ベイズで失敗していたコード

#データの加工 重回帰分析した時にstepAICで選択された変数を使う

#そのままだとエラー出た(Stanの型は int か real しかないから)からダミー変数に変換する

#でも変換するMjobが4つの変数に分解されてしまうから、Stanコードでの説明変数の選択方法はどうすればいいのかな？

#とりあえず・・・・

#専業主婦を0にして、MjobをMjobhealth、Mjobother、Mjobservices、Mjobteacherの4つの変数に分解する、みたいにやった

library(caret)

library(ggplot2)

tmp <- dummyVars(~., data=x)

#dummyVarsの"~." ……このニョロニョロの後ろの "."（ピリオド) は

#全ての質的変数を対象とすることを意味する

x.re <- as.data.frame(predict(tmp, x))

a <- x.re[,3]

b <- x.re[,14:23]

c <- x.re[,32:37]

d <- x.re[,44:45]

e <- x.re[,51:52]

f <- x.re[,55]

g <- x.re[,57]

d1<-data.frame(a, b, c, d, e, g)

data1 <- list(N=nrow(d1), SexM=d1[,1], Mjobhealth=d1[,2], Mjobother=d1[,3], Mjobservices=d1[,4], Mjobteacher=d1[,5],

Fjobhealth=d1[,7], Fjobother=d1[,8], Fjobservices=d1[,9], Fjobteacher=d1[,10],

Studytime=d1[, 11], Failures=d1[,12], Schoolsupyes=d1[,14], Famsupyes=d1[,16],

Higheryes=d1[,18], Freetime=d1[,19], Goout=d1[,20], Health=d1[,21], Y=d1[,22])

library(rstan)

#計算の高速化

#rstan\_options(auto\_write = T)と指定した上でMCMCを実行すると、拡張子が「.rds」であるRDSファイルが作成される

#再度のコンパイルをしなくても済むようにRDSファイルに保存しておく

rstan\_options(auto\_write = T)

#options(mc.cores = parallel::detectCores())と指定すると計算を並列化するので実行速度が上がる

options(mc.cores = parallel::detectCores())

#Y入れるの忘れてずっとエラー出てた！気をつける！

#dataと思って列指定していた$d1だから、d1の形を確認して列名指定する

fit1 <-stan(file = 'model5-3.stan', data = data1, seed = 1234, control = list(max\_treedepth = 11))

fit1

#以下のコメント文は前回やった時に記述したもの

#計算を高速化するやつ入れていない時のもの

#The largest R-hat is 3.89, indicating chains have not mixed.⇨収束しなかった 計算に30分くらいかかった

#最大ツリー深度を超えたから最大許容ツリー深度を増やす

#警告文にIncrease max\_treedepth above 10⇨control = list(max\_treedepth = 11)でデフォルトの10から増やす

#The largest R-hat is 3.89⇨悪化しちゃった なんで??

#SAMPLING FOR MODEL 'model5-3' NOW (CHAIN 1).

#Chain 1:

# Chain 1: Gradient evaluation took 0.000524 seconds

#Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 5.24 seconds.

#Chain 1: Adjust your expectations accordingly!

# Chain 1:

# Chain 1:

# Chain 1: Iteration: 1 / 2000 [ 0%] (Warmup)

#の状態だったので強制終了した

#何かしらの改善をする！

#本の収束に関するところを読む

######中村先生にきく！！#######

#少し上手くいかないので収束診断をファイルへ出力してみる

save.image(file = "Portugal.RData")

library(rstan)

library(ggmcmc)

load("Portugal.RData")

write.table(data.frame(summary(fit1)$summary),

file = "fit1-summary.txt",

sep = "\t",

quote = F,

col.names = NA)

ggmcmc(ggs(fit1),

file = "fit1-ggmcmc.pdf",

plot = c("traceplot", "density", "running","autocorrelation"))

#Rhatがずば抜けて大きかった Fjob(5.49)、Studytime(5.49)、Higheryes(5.49)を除いて実行してみる

library(caret)

library(ggplot2)

tmp <- dummyVars(~., data=x)

#dummyVarsの"~." ……このニョロニョロの後ろの "."（ピリオド) は

#全ての質的変数を対象とすることを意味する

x.re <- as.data.frame(predict(tmp, x))

View(x.re)

d1.re<-data.frame(x.re[,4], x.re[,14:18], x.re[,35],x.re[,37], x.re[ ,53] ,x.re[,55], x.re[,57])

data1.re <- list(N=nrow(d1), SexM=d1[,1], Mjobhealth=d1[,2], Mjobother=d1[,3], Mjobservices=d1[,4], Mjobteacher=d1[,5],Failures=d1[,7], Schoolsupyes=d1[,], Famsupyes=d1[,16], Freetime=d1[,19], Goout=d1[,20], Health=d1[,21], G1=d1[,22])

library(rstan)

#計算の高速化

#rstan\_options(auto\_write = T)と指定した上でMCMCを実行すると、拡張子が「.rds」であるRDSファイルが作成される

#再度のコンパイルをしなくても済むようにRDSファイルに保存しておく

rstan\_options(auto\_write = T)

#options(mc.cores = parallel::detectCores())と指定すると計算を並列化するので実行速度が上がる

options(mc.cores = parallel::detectCores())

#Y入れるの忘れてずっとエラー出てた！気をつける！

#dataと思って列指定していた$d1だから、d1の形を確認して列名指定する

#control = list(max\_treedepth = 11)もとってデフォルトにする

fit2 <-stan(file = 'model5-3re.stan', data = data1.re, seed = 1234)

fit2

#多分母親の仕事、父親の仕事が問題(3以上のカテゴリカル変数で、扱いが正しいのか不安)

#専業主婦、専業主夫を0にして、MjobをMjobhealth、Mjobother、Mjobservices、Mjobteacherの4つの変数に分解してやった

#↑これがダメだったかもしれないから、とりあえず母親の仕事、父親の仕事を除いてやってみる

library(rstan)

data2 <- list(N=nrow(d1), SexM1=d1[,1], Studytime1=d1[, 11], Failures1=d1[,12], Schoolsupyes1=d1[,14], Famsupyes1=d1[,16],

Higheryes1=d1[,18], Freetime1=d1[,19], Goout1=d1[,20], Health1=d1[,21], Y1=d1[,22])

fit2 <-stan(file = 'model5-3re.stan', data = data2, seed = 1234, control = list(max\_treedepth = 21))

#計算時間23分

#警告メッセージ

#最大ツリー深度を超えた

#No-U-Turn-Sampler（Stanが使用するHMCのバリアント）の構成には、各反復中に評価する木の深さに上限を設定する必要がある

#（詳細については、Stanマニュアルのハミルトニアンモンテカルロサンプリングの章を参照してください）。）

#これは、最大深度パラメータによって制御されます。最大許容ツリー深度に達すると、実行時間が過度に長くなるのを避けるために、NUTSが途中で終了していることを示します。max\_treedepth

# 非ベイズで重回分析した時にp値が小さいランキング1~3位の変数に絞る

library(rstan)

data3 <- list(N=nrow(d1), Studytime2=d1[, 11], Failures2=d1[,12], Schoolsupyes2=d1[,14], Y2=d1[,22])

fit3 <-stan(file = 'model5-3re1.stan', data = data3, seed = 1234)

fit3

# Rhatが全て1だから収束している

# 切片の事後平均が7.28、studytimeの事後平均が2.47 、failuresの事後平均が0.71 、schoolsupの事後平均が2.30

# 説明変数を3つに減らしたら成功したから、↑2つの重回がうまくいかなかったのは変数の多さが原因？

# どうやって説明変数選択すればいいのかな？本には重回帰の場合の例の説明変数が2つだけだったからなー

# 最尤推定と比較する

lm2<-lm(z$G1 ~ studytime + failures + schoolsup, data = z)

summary(lm2)

# 切片の係数が10.7964、studytimeの係数が0.4372 、failuresの係数が-1.4980、-2.1445 の事

####ベイズと非ベイズで単回帰の時は大差なかったが、重回帰(説明変数を3つ)では回帰係数に大きく差が出た###

####mcmcの設定を変えてみる###

library(rstan)

data1 <- list(N=nrow(d1), SexM=d1[,1], Mjobhealth=d1[,2], Mjobother=d1[,3], Mjobservices=d1[,4], Mjobteacher=d1[,5],

Fjobhealth=d1[,7], Fjobother=d1[,8], Fjobservices=d1[,9], Fjobteacher=d1[,10],

Studytime=d1[, 11], Failures=d1[,12], Schoolsupyes=d1[,14], Famsupyes=d1[,16],

Higheryes=d1[,18], Freetime=d1[,19], Goout=d1[,20], Health=d1[,21], Y=d1[,22])

stanmodel <- stan\_model(file = 'model5-3.stan')

fit4 <- sampling(

stanmodel,

data=data1,

control = list(max\_treedepth = 14),

chains=4, iter=6000, warmup=3000, thin=5

)

fit4

#The largest R-hat is 3.39, indicating chains have not mixed.

#Tail Effective Samples Size (ESS) is too low ⇨iter=4000に変更

#Running the chains for more iterations may help ⇨iter=4000に変更

#そしたら80分くらいでThe largest R-hat is 2.37⇨iter=7000に変更

#chains=4, iter=5000, warmup=2500, thin=5 ⇨The largest R-hat is 2.91

#なかなか収束しないのは統計モデルに問題があるのもしれない

#計算が終わらないことを中村先生に相談する前に

#より観測データへの当てはまりの良さそうな階層ベイズをやってみよう！

#階層ベイズ