ST440/540 Applied Bayesian Analysis
Lab activity for 4/1/2024

Final homework assignment is due this Friday
Abstract is due 4/12
I will send the exam later this week. It is due April 15

A. HOMEWORK AND CLASS PARTICIPATION SOLUTIONS

Let Y_t be the number of days with snow at RDU Airport in year t (below) and X_t=t be the year. Write a generalized linear model to test whether the rate of snowfall days is changing over time. Give the likelihood and prior and describe how you would carry out the test.

Since Y_t is a count, I would use a Poisson model. To allow the mean to change over time I would use X_t as a covariate in the log mean, so $Y_t \mid \lambda_t \sim \text{Poisson}(\lambda_t)$ with $\lambda_t = \exp(a + bX_t) > 0$. For uninformative prior I would use a,b~Normal(0,100). I would use MCMC to approximate the posterior of b and conclude the snowfall distribution is changing over time if the 95% interval excluded zero.

B. DISCUSSION QUESTIONS

(1) Let Y be the party affiliation of a voter and X be their annual income. Say Y is either R, D or I and X is continuous and positive. The goal is to build a model to predict party affiliation given income.

(a) Below are two modeling approaches based on logistic regression. Which do you prefer and why?

Model 1

Logit[Prob(Y=R)] = $a_1 + b_1 * X$ Logit[Prob(Y=D)] = $a_2 + b_2 * X$ Logit[Prob(Y=I)] = $a_3 + b_3 * X$

Model 2

Logit[Prob(Y=R)] = $a_1 + b_1 * X$ Logit[Prob(Y=D|Y≠ R)] = $a_2 + b_2 * X$

Model 1 doesn't make sense because the three probabilities don't necessarily sum to one.

(b) In the second model, what is the probability that a voter with income X=x is independent? Use notation Logit(Prob(Y=R)) = a_1+b_1X <=> Prob(Y=R) = expit(a_1+b_1X).

Prob(Y=I) = Prob(not R)* Prob(not D|not R) = [1-expit(a1+b1X)] [1-expit(a2+b2X)].

(c) In the second model, what is the interpretation of the parameter b_2?

Given that Y is either D or I, the log odds of D increase by b2 if X increases by 1.

(d) How would you modify this model if the covariate was X discrete with levels low, medium and high?

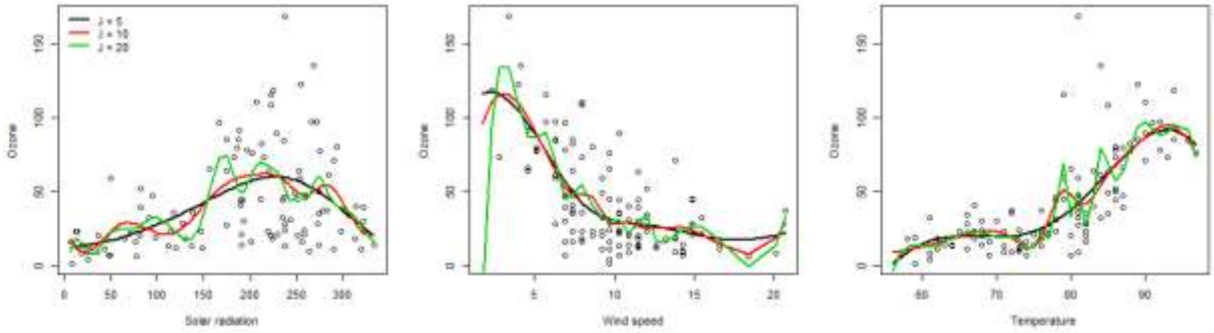
Add two dummy variables, X1 = 1 if medium and X1 = 0 otherwise and X2 = 1 if high and X2 = 0 otherwise,

Logit[Prob(Y=R)] = a1+ b1*X1 + c1*X2 Logit[Prob(Y=D|Y≠ R)] = a2+b2*X1+ c2*X2

(2) The plots below show the fit of a non-parametric regression model with

$$Y_i = a + \sum_{j=1}^{J} B_j(X_i)b_j + e_i$$

and flat priors for the regression coefficients a,b_1,…,b_J The three plots use the same response variable Y but different X variables. The code is on the final page



(a) Visually, which values of J look the best for each fit?

I'd say 5 or 10.

(b) How would you formally select J?

Cross validation, DIC or WAIC are all options.

(c) If the flat priors were replaced by normal priors b_j ~ Normal(0,v) with v ~ InvGamma(0.1,0.1), would you expect to need more or fewer basis functions? Why?

I'd expect we'd need more basis functions because the prior would prevent over-fitting so we could have more basis function and retain a stable fit.

(3) Consider the models

M1: Y ~ N(0, σ^2)

M2: Y|μ ~ N(μ, σ^2) and μ ~ N(0, cσ^2)

The Bayes factor comparing M2 and M1 is

$$BF = \frac{1}{\sqrt{1+c}} exp\left\{ - \frac{y^2}{2\sigma^2}\left(\frac{c}{c+1}\right)\right\}$$

(a) What happens to the Bayes factor as c -> infinity?

The prior becomes more uninformative and BF goes to zero, favoring the null model.

(b) What does this tell you about Bayes factors?

They are very sensitive to the prior.

(4) Take a minute to review the analysis of the Gambia data,

https://www4.stat.ncsu.edu/~bjreich/BSMdata/SSVS.html

Below is output from the SSVS model and Bayesian logistic regression with uninformative Gaussian priors for all parameters

```
SSVS model
       Inc_Prob       50%       5%    95%
Age         1.00      0.26     0.19   0.34
Netuse       1.00     -0.25    -0.34  -0.17
Treated      0.79     -0.13    -0.24   0.00
Green       1.00      0.29      0.21   0.37
PCH         0.56     -0.05    -0.19   0.00
```

```
Flat priors
                 Mean   SD     5%    95%
Age              0.27 0.05  0.18  0.37
Netuse          -0.25 0.05 -0.36 -0.15
Treated         -0.13 0.06 -0.25 -0.01
Green            0.29 0.05  0.19  0.39
PCH             -0.10 0.05 -0.20  0.01
```

(a) How do the results compare? Which model would you use?

The model fits are pretty similar, so I'd probably us flat priors because it's faster and easier to explain.

(b) We have now used two ways to determine if a covariates is "significant": (i) SSVS and inclusion probabilities>0.5 and (ii) a flat prior and seeing if zero is included in the posterior intervals. What are the pros and cons of these two approaches?

For models with only a few covariates I use posterior intervals, but if there are many covariates it's better to use SSVS.

(5) The data generated below has very strong collinearity.

(a) What do you anticipate the output of the SSVS model will be in tables below?

	Inc_Prob	**50%**	**5%**	**95%**
X1	-	-	-	-
X2	-	-	-	-
X3	-	-	-	-

Model	**Posterior probs**
NULL	-
X1	-
X2	-
X3	-
X1 + X2	-
X1 + X3	-
X2 + X3	-
X1 +X2 +X3	-

The values are

```
 Inc_Prob 50% 5% 95%
beta[1] 0.58 0.0 -5.23 1.55
beta[2] 0.65 0.4 -0.49 7.10
beta[3] 0.51 0.0 -0.67 1.51
```

Model probabilities:

```
Intercept + X1 + X2 Intercept + X2 Intercept + X1 + X2 + X3
0.201 0.173 0.144
Intercept + X2 + X3 Intercept + X3 Intercept + X1
0.132 0.118 0.117
Intercept + X1 + X3
0.114
```

(b) In real life, how would you handle this analysis?

Remove some covariates until the model has less collinearity.

Code

```
n  <- 100
p  <- 3
set.seed(919)
X1 <- rnorm(n)
X2 <- X1 + 0.01*rnorm(n)
X3 <- X2 + 0.01*rnorm(n)
X  <- cbind(X1,X2,X3)
Y  <- rnorm(n,X2,1)

> round(cor(X),4)
       X1     X2     X3
X1 1.0000 0.9999 0.9999
X2 0.9999 1.0000 0.9999
X3 0.9999 0.9999 1.0000

> summary(lm(Y~X))

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.06191    0.09623  -0.643    0.522
XX1         -8.13549   10.22983  -0.795    0.428
XX2         15.43476   13.65390   1.130    0.261
XX3         -6.13203    9.51324  -0.645    0.521

Residual standard error: 0.9493 on 96 degrees of freedom
Multiple R-squared:  0.5753,    Adjusted R-squared:  0.5621
F-statistic: 43.36 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
# SSVS model in JAGS
m <- textConnection("model{
    for(i in 1:n){
       Y[i]  ~ dnorm(mu[i],taue)
      mu[i] <- alpha + X[i,1]*beta[1] + X[i,2]*beta[2] + X[i,3]*beta[3]
    }
    for(j in 1:3){
        beta[j] <- gamma[j]*delta[j]
        gamma[j] ~ dbern(0.5)
        delta[j] ~ dnorm(0,taub)
    }
    alpha ~ dnorm(0,0.01)
    taub   ~ dgamma(0.1,0.1)
    taue   ~ dgamma(0.1,0.1)
}")

# Run JAGS
   library(rjags)
   data   <- list(Y=Y,X=X,n=n)
   burn   <- 10000
   iters  <- 50000
   chains <- 3
   model  <- jags.model(m,data = data, n.chains=chains,quiet=TRUE)
   update(model, burn, progress.bar="none")
   samps  <- coda.samples(model, variable.names=c("beta"),
                          thin=5, n.iter=iters, progress.bar="none")
   plot(samps)

# Summarize the posterior of beta
   beta    <- NULL
   for(l in 1:chains){
     beta <- rbind(beta,samps[[l]])
   }
   Inc_Prob <- apply(beta!=0,2,mean)
   Q        <- t(apply(beta,2,quantile,c(0.5,0.05,0.95)))
   out      <- cbind(Inc_Prob,Q)
   round(out,2)

# Compute model probabilities
   model <- "Intercept"
   names <- paste0("X",1:3)
   for(j in 1:3){
     model <- paste(model,ifelse(beta[,j]==0,"","+"))
     model <- paste(model,ifelse(beta[,j]==0,"",names[j]))
   }
   model_probs <- table(model)/length(model)
   model_probs <- sort(model_probs,dec=T)
   round(model_probs,3)

# Plot predicted versus fitted
   plot(X%*%colMeans(beta),Y)
```

Code for problem 2

```
library(splines)

data(airquality)
Ozone <- airquality[,1]
SR    <- airquality[,2]
Wind  <- airquality[,3]
Temp  <- airquality[,4]

par(mfrow=c(1,3))
for(i in 1:3){
 if(i==1){X <- SR;xlab <- "Solar radiation"}
 if(i==2){X <- Wind;xlab <- "Wind speed"}
 if(i==3){X <- Temp;xlab <- "Temperature"}

 Y    <- Ozone
 ylab <- "Ozone"

 ooo  <- order(X)
 Y    <- Y[ooo]
 X    <- X[ooo]
 plot(X,Y,xlab=xlab,ylab=ylab)
 m    <- c(5,10,20)
 for(j in 1:length(m)){
   B   <- bs(X,df=m[j])
   b   <- lm(Y ~ B)$coef
   lines(X,b[1] + B%*%b[-1],lwd=2,col=j)
  }
  if(i==1){
    legend("topleft",paste("J =",m),lwd=2,col=1:4,bty="n")
  }
}
```