

ST 540 Midterm exam

Ziyang Jiang

April 15, 2020

1. Introduction

The data consists of marathon performance data for men and women over the period 2015 to 2019. At first, we combine men and women data sets and delete observations that contain missing value. All following analysis will be based on this data set. The response variable Y is the marathon time in minutes. The covariates are two indicator variables $Vaporfly$ and $Gender$, more specifically, $Vaporfly = 1$ if Vaporfly shoes worn and $Vaporfly = 0$ if Vaporfly shoes not worn, $Gender=0$ for men and $Gender=1$ for women. Besides, we assign each runner a label between 1 and the number of unique runners J ($J = 578$), and assign each course a label between 1 and the number of unique courses K ($K = 95$). Here the unique course is determined by *year* and *marathon* together. The objective is to select a best fitting model and then determine the magnitude of the improvement due to wearing Vaporfly shoes. At the end of this report, we will test whether the improvement varies across gender, runner or course.

2. Methods

This report begins by making the assumption that the response variable of marathon time in minutes Y is approximately normally distributed. For this marathon data, the fixed effects model $Y_i = \beta_0 + \beta_1 Vaporfly_i + \beta_2 Gender_i + \varepsilon_i$ is inappropriate because of the fact that there are multiple performances for each runner and course. Moreover, some runners will run faster and some slower than average; similarly, some courses will be finished faster than average, and some slower. Therefore, this report will be based on linear mixed models.

In the first linear mixed model (**Model 1**), we take by-runner and by-course variability into account by adding random intercepts u_{0j} and w_{0k} , which adjust β_0 for runner j and course k . Each

runner and course has corresponding adjustment u_{0j} and w_{0k} , respectively. Defining model 1 in mathematical form as:

Likelihood:

$$\mu_i = \beta_0 + u_{0[\text{runner}[i]]} + w_{0[\text{course}[i]]} + \beta_1 \text{Vaporfly}_i + \beta_2 \text{Gender}_i$$

$$Y_i \sim N(\mu_i, \sigma_e^2) \text{ for } i \in \{1, \dots, n\}$$

$$\text{Priors: } u_{0j} \sim N(0, \sigma_u^2) \text{ for } j \in \{1, \dots, J\} \quad w_{0k} \sim N(0, \sigma_w^2) \text{ for } k \in \{1, \dots, K\}$$

$$\sigma_e^2, \sigma_u^2, \sigma_w^2 \sim \text{InvGamma}(0.1, 0.1) \quad \beta_0, \beta_1, \beta_2 \sim N(0, 10^2)$$

Consider that faster runners may show greater improvement when they wear Vaporfly shoes compared to not wear, or in course finished faster the improvement may be greater. The opposite situation may also hold. Therefore, in order to let the size of the effect for the predictor *Vaporfly* vary by runner and by course in **Model 2**, adjust β_1 by adding by-runner random slope u_{1j} and by-course random slopes w_{1k} based on Model 1. Defining model 2 in mathematical form as:

Likelihood:

$$\mu_{ijk} = \beta_0 + u_{0[\text{runner}[i]]} + w_{0[\text{course}[i]]} + (\beta_1 + u_{1[\text{runner}[i]]} + w_{1[\text{course}[i]]}) \text{Vaporfly}_i + \beta_2 \text{Gender}_i$$

$$Y_i \sim N(\mu_i, \sigma_e^2)$$

Prior:

$$(u_{0j}, u_{1j}) \sim N((0,0), \Sigma_u) \text{ for } j \in \{1, \dots, J\} \quad (w_{0k}, w_{1k}) \sim N((0,0), \Sigma_w) \text{ for } k \in \{1, \dots, K\}$$

$$\sigma_e^2, \sigma_u^2, \sigma_{u1}^2, \sigma_w^2, \sigma_{w1}^2 \sim \text{InvGamma}(0.1, 0.1) \quad \rho_u, \rho_w \sim U(-1, 1) \quad \beta_0, \beta_1, \beta_2 \sim N(0, 10^2)$$

In order to let the size of the effect for the predictor *Vaporfly* just vary by runner in **Model 3**, I just adjust β_1 by adding by-runner random slope u_{1j} based on Model 1. Defining model 3 in mathematical form as:

Likelihood:

$$\mu_{ijk} = \beta_0 + u_{0[\text{runner}[i]]} + (\beta_1 + u_{1[\text{runner}[i]]})\text{Vaporfly}_i + \beta_2\text{Gender}_i$$

$$Y_i \sim N(\mu_i, \sigma_e^2)$$

$$\text{Prior: } (u_{0j}, u_{1j}) \sim N((0,0), \Sigma_u) \text{ for } j \in \{1, \dots, J\} \quad \rho_u \sim U(-1,1)$$

$$\sigma_e^2, \sigma_{u0}^2, \sigma_{u1}^2 \sim \text{InvGamma}(0.1, 0.1) \quad \beta_0, \beta_1, \beta_2 \sim N(0, 10^2)$$

Similarly, in order to let the size of the effect for the predictor *Vaporfly* vary just by course in

Model 4, I just adjust β_1 by adding by-course random slope w_{1k} based on Model 1. Defining

model 4 in mathematical form as:

Likelihood:

$$\mu_{ijk} = \beta_0 + w_{0[\text{course}[i]]} + (\beta_1 + w_{1[\text{course}[i]]})\text{Vaporfly}_i + \beta_2\text{Gender}_i$$

$$Y_i \sim N(\mu_i, \sigma_e^2)$$

$$\text{Prior: } (w_{0k}, w_{1k}) \sim N((0,0), \Sigma_w) \text{ for } k \in \{1, \dots, K\} \quad \rho_w \sim U(-1,1)$$

$$\sigma_e^2, \sigma_{w0}^2, \sigma_{w1}^2 \sim \text{InvGamma}(0.1, 0.1) \quad \beta_0, \beta_1, \beta_2 \sim N(0, 10^2)$$

3. Computation

Four models were fitted in JAGS with same experimental setup. Two chains were run with the first 50,000 iterations discarded as burn-in and 100,000 samples for each chain were created. Besides, thinning set to 10.

4. Model comparisons

Different models were compared by using DIC, WAIC and convergence diagnostics. For Model 1, effective sample sizes (>1000) and Gelman and Rubin's convergence diagnostics (<1.1) for all parameters indicate good convergence. Besides, Model 1 has smallest DIC (=10009) and WAIC (=9996.721). For Model 2, effective sample sizes for ρ_u (≈ 337) and σ_{u1} (≈ 184) were smaller than 1,000 and Gelman and Rubin's convergence diagnostics were less

than 1.1 for all parameters, which indicate the convergence is not very ideal. And DIC (=10010) and WAIC (=10001.352) is greater than Model 1. For Model 3, effective sample sizes for ρ_u (≈ 110) and σ_{u1} (≈ 101) were smaller than 1,000 and Gelman and Rubin's convergence diagnostic for σ_{u1} was larger than 1.1, which indicate the convergence is unsatisfactory. And DIC (=10522) and WAIC(=10519.06) for Model 3 is large. For Model 4, effective sample sizes for ρ_w (≈ 866) and σ_{w1} (≈ 886) were smaller than 1,000 and Gelman and Rubin's convergence diagnostics were less than 1.1 for all parameters, which indicate the convergence is not good. And Model 4 has largest DIC(=10986) and WAIC(=10984.46).

5. Results

In order to determine the magnitude of the improvement, we select model 1 and focus on the posterior distributions of the parameter of interest β_1 . Approximately 95% of the posterior density of β_1 lies between -4.258 and -2.204, and thus we estimate that the marathon time will improve 2.204 to 4.258 minutes due to wearing Vaporfly shoes. Approximately 95% of the posterior density of the β_2 lies between 19.030 and 21.097 and this credible don't contain zero. And thus this is a weak evidence that marathon time will vary by gender. In model 1 gender will influence the intercept, and then we think gender will influence the size of Vaporfly effect.

In order to test whether the improvement varies across runner and/or course, we can use model 2. Since Gelman and Rubin's convergence diagnostics were less than 1.1 for all parameters, here we say Model 2 shows convergence according to this critical rule. Approximately 95% of the posterior density of ρ_u is lies between -0.9828 and 0.02096, most of them are negative. And approximately 95% of the posterior density of ρ_w is lies between -0.9738 and 0.5747, most of them are negative. There is only weak evidence that runners who can run faster than average exhibit smaller improvement than runners who run slower. Similarly, there is only weak evidence that in course finished faster the improvement is smaller than in courses finished slower.

R Code

```
rm(list=ls())

#Load the data

men<-read.csv("men_sampled_shoe.csv",header=T)

women<-read.csv("women_sampled_shoe.csv",header=T)

#delete obs with missing value

men<-na.omit(men)

women<-na.omit(women)

#combine men and women data

data<-rbind(men,women)

#response variable

Y<-data$time_minutes

#runner

rn<-unique(data$match_name)

runner<-rep(NA,length(Y))

for(R in 1:578){
  runner[data$match_name==rn[R]]<-R
}

#gender

gender<-c(rep(0,nrow(men)),rep(1,nrow(women)))

#course

cs<-unique(data[c("marathon","year")])

course<-rep(NA,length(Y))

for(C in 1:95){
  course[paste(data$marathon,data$year)==paste(cs[C,1],cs[C,2])]<-C
}

#vaporfly(0/1)

vaporfly<-as.numeric(data$vaporfly)
```

```
n<-length(Y) #number of performances
```

```
J<-578
```

```
K<-95
```

```
set.seed(123)
```

```
library(coda)
```

```
library(rjags)
```

```
###Model 1
```

```
model1_string<-textConnection("model{
```

```
  #likelihood
```

```
  for(i in 1:n){
```

```
    Y[i]~dnorm(mu[i],tau_e)
```

```
    mu[i]<-beta[1]+u[runner[i]]+w[course[i]]+beta[2]*vaporfly[i]+beta[3]*gender[i]}
```

```
  for(i in 1:J){
```

```
    u[i] ~ dnorm (0,tau_u)}
```

```
  for(i in 1:K){
```

```
    w[i] ~ dnorm (0,tau_w)}
```

```
  #priors
```

```
  for(i in 1:3){
```

```
    beta[i]~dnorm(0,0.01)
```

```
  }
```

```

tau_e~dgamma(0.1,0.1)
sigma_e <- 1/sqrt(tau_e)

tau_u ~ dgamma(0.1,0.1)
sigma_u <- 1/sqrt(tau_u)

tau_w ~ dgamma(0.1,0.1)
sigma_w <- 1/sqrt(tau_w)

#WAIC calculations
for(i in 1:n){
  like[i]<-dnorm(Y[i],mu[i],tau_e)
}
})

#Load the model
iters<-100000
burn<-50000
dat<-list(Y=Y,n=n,runner=runner,course=course,gender=gender,vaporfly=vaporfly,J=J,K=K)
init<-list(beta=rnorm(3),tau_e=10,tau_u=10,tau_w=10)
model1<-jags.model(model1_string,n.chains=2,inits=init,data=dat,quiet=TRUE)

#Generate samples
update(model1,burn,progress.bar="none")

```

```

samp1<-
coda.samples(model1,variable.names=c("beta","sigma_e","sigma_u","sigma_w"),n.iter=iters,th
in=10,progress.bar="none")

#Convergence diagnostics
plot(samp1) #graphical diagnostics for convergence
effectiveSize(samp1) # ESS over 1000 indicates convergence
gelman.diag(samp1) # R less than 1.1 indicates convergence
#all parameters show convergence

#summarize the output
summary(samp1)

# Compute DIC
dic1<-dic.samples(model1,n.iter=iters,progress.bar="none")
dic1

#Compute WAIC
waic1 <- coda.samples(model1,
  variable.names=c("like"),
  n.iter=iters, progress.bar="none")
like1 <- waic1[[1]]
fbar1 <- colMeans(like1)
P1 <- sum(apply(log(like1),2,var))
WAIC1 <- -2*sum(log(fbar1))+2*P1
WAIC1
P1

```



```
###Model 2
```

```
zero<-c(0,0)
```

```
model2_string<-textConnection("model{
```

```
  #likelihood
```

```
  for(i in 1:n){
```

```
    mu[i]<-beta[1]+u[runner[i],1]+w[course[i],1]+(beta[2]+u[runner[i],2]+w[course[i],2])*vaporfly[i]  
+beta[3]*gender[i]
```

```
    Y[i] ~ dnorm(mu[i],tau_e)
```

```
  }
```

```
#Constructing the covariance matrix and the corresponding precision matrix.
```

```
Sigma_u[1,1] <- sigma_u[1] * sigma_u[1];
```

```
Sigma_u[1,2] <- sigma_u[1] * sigma_u[2] * rho_u;
```

```
Sigma_u[2,1] <- sigma_u[2] * sigma_u[1] * rho_u;
```

```
Sigma_u[2,2] <- sigma_u[2] * sigma_u[2];
```

```
Prec_u[1:2,1:2]<-inverse(Sigma_u[,,])
```

```
Sigma_w[1,1] <- sigma_w[1] * sigma_w[1];
```

```
Sigma_w[1,2] <- sigma_w[1] * sigma_w[2] * rho_w;
```

```
Sigma_w[2,1] <- sigma_w[2] * sigma_w[1] * rho_w;
```

```
Sigma_w[2,2] <- sigma_w[2] * sigma_w[2];
```

```
Prec_w[1:2,1:2]<-inverse(Sigma_w[,,])
```

```
for(i in 1:J){
```

```
  u[i,1:2] ~ dmnorm (zero,Prec_u[,,])
```

```

for(i in 1:K){
  w[i,1:2] ~ dnorm(zero, Prec_w[,i])

#prior
for(i in 1:3){ beta[i] ~ dnorm(0,0.01)}

tau_e ~ dgamma(0.1,0.1)
sigma_e <- 1/sqrt(tau_e)

for(i in 1:2){
  tau_u[i] ~ dgamma(0.1,0.1)
  sigma_u[i] <- 1/sqrt(tau_u[i])
}

for(i in 1:2){
  tau_w[i] ~ dgamma(0.1,0.1)
  sigma_w[i] <- 1/sqrt(tau_w[i])
}

rho_u ~ dunif(-1,1)
rho_w ~ dunif(-1,1)

#WAIC calculations
for(i in 1:n){
  like[i] <- dnorm(Y[i], mu[i], tau_e)
}

```

```

})

#Load the model

iters<-100000

burn<-50000

dat<-

list(Y=Y,n=n,runner=runner,course=course,gender=gender,vaporfly=vaporfly,J=J,K=K,zero=zero
)

init<-

list(beta=rnorm(3),tau_e=10,tau_u=c(10,10),tau_w=c(10,10),rho_u=runif(1,-1,1),rho_w=runif(1,-1
,1))

model2<-jags.model(model2_string,n.chains=2,inits=init,data=dat,quiet=TRUE)

#Generate samples

update(model2,burn,progress.bar="none")

samp2<-

coda.samples(model2,variable.names=c("beta","sigma_e","sigma_u","sigma_w","rho_u","rho_
w"),n.iter=iters,thin=10,progress.bar="none")

#Convergence diagnostics

#plot(samp2) #graphical diagnostics for convergence

effectiveSize(samp2) # ESS over 1000 indicates convergence

gelman.diag(samp2) # R less than 1.1 indicates convergence

summary(samp2)

# Compute DIC

dic2<-dic.samples(model2,n.iter=iters,progress.bar="none")

```

dic2

Compute WAIC

```
waic2 <- coda.samples(model2,  
  variable.names=c("like"),  
  n.iter=iters, progress.bar="none")
```

```
like2 <- waic2[[1]]
```

```
fbar2 <- colMeans(like2)
```

```
P2 <- sum(apply(log(like2),2,var))
```

```
WAIC2 <- -2*sum(log(fbar2))+2*P2
```

WAIC2

P2

###Model 3

```
zero<-c(0,0)
```

```
model3_string<-textConnection("model{
```

```
  #likelihood
```

```
  for(i in 1:n){
```

```
    mu[i]<-beta[1]+u[runner[i],1]+(beta[2]+u[runner[i],2])*vaporfly[i]+beta[3]*gender[i]
```

```
    Y[i] ~ dnorm(mu[i],tau_e)
```

```
  }
```

```
  #Constructing the covariance matrix and the corresponding precision matrix.
```

```
  Sigma_u[1,1] <- sigma_u[1] * sigma_u[1];
```

```
  Sigma_u[1,2] <- sigma_u[1] * sigma_u[2] * rho_u;
```

```
  Sigma_u[2,1] <- sigma_u[2] * sigma_u[1] * rho_u;
```

```
  Sigma_u[2,2] <- sigma_u[2] * sigma_u[2];
```

```

Prec_u[1:2,1:2]<-inverse(Sigma_u[,,])

for(i in 1:J){
u[i,1:2] ~ dnorm (zero,Prec_u[,,])

#prior
for(i in 1:3){ beta[i] ~ dnorm(0,0.01)}

tau_e~dgamma(0.1,0.1)
sigma_e <- 1/sqrt(tau_e)

for(i in 1:2){
tau_u[i] ~ dgamma(0.1,0.1)
sigma_u[i] <- 1/sqrt(tau_u[i])
}

rho_u ~ dunif(-1,1)

#WAIC calculations
for(i in 1:n){
like[i]<-dnorm(Y[i],mu[i],tau_e)
}
})

```

```

#Load the model
iters<-100000
burn<-50000

```

```

dat<-list(Y=Y,n=n,runner=runner,gender=gender,vaporfly=vaporfly,J=J,zero=zero)
init<-list(beta=rnorm(3),tau_e=10,tau_u=c(10,10),rho_u=runif(1,-1,1))
model3<-jags.model(model3_string,n.chains=2,inits=init,data=dat,quiet=TRUE)

#Generate samples
update(model3, burn, progress.bar="none")
samp3<-
coda.samples(model3,variable.names=c("beta","sigma_e","sigma_u","rho_u"),n.iter=iters,thin=
10,progress.bar="none")

#Convergence diagnostics
plot(samp3) #graphical diagnostics for convergence
effectiveSize(samp3) # ESS over 1000 indicates convergence
gelman.diag(samp3) # R less than 1.1 indicates convergence

# Compute DIC
dic3<-dic.samples(model3,n.iter=iters,progress.bar="none")
dic3

# Compute WAIC
waic3 <- coda.samples(model3,
  variable.names=c("like"),
  n.iter=iters, progress.bar="none")
like3 <- waic3[[1]]
fbar3 <- colMeans(like3)
P3 <- sum(apply(log(like3),2,var))
WAIC3 <- -2*sum(log(fbar3))+2*P3

```

WAIC3

P3

```
###Model 4
```

```
zero<-c(0,0)
```

```
model4_string<-textConnection("model{
```

```
  #likelihood
```

```
  for(i in 1:n){
```

```
    mu[i]<-beta[1]+w[course[i],1]+(beta[2]+w[course[i],2])*vaporfly[i]+beta[3]*gender[i]
```

```
    Y[i] ~ dnorm(mu[i],tau_e)
```

```
  }
```

```
  #Constructing the covariance matrix and the corresponding precision matrix;
```

```
  Sigma_w[1,1] <- sigma_w[1] * sigma_w[1];
```

```
  Sigma_w[1,2] <- sigma_w[1] * sigma_w[2] * rho_w;
```

```
  Sigma_w[2,1] <- sigma_w[2] * sigma_w[1] * rho_w;
```

```
  Sigma_w[2,2] <- sigma_w[2] * sigma_w[2];
```

```
  Prec_w[1:2,1:2]<-inverse(Sigma_w[,,])
```

```
  for(i in 1:K){
```

```
    w[i,1:2] ~ dmnorm (zero,Prec_w[,,])
```

```
  #prior
```

```
  for(i in 1:3){ beta[i] ~ dnorm(0,0.01)}
```

```

tau_e~dgamma(0.1,0.1)
sigma_e <- 1/sqrt(tau_e)

for(i in 1:2){
  tau_w[i] ~ dgamma(0.1,0.1)
  sigma_w[i] <- 1/sqrt(tau_w[i])
}

rho_w ~ dunif(-1,1)

#WAIC calculations
for(i in 1:n){
  like[i]<-dnorm(Y[i],mu[i],tau_e)
}
})

#Load the model
iters<-100000
burn<-50000
dat<-list(Y=Y,n=n,course=course,gender=gender,vaporfly=vaporfly,K=K,zero=zero)
init<-list(beta=rnorm(3),tau_e=10,tau_w=c(10,10),rho_w=runif(1,-1,1))
model4<-jags.model(model4_string,n.chains=2,init=init,data=dat,quiet=TRUE)

#Generate samples
update(model4,burn,progress.bar="none")

```



```
samp4<-  
coda.samples(model4,variable.names=c("beta","sigma_e","sigma_w","rho_w"),n.iter=iters,thin  
=10,progress.bar="none")
```

```
#Convergence diagnostics
```

```
plot(samp4) #graphical diagnostics for convergence
```

```
effectiveSize(samp4) # ESS over 1000 indicates convergence
```

```
gelman.diag(samp4) # R less than 1.1 indicates convergence
```

```
# Compute DIC
```

```
dic4<-dic.samples(model4,n.iter=iters,progress.bar="none")
```

```
dic4
```

```
# Compute WAIC
```

```
waic4 <- coda.samples(model4,  
  variable.names=c("like"),  
  n.iter=iters, progress.bar="none")
```

```
like4 <- waic4[[1]]
```

```
fbar4 <- colMeans(like4)
```

```
P4 <- sum(apply(log(like4),2,var))
```

```
WAIC4 <- -2*sum(log(fbar4))+2*P4
```

```
WAIC4
```

```
P4
```