Assignment-5

Cheng

5/14/2019

# Question1

## (a) Fit the model using rstan. Is the quadratic approximation okay?

# load the data  
library(MASS)  
data(eagles)  
dd <- eagles  
str(dd)

## 'data.frame': 8 obs. of 5 variables:  
## $ y: int 17 29 17 20 1 15 0 1  
## $ n: int 24 29 27 20 12 16 28 4  
## $ P: Factor w/ 2 levels "L","S": 1 1 1 1 2 2 2 2  
## $ A: Factor w/ 2 levels "A","I": 1 1 2 2 1 1 2 2  
## $ V: Factor w/ 2 levels "L","S": 1 2 1 2 1 2 1 2

# transform data  
dd$pirate <- ifelse(dd$P == "L", 1, 0)  
dd$adult <- ifelse(dd$A == "A", 1, 0)  
dd$victim <- ifelse(dd$V == "L", 1, 0)

Train the model using Rstan

# Rstan setup  
m1='  
 data {  
 int N;  
 int<lower=0, upper=1> pirate[N];  
 int<lower=0, upper=1> adult[N];  
 int<lower=0, upper=1> victim[N];  
 int N\_y[N]; // number of successful applicants  
 int N\_n[N]; // total number of attempts  
 }  
  
 parameters {  
 real a;  
 real bp;  
 real bv;  
 real ba;  
 }  
  
 model {  
 vector[N] p;  
 a ~ normal(0,10);  
 bp ~ normal(0,5);  
 bv ~ normal(0,5);  
 ba ~ normal(0,5);  
   
 for (i in 1:N){  
 p[i] = inv\_logit(a + bp \* pirate[i] + ba \* adult[i] + bv \* victim[i]);  
 }  
  
 N\_y ~ binomial(N\_n, p);  
 }  
  
 generated quantities {  
 vector[N] log\_lik;  
 {  
 vector[N] p\_y;  
 for(n in 1:N) {  
 p\_y[n] = a + bp \* pirate[n] + ba \* adult[n] + bv \* victim[n];  
 log\_lik[n] = binomial\_logit\_lpmf(N\_y[n] | N\_n[n], p\_y[n]);  
 }  
 }  
 }  
'  
  
dat <- list(  
 N = NROW(dd),  
 N\_n = dd$n,  
 N\_y = dd$y,  
 pirate = dd$pirate,  
 adult = dd$adult,  
 victim = dd$victim  
)  
  
fit1<- stan(model\_code = m1,   
 data = dat,   
 iter = 1000,   
 chains = 2,   
 cores = 2)

Train the model using map() to see how quadratic approximation works

m1\_qr <- map(  
 alist(  
 y ~ dbinom( n, p),  
 logit(p) <- a + bp\*pirate + ba\*adult + bv\*victim,  
 a ~ dnorm(0, 10),  
 c(bp, ba, bv) ~ dnorm(0, 5)),  
 data=dd  
)

Compare the result of MCMC and quadratic approximation

# Rstan using MCMC  
print(fit1)

## Inference for Stan model: 28bbc314a66057d3ca610fd6cd423573.  
## 2 chains, each with iter=1000; warmup=500; thin=1;   
## post-warmup draws per chain=500, total post-warmup draws=1000.  
##   
## mean se\_mean sd 2.5% 25% 50% 75% 97.5% n\_eff  
## a 0.65 0.03 0.73 -0.74 0.16 0.62 1.13 2.15 604  
## bp 4.64 0.06 0.98 2.99 3.97 4.53 5.26 6.91 313  
## bv -5.05 0.06 1.06 -7.29 -5.67 -5.01 -4.31 -3.20 285  
## ba 1.14 0.02 0.56 0.10 0.76 1.12 1.54 2.25 625  
## log\_lik[1] -2.75 0.04 1.18 -5.83 -3.26 -2.34 -1.88 -1.73 749  
## log\_lik[2] -0.08 0.00 0.08 -0.32 -0.10 -0.05 -0.02 0.00 398  
## log\_lik[3] -2.56 0.03 0.95 -5.31 -2.81 -2.20 -1.93 -1.85 909  
## log\_lik[4] -0.17 0.01 0.20 -0.73 -0.23 -0.10 -0.05 -0.01 402  
## log\_lik[5] -1.44 0.03 0.56 -3.07 -1.65 -1.24 -1.04 -0.96 295  
## log\_lik[6] -1.82 0.03 0.96 -4.37 -2.18 -1.51 -1.14 -0.97 848  
## log\_lik[7] -0.52 0.03 0.49 -1.81 -0.68 -0.38 -0.18 -0.04 372  
## log\_lik[8] -2.47 0.05 1.21 -5.51 -3.12 -2.20 -1.56 -0.92 576  
## lp\_\_ -48.18 0.07 1.43 -51.72 -48.92 -47.92 -47.11 -46.31 397  
## Rhat  
## a 1  
## bp 1  
## bv 1  
## ba 1  
## log\_lik[1] 1  
## log\_lik[2] 1  
## log\_lik[3] 1  
## log\_lik[4] 1  
## log\_lik[5] 1  
## log\_lik[6] 1  
## log\_lik[7] 1  
## log\_lik[8] 1  
## lp\_\_ 1  
##   
## Samples were drawn using NUTS(diag\_e) at Wed May 15 00:45:12 2019.  
## For each parameter, n\_eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor on split chains (at   
## convergence, Rhat=1).

# quadratic approximation  
precis(m1\_qr)

## Mean StdDev 5.5% 94.5%  
## a 0.59 0.66 -0.47 1.65  
## bp 4.24 0.90 2.81 5.67  
## ba 1.08 0.53 0.23 1.93  
## bv -4.59 0.96 -6.13 -3.06

# "select()" i'm not sure why rmd would have error, it's ok when I run R file.... so i would like to skip this part and intepret the result from previors result  
# Comparing estimates  
# coef <-  
# select(tidy(fit1)[1:4,], term, estimate) %>%  
# rename(., MCMC = estimate) %>%  
# mutate(MCMC = round(MCMC, digits = 2)) %>%  
# mutate(quadratic = c(0.59, 4.24, -4.59, 1.08))  
#   
# print(coef)

As we can see from the table, there is a slight difference of estimates when using MCMC(rstan) and quadratic approximation(map), especially for the parameter bv and bp.

Following I would use the output from Rstan to intepret the results.

## (b) Now interpret the estimates.

post1 <- as.data.frame(fit1)  
print(mean(logistic(post1$a)))

## [1] 0.6413459

The intercept is at 0.69, meaning that if all other predictor variables are 0, then the probability of a successful attempt is 0.6433955

print(mean(logistic(post1$a + post1$bp)))

## [1] 0.9914473

print(mean(logistic(post1$a + post1$ba)))

## [1] 0.8430078

print(mean(logistic(post1$a + post1$bv)))

## [1] 0.01820828

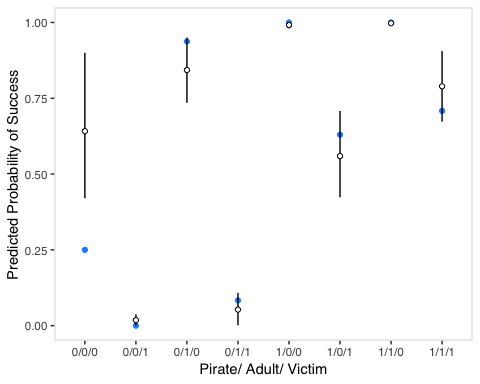
If the pirating eagle is large, then this probability increases to almost 100% If the pirating eagle is an adult, the probability of a successful attempt is quite high with a mean of about 84%. If the victim eagle is large, then it also doesn’t improve the chances much when the pirating eagle is an adult. While higher than before, it is still mostly below 10%.

### (1)

# create label  
lab <- c()  
for(i in c(1:nrow(dd))){  
 tmp <- c(paste(dd$pirate[i], dd$adult[i], dd$victim[i], sep="/"))  
 lab <- c(lab, tmp)  
}  
  
dd <-  
 dd %>%  
 mutate(case = factor(1:nrow(dd))) %>%  
 mutate(label = lab)

plot the Predicted Probability of Success

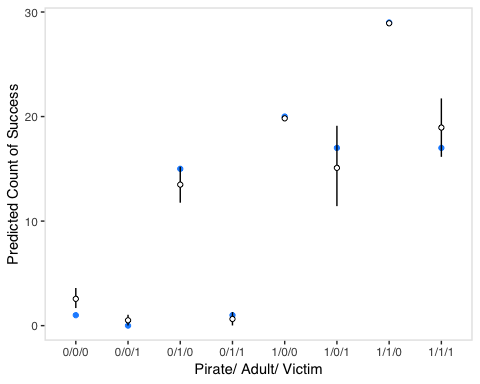
post1 <- as.data.frame(fit1)  
f\_mu <- function(p, v, a) plogis(post1$a + post1$bp \* p + post1$bv \* v + post1$ba \* a)  
p\_y\_hat <- mapply(f\_mu, p = (dd$pirate == 1), v = (dd$victim == 1), a = (dd$adult == 1) )  
  
# get expectation and 89% intervals of the expectation  
dd <-   
 dd %>%  
 mutate(p\_y = y / n,  
 p\_hat\_mean = colMeans(p\_y\_hat),  
 p\_hat\_hpdi\_l = apply(p\_y\_hat, 2, rethinking::HPDI)[1,], # 2: by row  
 p\_hat\_hpdi\_h = apply(p\_y\_hat, 2, rethinking::HPDI)[2,])  
  
# plot   
ggplot(dd) +   
 geom\_point(aes(x = label, y = p\_y), color = 'dodgerblue') +   
 geom\_segment(aes(x = label, xend = label, y = p\_hat\_hpdi\_l, yend = p\_hat\_hpdi\_h)) +   
 geom\_point(aes(x = label, y = p\_hat\_mean), shape = 21, fill = 'white') +   
 scale\_y\_continuous(limits = c(0, 1)) +  
 theme(panel.border = element\_rect(colour = "gray90", fill=NA, size=1),  
 panel.spacing.x = unit(-0.5, "mm"),  
 panel.spacing.y = unit(2, "lines")) +   
 labs(x = 'Pirate/ Adult/ Victim', y = 'Predicted Probability of Success')

 Blue filled points show the observed proportions whereas the estimates probability and 89% percent interval are shown in black and open points. Except for the case 0/0/0 (small, immature pirate and small victim), all observed proportions of success are within the 89% interval for the predicted probabilities.

### (2)

plot the Predicted Success Count

f\_mu <- function(n, p, v, a) n \* plogis(post1$a + post1$bp \* p + post1$bv \* v + post1$ba \* a)  
y\_hat <- mapply(f\_mu, n = dd$n, p = (dd$pirate == 1), v = (dd$victim == 1), a = (dd$adult == 1) )  
  
# how to choose the size?  
# f\_mu <- function(p, v, a) rbinom(n = 1e3, size = 20, prob = plogis(post1$a + post1$bp \* p + post1$bv \* v + post1$ba \* a ))  
# y\_hat <- mapply(f\_mu, p = (dd$pirate == 1), v = (dd$victim == 1), a = (dd$adult == 1))  
  
dd <-   
 dd %>%  
 mutate(y\_hat\_mean = colMeans(y\_hat),  
 y\_hat\_hpdi\_l = apply(y\_hat, 2, rethinking::HPDI)[1,], # 2: by row  
 y\_hat\_hpdi\_h = apply(y\_hat, 2, rethinking::HPDI)[2,])  
  
# plot  
ggplot(dd) +   
 geom\_point(aes(x = label, y = y), color = 'dodgerblue') +   
 geom\_segment(aes(x = label, xend = label, y = y\_hat\_hpdi\_l, yend = y\_hat\_hpdi\_h)) +   
 geom\_point(aes(x = label, y = y\_hat\_mean), shape = 21, fill = 'white') +   
 theme(panel.border = element\_rect(colour = "gray90", fill=NA, size=1),  
 panel.spacing.x = unit(-0.5, "mm"),  
 panel.spacing.y = unit(2, "lines")) +   
 labs(x = 'Pirate/ Adult/ Victim', y = 'Predicted Count of Success')

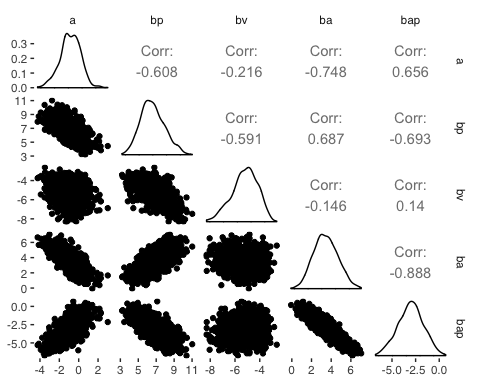
 Even though the predicted probability interval for the base case 0/0/0 did not contain the observed proportions of success, the observed number of successful attempts is contained in its 89% interval for the predicted success count.

## (c) Now try to improve the model. Consider an interaction between the pirate’s size and age (immature or adult). Compare this model to the previous one, using WAIC to interpret.

# Rstan setup: add interaction term  
m2='  
 data {  
 int N;  
 int<lower=0, upper=1> pirate[N];  
 int<lower=0, upper=1> adult[N];  
 int<lower=0, upper=1> victim[N];  
 int N\_y[N]; // number of successful applicants  
 int N\_n[N]; // total number of attempts  
 }  
  
 parameters {  
 real a;  
 real bp;  
 real bv;  
 real ba;  
 real bap;  
 }  
  
 model {  
 vector[N] p;  
 a ~ normal(0,10);  
 bp ~ normal(0,5);  
 bv ~ normal(0,5);  
 ba ~ normal(0,5);  
 bap ~ normal(0,5);  
   
 for (i in 1:N){  
 p[i] = inv\_logit(a + bp \* pirate[i] + ba \* adult[i] + bv \* victim[i] + bap \* adult[i] \* pirate[i]);  
 }  
  
 N\_y ~ binomial(N\_n, p);  
 }  
  
 generated quantities {  
 vector[N] log\_lik;  
 {  
 vector[N] p\_y;  
 for(n in 1:N) {  
 p\_y[n] = a + bp \* pirate[n] + ba \* adult[n] + bv \* victim[n] + bap \* adult[n] \* pirate[n];  
 log\_lik[n] = binomial\_logit\_lpmf(N\_y[n] | N\_n[n], p\_y[n]);  
 }  
 }  
 }  
'  
  
dat <- list(  
 N = NROW(dd),  
 N\_n = dd$n,  
 N\_y = dd$y,  
 pirate = dd$pirate,  
 adult = dd$adult,  
 victim = dd$victim  
)  
  
fit2<- stan(model\_code = m2,   
 data = dat,   
 iter = 1000,   
 chains = 2,   
 cores = 2)  
  
m3='  
data {  
int N;  
int N\_y[N]; // number of successful applicants  
int N\_n[N]; // total number of attempts  
}  
  
parameters {  
real a;  
  
}  
  
model {  
vector[N] p;  
a ~ normal(0,10);  
  
  
for (i in 1:N){  
p[i] = inv\_logit(a);  
}  
  
N\_y ~ binomial(N\_n, p);  
}  
  
generated quantities {  
vector[N] log\_lik;  
{  
 vector[N] p\_y;  
 for(n in 1:N) {  
 p\_y[n] = a ;  
 log\_lik[n] = binomial\_logit\_lpmf(N\_y[n] | N\_n[n], p\_y[n]);  
 }  
}  
}  
'  
  
dat <- list(  
 N = NROW(dd),  
 N\_n = dd$n,  
 N\_y = dd$y  
)  
  
fit3<- stan(model\_code = m3,   
 data = dat,   
 iter = 1000,   
 chains = 2,   
 cores = 2)

### Review correlations among posterior of parameters using improved model

#library(GGally)   
post2 <- as.data.frame(fit2)  
post2[,1:5] %>%  
 ggpairs() + theme\_tufte(base\_family = 'sans')



As before, we have some correlations between the coefficients: bap and ba, as well as bap and bp are negatively correlated. bp and baare now positively correlated and bp and bv are again negatively correlated but slightly less than in the model without interaction.

### Comparing WAIC

# initial model  
log\_lik\_m1 <- extract\_log\_lik(fit1, merge\_chains = FALSE)  
r\_eff <- relative\_eff(exp(log\_lik\_m1))  
loo\_m1 <- loo(log\_lik\_m1, r\_eff = r\_eff, cores = 2)  
waic1 <- loo(log\_lik\_m1)  
  
# include interaction term  
log\_lik\_m2 <- extract\_log\_lik(fit2, merge\_chains = FALSE)  
r\_eff <- relative\_eff(exp(log\_lik\_m2))  
loo\_m2 <- loo(log\_lik\_m2, r\_eff = r\_eff, cores = 2)  
waic2 <- loo(log\_lik\_m2)  
  
# only include intercept  
log\_lik\_m3 <- extract\_log\_lik(fit3, merge\_chains = FALSE)  
r\_eff <- relative\_eff(exp(log\_lik\_m3))  
loo\_m2 <- loo(log\_lik\_m3, r\_eff = r\_eff, cores = 2)  
waic3 <- loo(log\_lik\_m3)  
  
loo::compare(waic1, waic2, waic3)

## elpd\_diff se\_diff elpd\_loo p\_loo looic  
## waic2 0.0 0.0 -11.5 3.3 23.1  
## waic1 -5.4 1.7 -16.9 6.8 33.8  
## waic3 -67.4 31.3 -78.9 14.2 157.8

I compared 3 models: initial model, the model with interaction term and the model only included intercep. As we can see from the table, the model with interaction term got the best performance in this case.