Bernoulli regression model (Variable Selection)

Case study: Space shuttle Challenger disaster

Georgios P. Karagiannis @ MATH3341/4031 Bayesian statistics III/IV (practical implementation)

Back to README

```
rm(list=ls())
```

Aim

Students will become able to:

- produce Monte Carlo approximations of posterior quantities required for Bayesian analysis with the RJAGS R package
- implement Bayesian posterior analysis in R with RJAGS package

Students are not required to learn by heart any of the concepts discussed

---

Reading material

The material about RJAGS package is not examinable material, but it is provided for the interested student. It contains references that students can follow if they want to further explore the concepts introdced.

- Lecture notes:
  - the examples and exercises related to the Bernoulli model with conjugate prior
- Application (optional):
  - Dalal, S. R., Fowlkes, E. B., & Hoadley, B. (1989). Risk analysis of the space shuttle: Pre-Challenger prediction of failure. Journal of the American Statistical Association, 84(408), 945-957.
- References for *rjags*:
  - JAGS homepage
  - JAGS R CRAN Repository
  - JAGS Reference Manual
  - JAGS user manual
- Reference for *R*:
  - Cheat sheet with basic commands
- Reference of *rmarkdown* (optional):
  - R Markdown cheatsheet
  - R Markdown Reference Guide
  - knitr options
- Reference for *Latex* (optional):
  - Latex Cheat Sheet

New software

- R package `rjags` functions:

- jags.model{rjags}

- jags.samples{rjags}
- update{rjags}

Application: Challenger O-ring

On January 28, 1986, a routine launch was anticipated for the Challenger space shuttle. Seventy-three seconds into the flight, disaster happened: the shuttle broke apart, killing all seven crew members on board. Here is the video.

The Rogers Commission report on the space shuttle Challenger accident concluded that the accident was caused by a combustion gas leak through a joint in one of the booster rockets, which was sealed by a device called an O-ring. The Challenger accident was caused by gas leak through the 6 O-ring joints of the shuttle.

The commission further concluded that 0-rings do not seal properly at low temperatures.

Dalal, Fowlkes and Hoadley (1989) looked at the number of distressed O-rings (among the 6) for 23 previous shuttle flights, and the data-set is provided below. In the table below presents data from the 23 preaccident launches of the space shuttle is used to predict 0-ring performance under the Challenger launch conditions and relate it to the catastrophic failure of the shuttle. The the data-set is provided below, where in column *Defective.O.rings*, (1) stands for presence of at least one distressed O-ring, and (0) stands for absence of any distressed O-ring; while the rest columns are self explained.

```
# Load R package for printing
library(knitr)
library(kableExtra)
```

```
# load the data
#mydata <- read.csv("./challenger_data.csv")
mydata <- read.csv("https://raw.githubusercontent.com/georgios-stats/Bayesian_Statistics/master/Computer
# print data
## (that's a sophisticated command with fancy output, feel free to ignore it)
kable(mydata)%>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

On the night of January 27, 1986, the night before the space shuttle Challenger accident, there was a three-hour teleconference among people at Morton Thiokol, Marshall Space Flight Center, and Kennedy Space Center. The discussion focused on the forecast of a 31F temperature for launch time the next morning, and the effect of low temperature on 0-ring performance.

We are interested in finding if any (or both) of the variables *Damage.Incident* and *Temperature* can be characterised as discriminator for the occirance of a defective O-ring.

To answer the above we perform Bayesian analysis based on the observed data-set on the dates from *04/12/1981* to *01/12/1986*, and the variables *Damage.Incident* and *Temperature*. So ignore the variable *Leak.check.pressure*.

Model specification & posterior sampling

Let y_i denote the presence of a defective O-ring in the ith flight (0 for absence, and 1 for presence).

Regarding the statistical model, we assume that y_i can be modeled as observations generated independently from a Bernoulli distribution with with parameter p_i. Here, p_i denotes the relative frequency of defective O-rings at flight i. We drop the assumption of homogeneity in the parameters!!!

As we are interesting in studing if the outcome variable y: 'presence of a defective O-ring' depends on the input variable t: 'temperature', or s: 'pressure'.

Let t_i denote the temperature (in F) in the platform before the ith flight.

Let s_i denote the Leak check pressure (in PSI).

Here are some possible models of interest:

$$\mathscr{M}^I: \quad p(t;\beta_{\mathscr{M}^I},\mathscr{M}^I) = \frac{\exp(\beta_0)}{1+\exp(\beta_0)}$$

$$\mathscr{M}^{II}: \quad p(t;\beta_{\mathscr{M}^{II}},\mathscr{M}^{II}) = \frac{\exp(\beta_0+\beta_1 t)}{1+\exp(\beta_0+\beta_1 t)}$$

$$\mathscr{M}^{III}: \quad p(t;\beta_{\mathscr{M}^{III}},\mathscr{M}^{III}) = \frac{\exp(\beta_0+\beta_2 s)}{1+\exp(\beta_0+\beta_2 s)}$$

$$\mathscr{M}^{IV}: \quad p(t;\beta_{\mathscr{M}^{IV}},\mathscr{M}^{IV}) = \frac{\exp(\beta_0+\beta_1 t+\beta_2 s)}{1+\exp(\beta_0+\beta_1 t+\beta_2 s)}$$

$$\mathscr{M}^{V}: \quad p(t;\beta_{\mathscr{M}^{V}},\mathscr{M}^{V}) = \frac{\exp(\beta_0+\beta_1 t+\beta_2 s+\beta_3 ts)}{1+\exp(\beta_0+\beta_1 t+\beta_2 s+\beta_3 ts)}$$

In the models above, we consider standardise the input variables as

$$t_i \leftarrow \frac{t_i-\bar{t}}{\sqrt{s_t^2}}$$

and

$$s_i \leftarrow \frac{s_i-\bar{s}}{\sqrt{s_s^2}}$$

, in order to eliminate the unites and have the same uning in bothe sides of the equality.

Regarding the prior model, we assign a Normal prior distribution, with mean hyper-parameter b_0 and variance hyper-parameter B_0, on the unknown parameter β to account for the uncertainty about it.

Hmmmm... we could use other priors too ... I just picked one ...

The Bayesian hierarchical model under consideration is:

$$\begin{cases} y_i|\mathscr{M},\beta_{\mathscr{M}} & \sim \text{Bernoulli}(p(x_i;\mathscr{M},\beta_{\mathscr{M}})), \quad \text{for, } i=1,...,n \\ p(x_i;\mathscr{M},\beta_{\mathscr{M}}) & = \frac{\exp(x_i^\top \beta_{\mathscr{M}})}{1+\exp(x_i^\top \beta_{\mathscr{M}})} \\ \beta_j|\mathscr{M} & \sim (1-\mathscr{M}_j)1_0(\beta_j)+\mathscr{M}_j \text{N}(\beta_j|\mu_0,\sigma_0^2), \quad j=1,...,d \\ \mathscr{M} & = (\gamma_1,...,\gamma_d) \\ \gamma_j & \sim \text{Bernoulli}(\varpi_0), \quad j=1,...,d \end{cases}$$

with hyper-parameter values $\mu_0 = 0.0$, $\sigma_0^2 = 100.0$, and $\varpi_0 = 0.5$.

Task

Write the RJAGS program implementing the hierarchical model above, in order a sample of size $N = 100000$ from the posterior distribution

$$(\mathscr{M}^{(j)}, \beta^{(j)}_{\mathscr{M}^{(j)}}) \sim \pi(\mathscr{M}, \beta_{\mathscr{M}}|y_{1:n}) \ , \quad j = 1, ..., N.$$

... your answer

Load the library

```
# Load JAGS
library(rjags)
```

Create an input script, for rjags, containing the Bayesian hierarchical model

```
# init <- list( betaT = rep(0,dmax),
#               ind = rep(0,dmax-1),
#               pp = 0.5)

hierarhicalmodel="
  model {
    for (i in 1:n) {
      mean[i] <- exp(inprod(X[i,],beta)) / (1+exp(inprod(X[i,],beta)))
      y[i] ~ dbern(mean[i])
    }
    betaT[1] ~ dnorm( 0 , 0.1 )
    beta[1] <- betaT[1]
    pp ~ dbeta(1.0,1.0)
    for (j in 1:(dmax-1)) {
      ind[j] ~ dbern( pp )
      betaT[j+1] ~ dnorm( 0 , 0.1 )
      beta[j+1] <- ind[j] * betaT[j+1]
    }
  }
"
```

Create an input list, for jags, containing the data and fixed parameters of the model

```
y_obs <- mydata[ -nrow(mydata) , 4 ] # exclude the last row, and use only the 4th column
y_obs <- as.numeric(y_obs==1)        # make it numeric
n_obs <- length( y_obs )
X_obs <- cbind( rep(1,n_obs),
                as.numeric(mydata[ -nrow(mydata),3]),
                as.numeric(mydata[ -nrow(mydata),5]),
                as.numeric(mydata[ -nrow(mydata),3]) *as.numeric(mydata[ -nrow(mydata),5])  )

# Units correction
X_obs[,2] <- (X_obs[,2]-mean(X_obs[,2])) / sd(X_obs[,2])
X_obs[,3] <- (X_obs[,3]-mean(X_obs[,3])) / sd(X_obs[,3])
X_obs[,4] <- (X_obs[,4]-mean(X_obs[,4])) / sd(X_obs[,4])
```

```
dmax = dim(X_obs)[2]

data.bayes <- list(y = y_obs,
          X = X_obs,
          n = n_obs,
          dmax = dmax)
```

Create an input list, for jags, containing the data and fixed parameters of the model

```
model.smpl <- jags.model( file = textConnection(hierarhicalmodel),
                          data = data.bayes)
```

Initialize the sampler with $N_{\text{adapt}} = 1000$ iterations.

```
adapt(object = model.smpl,
      n.iter = 10^5)
```

Generate a posterior sample of size $N = 10000$.

```
N = 10^5
n.thin = 10^1
n.iter = N * n.thin
output = jags.samples( model = model.smpl,
                       variable.names = c("ind","beta"),
                       n.iter = n.iter,
                       thin = n.thin,
                       )
save.image(file="BernoulliRegressionModelVS.RData")
```

Check the names of the variables sampled

```
names(output)
```

```
## [1] "beta" "ind"
```

Check the dimensions of each of the variables sampled

```
dim( output$beta )
```

```
##           iteration     chain
##         4    100000         1
```

```
dim( output$ind )
```

```
##           iteration     chain
##         3    100000         1
```

```
# the first dimension is the numbers of columns of the variable
# the second dimention is the size of the sample drawn
# the thirs dimention is the number of the sub-samples drawn (in our case it is just 1)
```

Copy the sample of each variable in a vector with a more friendly name...

```
ind.smpl <-output$ind
beta.smpl <-output$beta
```

Task

Calculate the marginal posterior model probabilities of models $\mathscr{M}^I = (0,0,0)$, $\mathscr{M}^{II} = (1,0,0)$, $\mathscr{M}^{III} = (0,1,0)$, $\mathscr{M}^{IV} = (1,1,0)$, $\mathscr{M}^V = (1,1,1)$ $\mathscr{M}^{VI} = (0,0,1)$, $\mathscr{M}^{VII} = (1,0,1)$, and $\mathscr{M}^{VIII} = (0,1,1)$:

$$\Pi(\mathscr{M}^I|y_{1:n}) = \Pi(\mathscr{M}^I = (0,0,0)|y_{1:n})$$

$$\Pi(\mathscr{M}^{II}|y_{1:n}) = \Pi(\mathscr{M}^{II} = (1,0,0)|y_{1:n})$$

$$\Pi(\mathscr{M}^{III}|y_{1:n}) = \Pi(\mathscr{M}^{III} = (0,1,0)|y_{1:n})$$
$$\Pi(\mathscr{M}^{IV}|y_{1:n}) = \Pi(\mathscr{M}^{IV} = (1,1,0)|y_{1:n})$$
$$\Pi(\mathscr{M}^{V}|y_{1:n}) = \Pi(\mathscr{M}^{V} = (1,1,1)|y_{1:n})$$

$$\Pi(\mathscr{M}^{VI}|y_{1:n}) = \Pi(\mathscr{M}^{VI} = (0,0,1)|y_{1:n})$$
$$\Pi(\mathscr{M}^{VII}|y_{1:n}) = \Pi(\mathscr{M}^{VII} = (1,0,1)|y_{1:n})$$

$$\Pi(\mathscr{M}^{VII}|y_{1:n}) = \Pi(\mathscr{M}^{VII} = (0,1,1)|y_{1:n})$$

... your answer

It is

```
# Compute the marginal distributions
pr_1 = mean(as.numeric(ind.smpl[1,,]==0) * as.numeric(ind.smpl[2,,]==0)* as.numeric(ind.smpl[3,,]==0))
pr_2 = mean(as.numeric(ind.smpl[1,,]==1) * as.numeric(ind.smpl[2,,]==0)* as.numeric(ind.smpl[3,,]==0))
pr_3 = mean(as.numeric(ind.smpl[1,,]==0) * as.numeric(ind.smpl[2,,]==1)* as.numeric(ind.smpl[3,,]==0))
pr_4 = mean(as.numeric(ind.smpl[1,,]==1) * as.numeric(ind.smpl[2,,]==1)* as.numeric(ind.smpl[3,,]==0))
pr_5 = mean(as.numeric(ind.smpl[1,,]==1) * as.numeric(ind.smpl[2,,]==1)* as.numeric(ind.smpl[3,,]==1))
pr_6 = mean(as.numeric(ind.smpl[1,,]==0) * as.numeric(ind.smpl[2,,]==0)* as.numeric(ind.smpl[3,,]==1))
pr_7 = mean(as.numeric(ind.smpl[1,,]==1) * as.numeric(ind.smpl[2,,]==0)* as.numeric(ind.smpl[3,,]==1))
pr_8 = mean(as.numeric(ind.smpl[1,,]==0) * as.numeric(ind.smpl[2,,]==1)* as.numeric(ind.smpl[3,,]==1))
# Print the marginal distributions
pr_1
```

```
## [1] 0.09354
```

```
pr_2
```

```
## [1] 0.40226
```

```
pr_3
```

```
## [1] 0.00694
```

```
pr_4
```

```
## [1] 0.10706
```

```
pr_5
```

```
## [1] 0.22981
```

```
pr_6
```

```
## [1] 0.00504
```

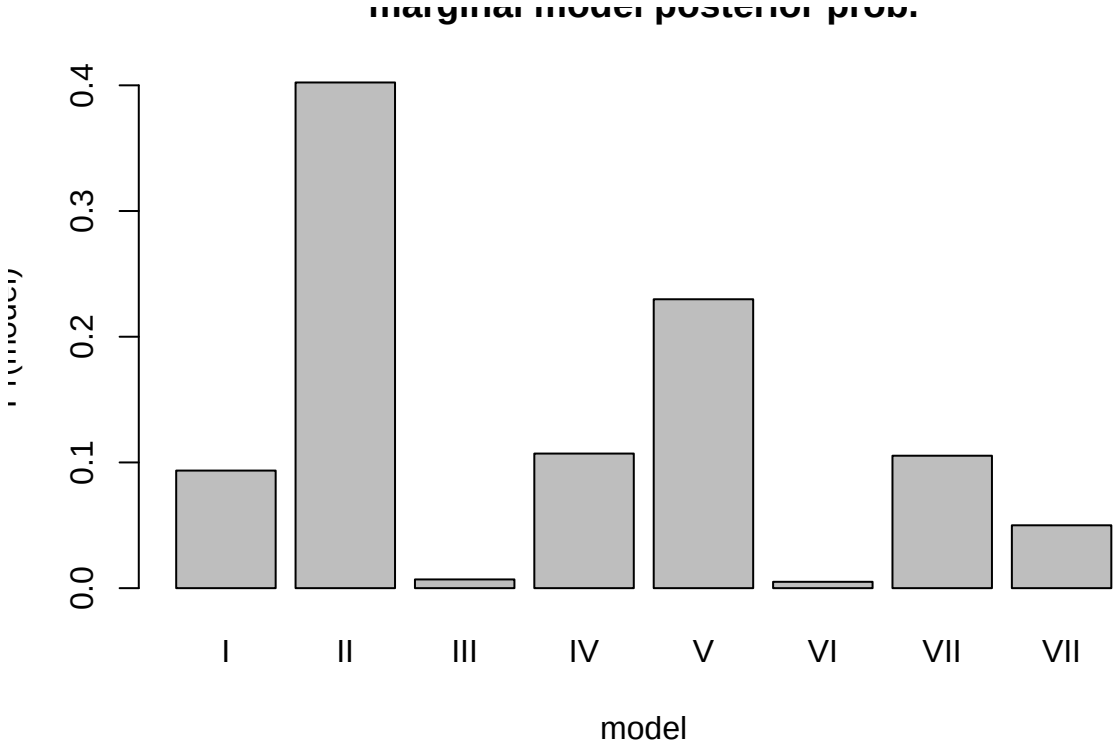
```
pr_7
```

```
## [1] 0.10534
```

```
pr_8
```

```
## [1] 0.05001
```

```
# plot the probabilities
Pr <- c(pr_1, pr_2,pr_3, pr_4, pr_5, pr_6, pr_7, pr_8)
M <- c(1:8)
barplot(Pr,
        names.arg = c('I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VII'),
        xlab = 'model',
        ylab = 'Pr(model)',
        main = 'marginal model posterior prob.')
```



Task

Assume that only one of the model in the set

$$\mathcal{M} = \{\mathscr{M}^I, \mathscr{M}^{II}, \mathscr{M}^{III}, \mathscr{M}^{IV}, \mathscr{M}^V\}$$

is of interest. This is because it is

Calculate the posterior model probabilities of models $\mathscr{M}^I = (0,0,0)$, $\mathscr{M}^{II} = (1,0,0)$, $\mathscr{M}^{III} = (0,1,0)$, $\mathscr{M}^{IV} = (1,1,0)$, $\mathscr{M}^V = (1,1,1)$, given the model collection $\mathcal{M}$.

$$\Pi(\mathscr{M}^I|y_{1:n},\mathcal{M}) = \Pi(\mathscr{M}^I = (0,0,0)|y_{1:n},\mathcal{M})$$

$$\Pi(\mathscr{M}^{II}|y_{1:n},\mathcal{M}) = \Pi(\mathscr{M}^{II} = (1,0,0)|y_{1:n},\mathcal{M})$$

$$\begin{aligned}\Pi(\mathscr{M}^{III}|y_{1:n},\mathcal{M}) &= \Pi(\mathscr{M}^{III} = (0,1,0)|y_{1:n},\mathcal{M})\\ \Pi(\mathscr{M}^{IV}|y_{1:n},\mathcal{M}) &= \Pi(\mathscr{M}^{IV} = (1,1,0)|y_{1:n},\mathcal{M})\\ \Pi(\mathscr{M}^{V}|y_{1:n},\mathcal{M}) &= \Pi(\mathscr{M}^{V} = (1,1,1)|y_{1:n},\mathcal{M})\end{aligned}$$

Which model is a posteriori the most probabile?

... your answer

It is

```
# Compute the marginal distributions
pr_1 = mean(as.numeric(ind.smpl[1,,]==0) * as.numeric(ind.smpl[2,,]==0)* as.numeric(ind.smpl[3,,]==0))
pr_2 = mean(as.numeric(ind.smpl[1,,]==1) * as.numeric(ind.smpl[2,,]==0)* as.numeric(ind.smpl[3,,]==0))
pr_3 = mean(as.numeric(ind.smpl[1,,]==0) * as.numeric(ind.smpl[2,,]==1)* as.numeric(ind.smpl[3,,]==0))
pr_4 = mean(as.numeric(ind.smpl[1,,]==1) * as.numeric(ind.smpl[2,,]==1)* as.numeric(ind.smpl[3,,]==0))
pr_5 = mean(as.numeric(ind.smpl[1,,]==1) * as.numeric(ind.smpl[2,,]==1)* as.numeric(ind.smpl[3,,]==1))
pr_12345 = pr_1+pr_2+pr_3+pr_4+pr_5
# Compute the conditional distributions
pr_1 = pr_1/pr_12345
pr_2 = pr_2/pr_12345
pr_3 = pr_3/pr_12345
pr_4 = pr_4/pr_12345
pr_5 = pr_4/pr_12345
# Print the conditional distributions
pr_1
```

```
## [1] 0.1114089
```

```
pr_2
```

```
## [1] 0.4791034
```

```
pr_3
```

```
## [1] 0.008265742
```

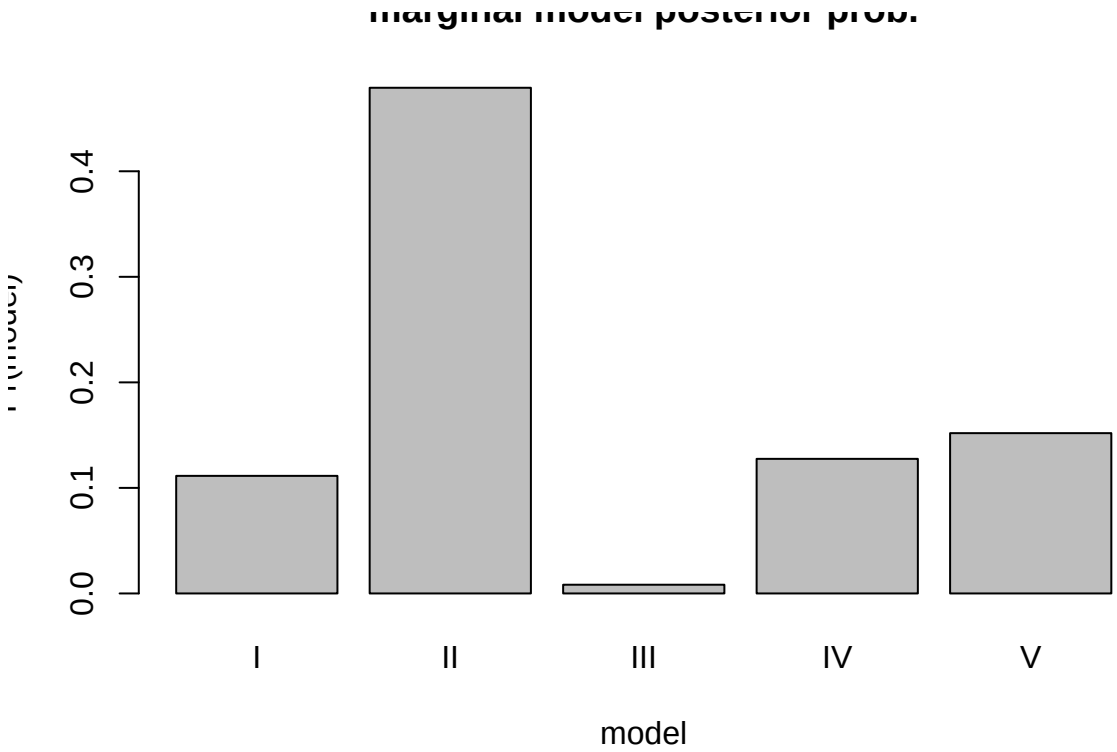
```
pr_4
```

```
## [1] 0.1275116
```

```
pr_5
```

```
## [1] 0.15187
```

```
# plot the probabilities
Pr <- c(pr_1, pr_2, pr_3, pr_4, pr_5)
barplot(Pr,
        names.arg = c('I', 'II', 'III', 'IV', 'V'),
        xlab = 'model',
        ylab = 'Pr(model)',
        main = 'marginal model posterior prob.')
```



The most model with the highest marginal model posterior probability is

$$\mathscr{M}^{II}: \quad p(t; \beta_{\mathscr{M}^{II}}, \mathscr{M}^{II}) = \frac{\exp(\beta_0 + \beta_1 t)}{1 + \exp(\beta_0 + \beta_1 t)}$$