Hands-on training session 4

BLCMs for Se & Sp estimation with covariates

Sonja Hartnack   Eleftherios Meletis

2020-02-12

Session 4: BLCMs for Se & Sp estimation with covariates

Overview

Date/time:

- 20th February 2020
- 16.00 - 17.00

Teachers:

- Sonja Hartnack (presenter)
- Eleftherios Meletis

Recap

We looked at a GLM-type formulation of this model yesterday:

```
model{

  for(i in 1:N){
    Status[i] ~ dcat(prob[i, ])

      prob[i,1] <- (prev[i] * ((1-se[1])*(1-se[2]))) +
                  ((1-prev[i]) * ((sp[1])*(sp[2])))
      prob[i,2] <- (prev[i] * ((se[1])*(1-se[2]))) +
                  ((1-prev[i]) * ((1-sp[1])*(sp[2])))
      prob[i,3] <- (prev[i] * ((1-se[1])*(se[2]))) +
                  ((1-prev[i]) * ((sp[1])*(1-sp[2])))
      prob[i,4] <- (prev[i] * ((se[1])*(se[2]))) +
                  ((1-prev[i]) * ((1-sp[1])*(1-sp[2])))

      logit(prev[i]) <- intercept + population_effect[Population[i]]
  }

  intercept ~ dnorm(0, 0.33)
```

We can think of “Population” as a covariate, and add others. For example:

```
se1 <- 0.9
sp1 <- 0.95
sp2 <- 0.99
se2 <- 0.8

int <- -1.5
pop_eff <- c(0, 1.5, 3)
sex_eff <- c(0, 0.2)
age_eff <- 0.1
N <- 300

simdata <- data.frame(
  Population = sample(seq_along(pop_eff), N, replace=TRUE),
  Sex = sample(seq_along(sex_eff), N, replace=TRUE),
  Age = runif(N, -3, 3)
  )
simdata$logitprob <- with(simdata, int + pop_eff[Population] + sex_eff[Sex] + age_eff*Age)

```

We can cheat a bit and use template.jags to help set up part of the model:

```r
simdata$Population <- factor(simdata$Population, levels=seq_along(pop_eff), labels=paste0('Pop_', seq_along(pop_eff)))
simdata$Sex <- factor(simdata$Sex, levels=seq_along(sex_eff), labels=c('female','male'))

runjags::template.jags(truestatus ~ Population + Sex + Age, data=simdata, file='glmtemplate.bug', family='binomial', effect.prior = 'dnorm(0, 0.1)', write.data=FALSE)
```

```
## Your model template was created at "glmtemplate.bug" - it is highly advisable to examine the model syntax to be sure it is as intended
## You can then run the model using run.jags("glmtemplate.bug", data=data) - where "data" is the same data frame specified to the template.jags function
```

The lines you need from this auto-generated model are #17:

```
##  logit(regression_prob[i]) <- intercept + Age_coefficient *
```

And also lines 21:27 - changing priors as needed:

```
## intercept ~ dnorm(0, 0.1)
## Age_coefficient ~ dnorm(0, 0.1)
## Population_effect[1] <- 0    # Factor level "Pop_1"
## Population_effect[2] ~ dnorm(0, 0.1)    # Factor level "Pop_2"
## Population_effect[3] ~ dnorm(0, 0.1)    # Factor level "Pop_3"
## Sex_effect[1] <- 0    # Factor level "female"
## Sex_effect[2] ~ dnorm(0, 0.1)    # Factor level "male"
```

Line 33 may also be helpful (we don't need deviance, dic or resid.sum.sq):

```
## #monitor# intercept, Age_coefficient, Population_effect, Sex_effect, deviance, dic, resid.sum.sq
```

And lines 39-57 for initial values:

```
## inits{
## "intercept" <- 1
## "Age_coefficient" <- -1
## "Population_effect" <- c(NA, -1, -1)
## "Sex_effect" <- c(NA, 1)
## }
##
## inits{
## "intercept" <- 1
## "Age_coefficient" <- -1
## "Population_effect" <- c(NA, -1, -1)
## "Sex_effect" <- c(NA, -1)
## }
```

Combine the two models, add initial values for se and sp, and also add a ‘#data# N, Status, Age, Population, Sex´ block, and modify the priors for test1 se/sp as estimated from session 1.

To run the model:

```
N <- nrow(simdata)
simdata$Status <- with(simdata, factor(interaction(Test1, Test2),
↪  levels=c(’0.0’,’1.0’,’0.1’,’1.1’)))
```

```
results <- run.jags(’HW_GLM_coefs.bug’, data=simdata)
results
```

```
##
## JAGS model summary statistics from 20000 samples (chains = 2;
↪  adapt+burnin = 5000):
##
##                         Lower95   Median Upper95    Mean
## intercept               -2.5118  -1.8129 -1.1484 -1.8198
## Age_coefficient        -0.06389  0.11134 0.28753 0.11173
## Population_effect[1]          0        0       0       0
## Population_effect[2]     1.0971   1.8202  2.6101  1.8276
## Population_effect[3]     1.5928   2.3509  3.0997  2.3543
## Sex_effect[1]                 0        0       0       0
## Sex_effect[2]         -0.048858  0.52302  1.1071 0.52299
## se[1]                   0.84597  0.88955  0.9296 0.88868
## se[2]                   0.73011  0.80753 0.88197 0.80619
## sp[1]                   0.92561  0.95236 0.97479 0.95117
## sp[2]                   0.89179  0.94748  0.9951  0.9449
##
##                            SD    Mode      MCerr MC%ofSD
## intercept             0.34224 -1.8047   0.008943     2.6
## Age_coefficient       0.08933 0.10934 0.00082667     0.9
## Population_effect[1]        0       0          -       -
## Population_effect[2]  0.38769  1.8211  0.0086571     2.2
## Population_effect[3]  0.38546  2.3499  0.0087032     2.3
## Sex_effect[1]               0       0
```

Compare these to the simulation parameters:

```
## int:  -1.5
```

```
## age_eff:  0.1
```

```
## pop_eff:  0, 1.5, 3
```

```
## sex_eff:  0, 0.2
```

```
## se1:  0.9
```

```
## se2:  0.8
```

```
## sp1:  0.95
```

```
## sp2:  0.99
```

Exercise

Instructions here

Summary

Take-away points