Chapter 12 Questions

Melissa Van Bussel

August 2, 2018

Points for Chapter 12: 2 + 3 + 3 = 8 total points.

Easy Questions (2 points total)

12E1

a) will have more shrinkage.

12E2

There's obviously more than one way to do this, but here is one way:

$$
\begin{aligned}
y_i &\sim \text{Binomial}(1, p_i) \\
\text{logit}(p_i) &= \alpha_{group}[i] + \beta x_i \\
\alpha_{group} &\sim \text{Normal}(\mu, \sigma) \\
\beta &\sim \text{Normal}(0, 1) \\
\mu &\sim \text{Normal}(0, 10) \\
\sigma &\sim \text{Uniform}(0, 5)
\end{aligned}
$$

12E3

Again, there is obviously more than one way to do this (in fact, there are infinite ways to do this), but here is one example:

$$
\begin{aligned}
y_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &\sim \alpha_{group}[i] + \beta x_i \\
\alpha_{group} &\sim \text{Normal}(\mu_2, \sigma_2) \\
\beta &\sim \text{Normal}(0, 1) \\
\sigma &\sim \text{HalfCauchy}(0, 1) \\
\mu_2 &\sim \text{Normal, 0, 10} \\
\sigma_2 &\sim \text{HalfCauchy}(0, 1)
\end{aligned}
$$

12E4

Here is one example:

$$
\begin{aligned}
y_i &\sim \text{Poisson}(\lambda_i) \\
\log \lambda_i &= \alpha_{group}[i] + \beta x_i \\
\alpha_{group} &\sim \text{Normal}(\mu, \sigma) \\
\beta &\sim \text{Normal}(0, 1) \\
\mu &\sim \text{Normal}(0, 10) \\
\sigma &\sim \text{Uniform}(0, 5)
\end{aligned}
$$

12E5

Here is one example:

$$
\begin{aligned}
y_i &\sim \text{Poisson}(\lambda_i) \\
\log \lambda_i &= \alpha_{group}[i] + \alpha_{day}[i] + \beta x_i \\
\alpha_{group} &\sim \text{Normal}(\mu, \sigma) \\
\alpha_{day} &\sim \text{Normal}(0, \sigma_2) \\
\beta &\sim \text{Normal}(0, 1) \\
\mu &\sim \text{Normal}(0, 10) \\
\sigma &\sim \text{Uniform}(0, 5) \\
\sigma_2 &\sim \text{Uniform}(0, 5)
\end{aligned}
$$

#Medium Questions (points total)

12M3 (1 point)

```
library(rethinking)
```

```
## Loading required package: rstan
## Warning: package 'rstan' was built under R version 3.3.3
## Loading required package: ggplot2
## Warning: package 'ggplot2' was built under R version 3.3.3
## Loading required package: StanHeaders
## Warning: package 'StanHeaders' was built under R version 3.3.3
## rstan (Version 2.17.3, GitRev: 2e1f913d3ca3)
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
## Loading required package: parallel
## rethinking (Version 1.59)
```

```
data("reedfrogs")
d <- reedfrogs

# setup the data the same way as in the chapter
d$pred <- ifelse(d$pred == "no", 0, 1)
```

```
d$big <- ifelse(d$size == "big", 1, 0)
d$tank <- 1:nrow(d)

# model from the chapter
lmod12m3_gaussian <- map2stan(
        alist(
            surv ~ dbinom(density, p),
            logit(p) <- a_tank[tank],
            a_tank[tank] ~ dnorm(a, sigma_tank),
            a ~ dnorm(0, 10),
            sigma_tank ~ dcauchy(0, 1)
        ), data = d, chains = 4)
```

```
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Computing WAIC
```

```
## Constructing posterior predictions
```

```
## Aggregated binomial counts detected. Splitting to 0/1 outcome for WAIC calculation.
```

```
# cauchy model
lmod12m3_cauchy <- map2stan(
        alist(
            surv ~ dbinom(density, p),
            logit(p) <- a_tank[tank],
            a_tank[tank] ~ dcauchy(a, scale_tank),
            a ~ dnorm(0, 10),
            scale_tank ~ dcauchy(0, 1)
        ), data = d, chains = 4,
        control = list(adapt_delta = 0.99), cores = 4)
```

```
## Warning: There were 976 transitions after warmup that exceeded the maximum treedepth. Increase max_t
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

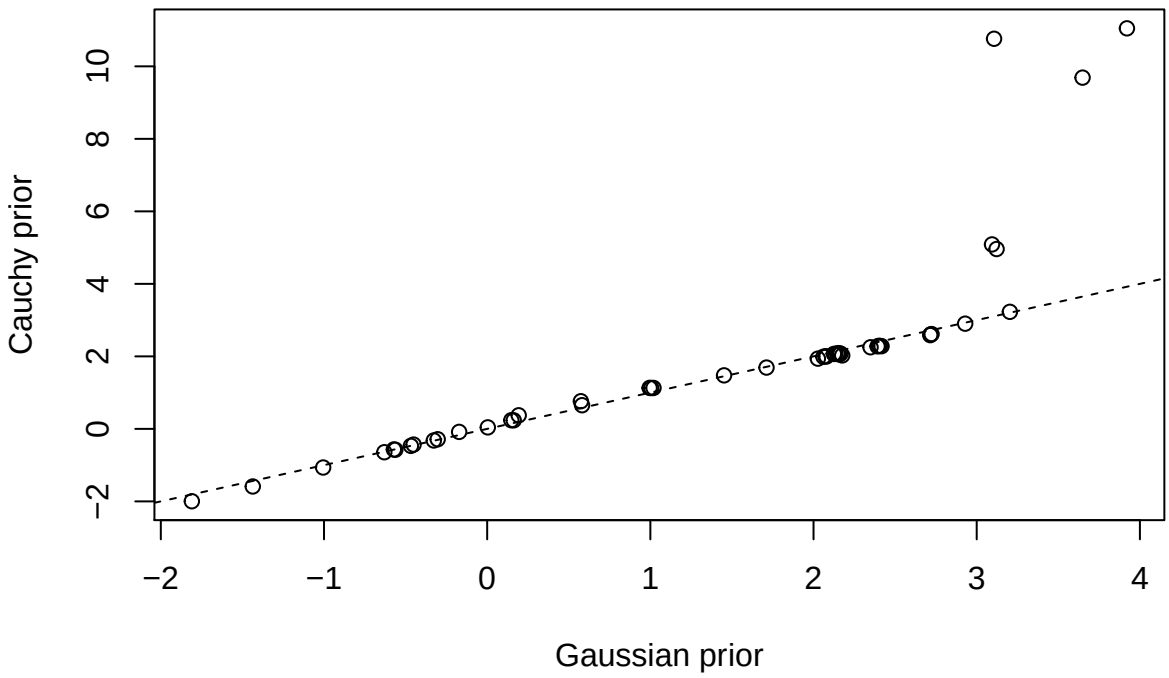
```
## Computing WAIC
```

```
## Constructing posterior predictions
```

```
## Aggregated binomial counts detected. Splitting to 0/1 outcome for WAIC calculation.
```

```
samples_gaussian <- extract.samples(lmod12m3_gaussian)
a_tank_gaussian <- apply(samples_gaussian$a_tank, 2, mean)
samples_cauchy <- extract.samples(lmod12m3_cauchy)
a_tank_cauchy <- apply(samples_cauchy$a_tank, 2, mean)

plot(a_tank_gaussian, a_tank_cauchy, xlab = "Gaussian prior",
     ylab="Cauchy prior")
abline(a = 0, b = 1, lty = 2)
```



The Gaussian prior causes more shrinkage of the extreme values than the Cauchy prior does, which is why there are a few extreme points on the plot.

12M4 (2 points)

```
data(chimpanzees)
d <- chimpanzees
d$recipient <- NULL # get rid of the NAs since Stan can't handle them
d$block_id <- d$block

# the model from the chapter
m12.5 <- map2stan(
          alist(
            pulled_left ~ dbinom(1, p),
            logit(p) <- a + a_actor[actor] + a_block[block_id] +
            (bp + bpc*condition)*prosoc_left,
            a_actor[actor] ~ dnorm(0, sigma_actor),
            a_block[block_id] ~ dnorm(0, sigma_block),
            c(a,bp,bpc) ~ dnorm(0, 10),
            sigma_actor ~ dcauchy(0, 1),
            sigma_block ~ dcauchy(0, 1)),
data = d)
```

```
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems

## Computing WAIC

## Constructing posterior predictions
```

```
# model for the question
lmod12m4 <- map2stan(
      alist(
        pulled_left ~ dbinom(1, p),
        logit(p) <- a_actor[actor] + a_block[block_id] +
        (bp + bpc*condition)*prosoc_left,
        a_actor[actor] ~ dnorm(a, sigma_actor),
        a_block[block_id] ~ dnorm(g, sigma_block),
        c(a,g,bp,bpc) ~ dnorm(0, 10),
        sigma_actor ~ dcauchy(0, 1),
        sigma_block ~ dcauchy(0, 1)),
data = d)
```

```
## Warning: There were 359 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: There were 18 transitions after warmup that exceeded the maximum treedepth. Increase max_tre
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems

## Computing WAIC
## Constructing posterior predictions

## Warning in map2stan(alist(pulled_left ~ dbinom(1, p), logit(p) <- a_actor[actor] + : There were 359 d
## Check the chains (trace plots, n_eff, Rhat) carefully to ensure they are valid.
```

```
precis(m12.5)
```

```
## 13 vector or matrix parameters omitted in display. Use depth=2 to show them.

##              Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## a            0.58   1.18      -0.98       2.16   152 1.02
## bp           0.81   0.27       0.34       1.18   359 1.00
## bpc         -0.12   0.31      -0.67       0.32   468 1.00
## sigma_actor  2.35   1.05       0.97       3.66   227 1.00
## sigma_block  0.25   0.21       0.01       0.47   135 1.00
```

```
precis(lmod12m4)
```

```
## Warning in precis(lmod12m4): There were 359 divergent iterations during sampling.
## Check the chains (trace plots, n_eff, Rhat) carefully to ensure they are valid.

## 13 vector or matrix parameters omitted in display. Use depth=2 to show them.

##              Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## a            3.52   8.07     -10.36      10.65    12 1.31
## g           -3.26   8.22      -9.21      11.52    11 1.33
## bp           0.89   0.20       0.57       1.23   314 1.03
## bpc         -0.15   0.23      -0.52       0.27   390 1.00
## sigma_actor  1.84   0.79       1.24       3.02     9 1.27
```

```
## sigma_block  0.17    0.14        0.07        0.34      7 1.34
```

The new model doesn't do nearly as well – the n_eff values aren't as good, and Rhat has not converged to 1.

Hard Questions (4 points total)

12H2 (2 points)

```
library(rethinking)
data(Trolley)
d <- Trolley
dat <- list(
  response = d$response,
  action = d$action,
  intention = d$intention,
  contact = d$contact,
  id = coerce_index(d$id)) # Use Richard's function to easily get the ids looking nicer
```

Now we want to create two different models

```
# no varying intercepts on individuals
lmod12h2_1 <- map2stan(
            alist(
              response ~ dordlogit(phi,cutpoints),
              phi <- bA*action + bI*intention + bC*contact,
              c(bA,bI,bC) ~ dnorm(0, 1),
              cutpoints ~ dnorm(0, 10)
            ), start = list(
                 cutpoints = seq(-2.5, 1.5, length.out = 6)),
data = dat)
```

```
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Computing WAIC
```

```
## Constructing posterior predictions
```

```
# varying intercepts on individuals
lmod12h2_2 <- map2stan(
            alist(
              response ~ dordlogit(phi,cutpoints),
              phi <- a_id[id] + bA*action + bI*intention + bC*contact,
              a_id[id] ~ dnorm(0, sigma_id),
              c(bA,bI,bC) ~ dnorm(0, 1),
              cutpoints ~ dnorm(0, 10),
              sigma_id ~ dcauchy(0, 1)
            ), start = list(
              cutpoints = seq(-2.5, 1.5, length.out = 6)),
data = dat)
```

```
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
## Computing WAIC
## Constructing posterior predictions
```

```
compare(lmod12h2_1, lmod12h2_2)
```

```
##               WAIC pWAIC dWAIC weight     SE   dSE
## lmod12h2_2 31340.8 356.2     0      1 177.70    NA
## lmod12h2_1 37089.8   9.0  5749      0  76.21 172.4
```

We can see that the second model (the varying intercepts model) takes all of the Akaike weight, so it is much more preferable than when we don't consider varying intercepts.

We can also look at the coefficients for each of the two models:

```
precis(lmod12h2_1)
```

```
## 6 vector or matrix parameters omitted in display. Use depth=2 to show them.
```

```
##     Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## bA -0.71   0.04      -0.78      -0.65   785    1
## bI -0.72   0.04      -0.78      -0.67  1000    1
## bC -0.96   0.05      -1.04      -0.88  1000    1
```

```
precis(lmod12h2_2)
```

```
## 337 vector or matrix parameters omitted in display. Use depth=2 to show them.
```

```
##           Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## bA       -0.96   0.04      -1.03      -0.89  1000    1
## bI       -0.95   0.04      -1.01      -0.89  1000    1
## bC       -1.27   0.05      -1.36      -1.20  1000    1
## sigma_id  1.89   0.08       1.76       2.02  1000    1
```

The coefficients are much stronger in the varying intercepts model.

12H3 (2 points)

We have to use Richard McElreath's **coerce_index()** function again.

```
d <- Trolley
dat <- list(
          response = d$response,
          action = d$action,
          intention = d$intention,
          contact = d$contact,
          id = coerce_index(d$id),
          story = coerce_index(d$story))

lmod12h3 <- map2stan(
          alist(
            response ~ dordlogit(phi,cutpoints),
            phi <- a_id[id] + a_story[story] +
            bA*action + bI*intention + bC*contact,
            a_id[id] ~ dnorm(0, sigma_id),
            a_story[story] ~ dnorm(0, sigma_story),
            c(bA,bI,bC) ~ dnorm(0, 1),
            cutpoints ~ dnorm(0, 10),
```

```
            sigma_id ~ dcauchy(0, 1),
            sigma_story ~ dcauchy(0, 1)
), start = list(
  cutpoints = seq(-2.5, 1.5, length.out = 6)),
data = dat, chains = 1, cores = 4)
```

```
## In file included from C:/Users/Melissa/Documents/R/win-library/3.3/BH/include/boost/config.hpp:39:0,
##                  from C:/Users/Melissa/Documents/R/win-library/3.3/BH/include/boost/math/tools/config
##                  from C:/Users/Melissa/Documents/R/win-library/3.3/StanHeaders/include/stan/math/rev/
##                  from C:/Users/Melissa/Documents/R/win-library/3.3/StanHeaders/include/stan/math/rev/
##                  from C:/Users/Melissa/Documents/R/win-library/3.3/StanHeaders/include/stan/math/rev/
##                  from C:/Users/Melissa/Documents/R/win-library/3.3/StanHeaders/include/stan/math/rev/
##                  from C:/Users/Melissa/Documents/R/win-library/3.3/StanHeaders/include/stan/math.hpp
##                  from C:/Users/Melissa/Documents/R/win-library/3.3/StanHeaders/include/src/stan/model
##                  from file1e286dc15ec.cpp:8:
## C:/Users/Melissa/Documents/R/win-library/3.3/BH/include/boost/config/compiler/gcc.hpp:186:0: warning:
##  #  define BOOST_NO_CXX11_RVALUE_REFERENCES
##  ^
## <command-line>:0:0: note: this is the location of the previous definition
##
## SAMPLING FOR MODEL 'response ~ dordlogit(phi, cutpoints)' NOW (CHAIN 1).
##
## Gradient evaluation took 0.008 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 80 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 316.185 seconds (Warm-up)
##                311.812 seconds (Sampling)
##                627.997 seconds (Total)
##
##
## SAMPLING FOR MODEL 'response ~ dordlogit(phi, cutpoints)' NOW (CHAIN 1).
##
## Gradient evaluation took 0.005 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 50 seconds.
## Adjust your expectations accordingly!
##
##
## WARNING: No variance estimation is
##          performed for num_warmup < 20
```

```
##
## Iteration: 1 / 1 [100%]  (Sampling)
##
##  Elapsed Time: 0 seconds (Warm-up)
##                0.012 seconds (Sampling)
##                0.012 seconds (Total)

## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems

## Computing WAIC

## Constructing posterior predictions

## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

To determine whether or not there is evidence that we should include varying intercepts on *story*, we should compare the model we just made to the two we made in the previous question.

```
compare(lmod12h2_1, lmod12h2_2, lmod12h3)
```

```
##               WAIC pWAIC  dWAIC weight     SE    dSE
## lmod12h3   30688.9 364.8    0.0      1 179.67     NA
## lmod12h2_2 31340.8 356.2  651.9      0 177.70  48.35
## lmod12h2_1 37089.8   9.0 6400.9      0  76.21 175.02
```

Our model which includes varying intercepts on the *story* variable takes all of the Akaike weight, so there is evidence that we should include this in our model.