

Hands-on training session 4

BLCMs for Se & Sp estimation with covariates

Sonja Hartnack Eleftherios Meletis

2020-02-19

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:

```
1  model{
2
3    for(i in 1:N){
4      Status[i] ~ dcat(prob[i, ])
5
6      prob[i,1] <- (prev[i] * ((1-se[1])*(1-se[2]))) +
7                  ((1-prev[i]) * ((sp[1])*(sp[2])))
8      prob[i,2] <- (prev[i] * ((se[1])*(1-se[2]))) +
9                  ((1-prev[i]) * ((1-sp[1])*(sp[2])))
10     prob[i,3] <- (prev[i] * ((1-se[1])*(se[2]))) +
11                 ((1-prev[i]) * ((sp[1])*(1-sp[2])))
12     prob[i,4] <- (prev[i] * ((se[1])*(se[2]))) +
13                 ((1-prev[i]) * ((1-sp[1])*(1-sp[2])))
14
15     logit(prev[i]) <- intercept +
16       ↪ population_effect[Population[i]]
17   }
18
19   intercept ~ dnorm(0, 0.33)
```

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

```
1  se1 <- 0.9
2  sp1 <- 0.95
3  sp2 <- 0.99
4  se2 <- 0.8
5
6  int <- -1.5
7  pop_eff <- c(0, 1.5, 3)
8  sex_eff <- c(0, 0.2)
9  age_eff <- 0.1
10 N <- 300
11
12 simdata <- data.frame(
13   Population = sample(seq_along(pop_eff), N, replace=TRUE),
14   Sex = sample(seq_along(sex_eff), N, replace=TRUE),
15   Age = runif(N, -3, 3)
16 )
17 simdata$logitprob <- with(simdata, int + pop_eff[Population] +
18   ↪ sex_eff[Sex] + age_eff*Age)
```

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

```
1  simdata$Population <- factor(simdata$Population,  
  ↪  levels=seq_along(pop_eff), labels=paste0('Pop_',  
  ↪  seq_along(pop_eff)))  
2  simdata$Sex <- factor(simdata$Sex, levels=seq_along(sex_eff),  
  ↪  labels=c('female','male'))  
3  
4  runjags::template.jags(truestatus ~ Population + Sex + Age,  
  ↪  data=simdata, file='glmtemplate.bug', family='binomial',  
  ↪  effect.prior = 'dnorm(0, 0.1)', write.data=FALSE)  
  
1  ## Your model template was created at "glmtemplate.bug" - it is  
  ↪  highly advisable to examine the model syntax to be sure it is  
  ↪  as intended  
2  ## 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:

```
1  ## logit(regression_prob[i]) <- intercept + Age_coefficient *  
  ↪  Age[i] + Population_effect[Population[i]] +
```

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

```
1  ## intercept ~ dnorm(0, 0.1)
2  ## Age_coefficient ~ dnorm(0, 0.1)
3  ## Population_effect[1] <- 0    # Factor level "Pop_1"
4  ## Population_effect[2] ~ dnorm(0, 0.1)    # Factor level "Pop_2"
5  ## Population_effect[3] ~ dnorm(0, 0.1)    # Factor level "Pop_3"
6  ## Sex_effect[1] <- 0    # Factor level "female"
7  ## 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):

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


And lines 39-57 for initial values:

```
1  ## inits{
2  ## "intercept" <- 1
3  ## "Age_coefficient" <- -1
4  ## "Population_effect" <- c(NA, -1, -1)
5  ## "Sex_effect" <- c(NA, 1)
6  ## }
7  ##
8  ## inits{
9  ## "intercept" <- 1
10 ## "Age_coefficient" <- -1
11 ## "Population_effect" <- c(NA, -1, -1)
12 ## "Sex_effect" <- c(NA, -1)
13 ## }
```

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:

```
1 N <- nrow(simdata)
2 simdata$Status <- with(simdata, factor(interaction(Test1, Test2),
  ↳ levels=c('0.0','1.0','0.1','1.1'))))

1 results <- run.jags('HW_GLM_coefs.bug', data=simdata)
2 results
```

```

1  ##
2  ## JAGS model summary statistics from 20000 samples (chains = 2;
   ↪ adapt+burnin = 5000):

```

```

3  ##
4  ##           Lower95   Median Upper95     Mean
5  ## intercept       -2.5118 -1.8129 -1.1484 -1.8198
6  ## Age_coefficient  -0.06389 0.11134 0.28753 0.11173
7  ## Population_effect[1]      0      0      0      0
8  ## Population_effect[2]      1.0971  1.8202  2.6101  1.8276
9  ## Population_effect[3]      1.5928  2.3509  3.0997  2.3543
10 ## Sex_effect[1]           0      0      0      0
11 ## Sex_effect[2]        -0.048858 0.52302  1.1071 0.52299
12 ## se[1]                0.84597 0.88955  0.9296 0.88868
13 ## se[2]                0.73011 0.80753 0.88197 0.80619
14 ## sp[1]                0.92561 0.95236 0.97479 0.95117
15 ## sp[2]                0.89179 0.94748  0.9951  0.9449

```

```

16 ##
17 ##           SD      Mode      MCerr MC%ofSD
18 ## intercept      0.34224 -1.8047   0.008943    2.6
19 ## Age_coefficient  0.08933 0.10934 0.00082667    0.9
20 ## Population_effect[1]      0      0      -      -
21 ## Population_effect[2]      0.38769  1.8211  0.0086571    2.2
22 ## Population_effect[3]      0.38546  2.3499  0.0087032    2.3
23 ## Sex_effect[1]           0      0

```

Compare these to the simulation parameters:

```
1  ## int:  -1.5
1  ## age_eff:  0.1
1  ## pop_eff:  0, 1.5, 3
1  ## sex_eff:  0, 0.2
1  ## se1:  0.9
1  ## se2:  0.8
1  ## sp1:  0.95
1  ## sp2:  0.99
```

Exercise

Instructions here

Summary

Take-away points