        94.5%    n_eff    Rhat4
## alpha           1.87927775 0.16261693  1.6198659  2.144023300 3336.057 1.000141
## beta_category  -0.09673589 0.08185840 -0.2278951  0.032297523 4023.144 1.000519
## beta_technique -0.12740575 0.07839842 -0.2538082 -0.003977994 3803.862 1.000012
```

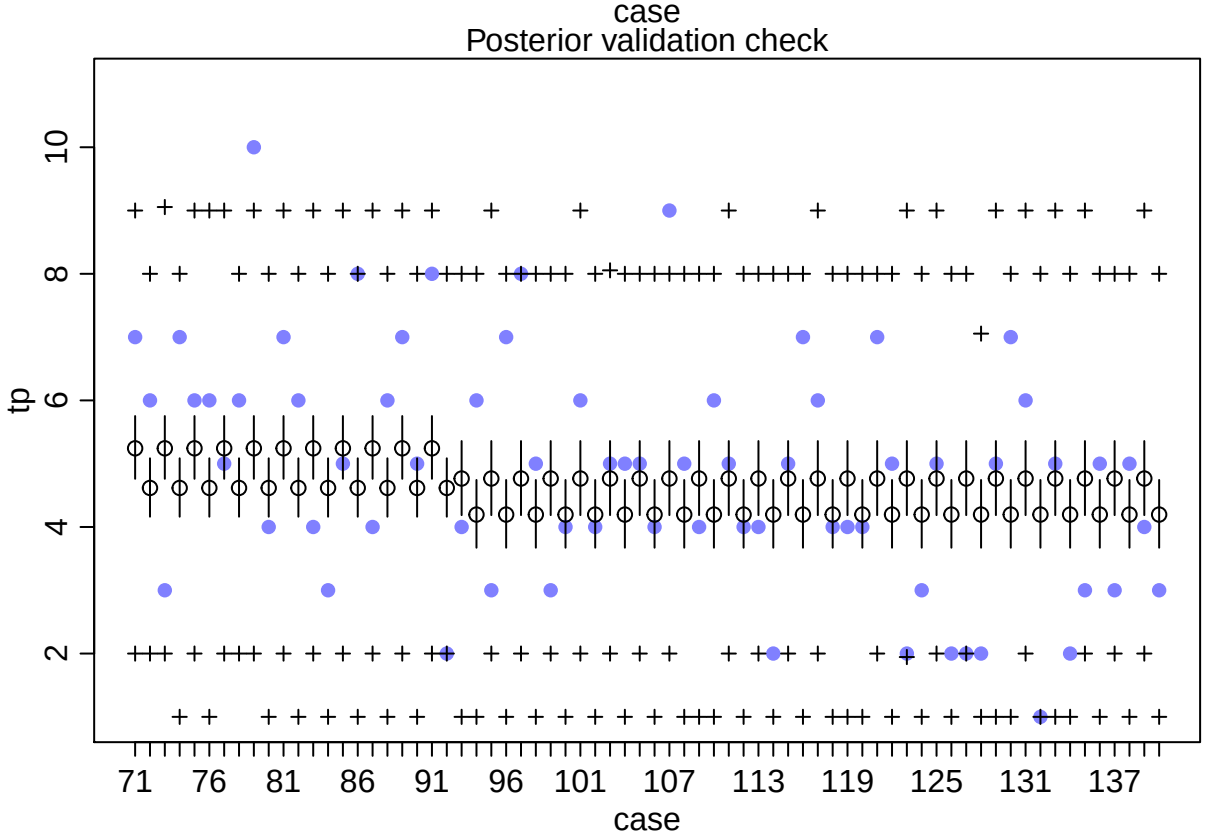
Perfect, we have `n_eff` in the thousands and $\widehat{R} < 1.01$.

Step 3: A Discussion of the Priors

First off, let us start by performing a prior predictive check of our final model, model 3

```
postcheck(m3, window = 70)
```





So it seems that our model is not perfect, but we don't want it to be anyways as this would cause over fitting. It seems like model 3 is reasonable and that our priors are also reasonable. Lets us see what happens when we change them to the default prior of $\mathsf{Normal}(0, 10)$.

```
m3_changed <- ulam(
  alist(
    tp ~ poisson(lambda),
    log(lambda) <- alpha + beta_category*category + beta_technique*technique, # log link
    alpha ~ normal(0, 10),
    beta_category ~ normal(0, 10),
    beta_technique ~ normal(0, 10)
  ), data = df, cores = 2, chains = 4, cmdstan = TRUE, log_lik = TRUE, iter = 5e3
)
```

Let's check the diagnostics now that we have changed the priors.

```
precis(m3_changed, depth = 4)
```

```
##                       mean         sd       5.5%        94.5%    n_eff    Rhat4
## alpha           1.88521660 0.16407158  1.6259334  2.152361650 3106.605 1.001321
## beta_category  -0.09787057 0.08404382 -0.2338138  0.035201387 3452.113 1.000839
## beta_technique -0.12997726 0.07807970 -0.2539360 -0.003302358 3737.395 1.001120
```

Seems like the default prior of $\mathsf{Normal}(0, 10)$. would also work.

Step 4: Comparing the models using LOO

Lets compare our models now

```
(loo_est <- compare(m0, m1, m2, m3, func=LOO))
```

```
##        PSIS       SE     dPSIS      dSE     pPSIS    weight
## m2 590.5017 12.63204 0.0000000       NA 1.6130964 0.3446030
## m3 590.9565 12.47244 0.4547684 2.030127 2.3763735 0.2745158
## m0 591.5132 13.40058 1.0114183 2.995343 0.8401307 0.2078224
## m1 591.8793 13.14986 1.3775241 3.393493 1.5767756 0.1730587
```

Interestingly, m2 is considered the best model.

```
loo_est[2,3] + c(-1,1) * loo_est[2,4] * 1.96
```

```
## [1] -3.524281  4.433818
```

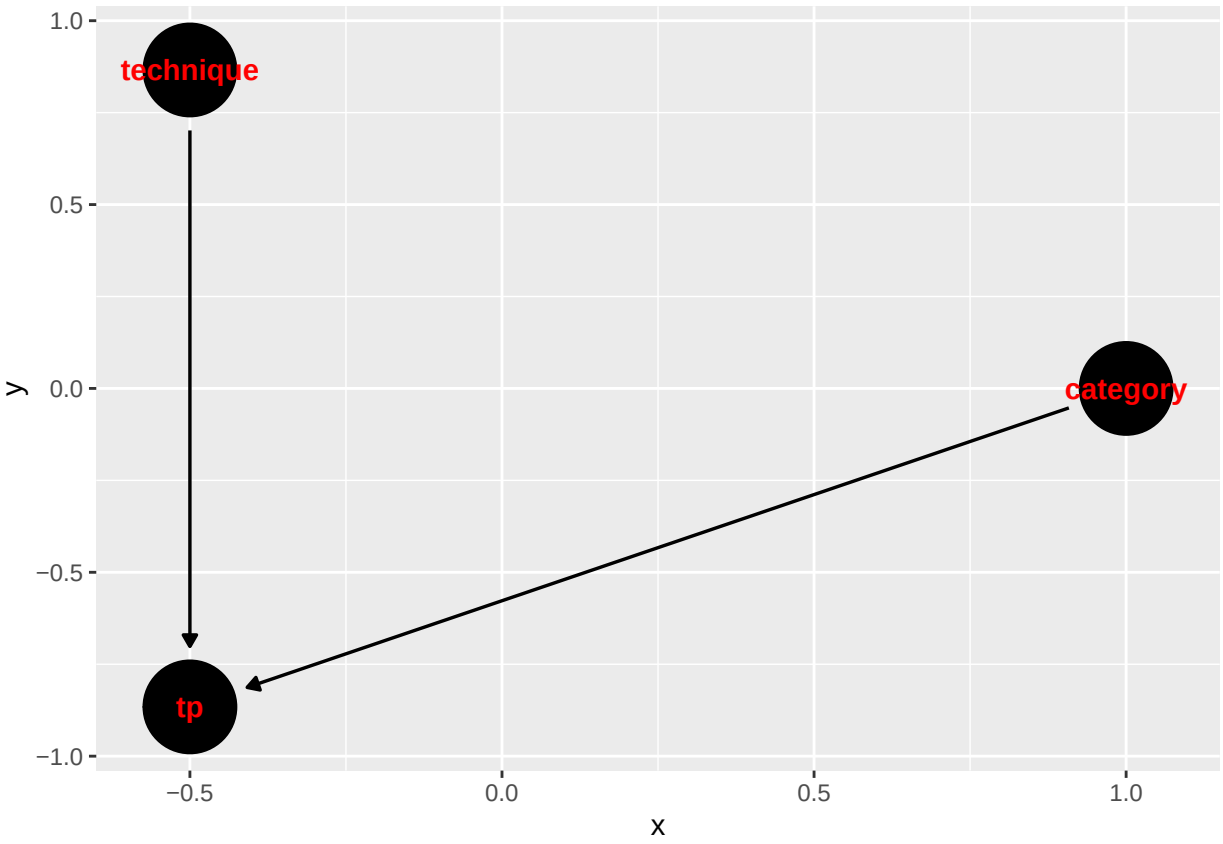
Step 5: Interpret Results

TODO

Step 6: Causal Model (DAG)

Below is the causal model used to model our assumptions

```
dag <- dagitty("dag{category -> tp <- technique}")
ggdag(dag, layout = "circle", text_col = "red")
```



Step 7: Presentation of diagnostics from running Stan on the 'final' model

TODO

Trankplots

TODO

R hat

TODO

Sample Size (`n_eff`)

TODO