Exam Bayesian Methods

The exam will be available on 12. August 2017, 11:00.

Solutions will be submitted (sent via email to `oliver@kirchkamp.de`) 45 minutes after the exam is on-line. A late submission penalty applies.

For all your answers write down (all) the commands you use to obtain these answers. Also include the answers in a short form (for each question do *not* include more than one line of output – usually your answer should just be one or two numbers).

1. Consider the following sample of x:

```
set.seed(123)
x <- rnorm(5,5,1)
```

   Assume that $x \sim N(\mu, 1)$ where N is the normal distribution. You compare three models: $\mu_1 = 3.8$, $\mu_2 = 4$, $\mu_3 = 6$.

   a) Explain briefly why using pseudopriors can help with model selection.

   b) Discuss briefly whether pseudopriors can help here?

      [[They could not — once the model is selected there are no other parameters left.]]

   c) Assume a vague prior. How probable is each of these three models? Use `jags` to estimate the posterior probability of each model.

      [[Please note that these three models are each determined by a single parameter (μ). In the lecture we studied a more complicated case which involved estimating a relationship between two variables. Here we have only a single variable (x), this simplifies matters a lot.]]

```
library(runjags)
library(coda)
c.model <- 'model {
for (i in 1:length(x)) {
   x[i] ~ dnorm(mu[m],1)
}
 m ~ dcat(modelProb)
}'
c.data<-list(x=x,modelProb=c(1,1,1),mu=c(3.8,4,6))
c.jags<-run.jags(c.model,c.data,monitor=c("m"))
with(data.frame(as.mcmc(c.jags)),table(m)/length(m))

## m
##       1       2       3
## 0.03205 0.12080 0.84715
```

2. Consider the following model for JAGS:

```
'model {
for (i in 1:length(y)) {
    y[i] ~ dnorm(mu[i],tau[K+1])
    mu[i]<-inprod(beta+nu[,group[i]],X[i,])
}
for (k in 1:K) {
    beta[k] ~ dnorm (0,.0001)
    for (j in 1:max(group)) {
        nu[k,j] ~ dnorm(0,tau[k])
    }
}
for (k in 1:(K+1)) {
    tau[k] ~ dgamma(m[k]^2/d[k]^2,m[k]/d[k]^2);
    m[k] ~ dexp(1); d[k] ~ dexp(1);
}
}'
```

a) Which type of models can this JAGS model estimate?

   [[A model with K many random effects (one random effect for each fixed effect)]]

b) When you call `run.jags` you can specify a `data` argument. What type of model do you estimate when neither `nu` nor `tau` are included in this data argument.

   [[You include a random effect for each coefficient of the regression. (There is one `nu[k,]` for each `beta[k]`). If `nu` is not restricted by `data`, then `nu` follows the specification of the model, i.e. it can be a random effect.]]

c) Now assume that with the `data` argument you include a matrix `nu` where `nu[1,]=NA` and `nu[-1,]=0` (i.e. the first column of `nu` is `NA`, all other columns are zero). What model do you estimate now?

   [[You include a random effect only for the first coefficient of the regression (usually the intercept).]]

d) Can you achieve a similar result if you include a vector `tau` with your `data` argument (and not a matrix `nu`)?

   [[The vector `tau` should have length K+1. The first and the last element of `tau` should be NA (i.e. we use the vague prior from the model), all other elements should be set to a large value (e.g. 10^5, i.e. we assume the corresponding elements of `nu` to be only slightly different from zero, hence the random effect almost vanish.]]

3. Consider the following data:

```
x<-1:10
y<-pmax(x-5,0)
```

y is censored from below at 0, i.e. when you observe y=0 you only know that the underlying variable $y \leqslant 0$. For y>0 you know that $y = y$. With your data you estimate the following:

$$y = \beta_0 + \beta_1 x + u.$$

a) Write a model in JAGS to solve this problem.

```
intreg.model <- 'model {
   for (i in 1:length(y)) {
      y[i] ~ dnorm(inprod(beta,X[i,]),tau)
      notCens[i] ~ dinterval(y[i],0)
   }
   for (k in 1:K) {
      beta[k] ~ dnorm(0,.0001)
   }
   tau ~ dgamma(m^2/d^2,m/d^2);
   m ~ dexp(1); d ~ dexp(1);
}'
int.data<-within(list(y=y,notCens=as.numeric(y>0),
                      X=cbind(1,x),K=2),y[!notCens]<-NA)
ini <- with(int.data,function(i) list(y=ifelse(notCens,NA,-1)))
run.jags(intreg.model,int.data,inits=ini,monitor="beta")

##
## JAGS model summary statistics from 20000 samples (chains = 2; adapt+burni
##
##          Lower95  Median Upper95    Mean     SD    Mode     MCerr MC%ofSD
## beta[1] -8.9551 -5.7183 -3.5707 -5.9332 1.3806 -5.5187 0.092367     6.7
## beta[2] 0.80377  1.0827  1.5001  1.1081 0.1743  1.0614 0.011583     6.6
##
##          SSeff   AC.10   psrf
## beta[1]    223 0.80464 1.0003
## beta[2]    226 0.78725 1.0009
##
## Total time taken: 0.4 seconds
```

b) You suspect that y contains outliers. How can you make the estimation of β robust?

```
intreg2.model <- 'model {
   for (i in 1:length(y)) {
      y[i] ~ dt(inprod(beta,X[i,]),tau,df)
      notCens[i] ~ dinterval(y[i],0)
   }
   for (k in 1:K) {
      beta[k] ~ dnorm(0,.0001)
```

```
  }
  tau ~ dgamma(m^2/d^2,m/d^2);
  m ~ dexp(1); d ~ dexp(1);
  df ~ dexp(1/30)
}'
```

```r
run.jags(intreg2.model,int.data,inits=ini,monitor=c("beta","df"))
```

```
##
## JAGS model summary statistics from 20000 samples (chains = 2; adapt+burni
##
##          Lower95  Median Upper95    Mean      SD    Mode    MCerr MC%ofSD
## beta[1] -10.043 -5.8485 -3.3944 -6.2399  1.6752 -5.5036  0.17435    10.4
## beta[2] 0.80114  1.0989  1.6344  1.1461 0.20838  1.0527 0.021026    10.1
## df      0.45221   24.57  95.833   33.69  30.558  13.041  0.48282     1.6
##
##          SSeff    AC.10    psrf
## beta[1]     92  0.91216  1.0063
## beta[2]     98  0.90062  1.0075
## df        4006 0.010935 0.99997
##
## Total time taken: 1.6 seconds
```