Session 8

Practical session

Matt Denwood

2022-06-10

Practical session

NOTE: THIS MATERIAL IS NOT YET FINALISED, PLEASE CHECK BACK SOON!

This is the final session of the course!

We have three options to choose from:

1. Finish up working on practical exercises from previous sessions
2. Work on your own data and ask us for help/advice
3. Look at how to implement custom distributions/functions via a JAGS module

JAGS modules

TODO: redo so it is covariates

- covariates on Se/Sp
- covariates on prevalence (random effects)

Example: cervical cancer screening (covariates: age, pregnancy)

TODO: Ask Sonja if she has any material (or Lef)

Model at individual vs group level

TODO: session 8 = work on own data

Models for diagnostic test evaluation require:

- At least 2 tests
- At least 2 populations, but preferably 3 or more

JAGS modules

TODO: redo so it is covariates

- covariates on Se/Sp
- covariates on prevalence (random effects)

Example: cervical cancer screening (covariates: age, pregnancy)

TODO: Ask Sonja if she has any material (or Lef)

Model at individual vs group level

TODO: session 8 = work on own data

Models for diagnostic test evaluation require:

- At least 2 tests
- At least 2 populations, but preferably 3 or more

Incorporating imperfect sensitivity and specificity into more complex models

Logistic regression in JAGS

```
model{

  for(i in 1:N){
    Observation[i] ~ dbern(prob[i])
    logit(prob[i]) <- intercept + beta1[Category[i]] + beta2*Covariate[i]
  }

  intercept ~ dnorm(0, 0.01)
  beta1[1] <- 0
  for(c in 2:NC){
    beta1[c] ~ dnorm(0, 0.01)
  }
  beta2 ~ dnorm(0, 0.01)

  #data# N, Observation, NC, Category, Covariate
  #monitor# intercept, beta1, beta2
  #inits# intercept, beta1, beta2
}
```

```
model{

  for(i in 1:N){
    Observation[i] ~ dbern(obs_prob[i])
    obs_prob[i] <- prob[i]*se + (1-prob[i])*(1-sp)
    logit(prob[i]) <- intercept + beta1[Category[i]] + beta2*Covariate[i]
  }

  se ~ dbeta(1,1)T(1-sp, )
  sp ~ dbeta(1,1)

  intercept ~ dnorm(0, 0.01)
  beta1[1] <- 0
  for(c in 2:NC){
    beta1[c] ~ dnorm(0, 0.01)
  }
  beta2 ~ dnorm(0, 0.01)

  #data# N, Observation, NC, Category, Covariate
  #monitor# intercept, beta1, beta2, se, sp
  #inits# intercept, beta1, beta2, se, sp
}
```

```
model{

  for(i in 1:N){
    Observation[i] ~ dbern(obs_prob[i])
    obs_prob[i] <- prob[i]*se + (1-prob[i])*(1-sp)
    logit(prob[i]) <- intercept + beta1[Category[i]] + beta2*Covariate[i]
  }

  se ~ dbeta(148.43, 16.49)T(1-sp, )
  sp ~ dbeta(240.03, 12.63)

  intercept ~ dnorm(0, 0.01)
  beta1[1] <- 0
  for(c in 2:NC){
    beta1[c] ~ dnorm(0, 0.01)
  }
  beta2 ~ dnorm(0, 0.01)

  #data# N, Observation, NC, Category, Covariate
  #monitor# intercept, beta1, beta2, se, sp
  #inits# intercept, beta1, beta2, se, sp
}
```

```
model{

  for(i in 1:N){
    Observation[i] ~ dbern(obs_prob[i])
    obs_prob[i] <- prob[i]*se + (1-prob[i])*(1-sp)
    logit(prob[i]) <- intercept + beta1[Category[i]] + beta2*Covariate[i]
  }

  se <- 0.9
  sp <- 0.95

  intercept ~ dnorm(0, 0.01)
  beta1[1] <- 0
  for(c in 2:NC){
    beta1[c] ~ dnorm(0, 0.01)
  }
  beta2 ~ dnorm(0, 0.01)

  #data# N, Observation, NC, Category, Covariate
  #monitor# intercept, beta1, beta2
  #inits# intercept, beta1, beta2
}
```

```
model{

  for(i in 1:N){
    Observation[i] ~ dbern(obs_prob[i])
    obs_prob[i] <- prob[i]*se + (1-prob[i])*(1-sp)
    logit(prob[i]) <- intercept + beta1[Category[i]] + beta2*Covariate[i]
  }

  #data# se, sp

  intercept ~ dnorm(0, 0.01)
  beta1[1] <- 0
  for(c in 2:NC){
    beta1[c] ~ dnorm(0, 0.01)
  }
  beta2 ~ dnorm(0, 0.01)

  #data# N, Observation, NC, Category, Covariate
  #monitor# intercept, beta1, beta2
  #inits# intercept, beta1, beta2
}
```

```
model{

  for(i in 1:N){
    Observation[i] ~ dbern(obs_prob[i])
    obs_prob[i] <- prob[i]*se[Test[i]] + (1-prob[i])*(1-sp[Test[i]])
    logit(prob[i]) <- intercept + beta1[Category[i]] + beta2*Covariate[i]
  }

  #data# se, sp

  intercept ~ dnorm(0, 0.01)
  beta1[1] <- 0
  for(c in 2:NC){
    beta1[c] ~ dnorm(0, 0.01)
  }
  beta2 ~ dnorm(0, 0.01)

  #data# N, Observation, NC, Category, Covariate, Test
  #monitor# intercept, beta1, beta2
  #inits# intercept, beta1, beta2
}
```

Other types of GL(M)M

You can use template.jags as inspiration:

```
template.jags(weight ~ group, family="gaussian", data=data, file="linear_model.txt")
## Your model template was created at "linear_model.txt" - 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("linear_model.txt")
results <- run.jags("linear_model.txt")
## Loading required namespace: rjags
## module glm loaded
## module dic loaded
```

```
results
##
## JAGS model summary statistics from 20000 samples (chains = 2; adapt+burnin = 5000):
##
##                      Lower95   Median Upper95      Mean
## regression_precision 0.87586   1.9873  3.4635    2.0647
## intercept              4.568   5.0328  5.4847    5.0312
## group_effect[1]            0        0       0         0
## group_effect[2]      -1.0291 -0.37196 0.27904 -0.36941
## deviance              40.186   42.708  48.621    43.398
## resid.sum.sq          8.7293   9.4208  12.132    9.8121
##
##                           SD     Mode     MCerr MC%ofSD
## regression_precision 0.68618   1.8649  0.005411     0.8
## intercept             0.2323   5.0397 0.0016426     0.7
## group_effect[1]            0        0        --      --
## group_effect[2]      0.33033 -0.36384 0.0023358     0.7
## deviance              2.6319   41.688  0.021297     0.8
## resid.sum.sq          1.2258   9.0318 0.0096191     0.8
##
##                      SSeff       AC.10    psrf
## regression_precision 16081  -0.0061548  1.0003
## intercept            20000  0.00097251 0.99997
## group_effect[1]         --          --      --
## group_effect[2]      20000 -0.00095213  1.0001
```

Supported features:

- Gaussian, binomial, Poisson, negative binomial, ZIB, ZIP, ZINB
- Random intercepts

We can also add (currently manually):

- Random slopes
- Spline terms
- Interval censoring

What about other models?

MCMC is highly flexible!

What about other models?

MCMC is highly flexible!

We can have:

- Hidden Markov models
- State Space models
- Other types of latent class model

What about other models?

MCMC is highly flexible!

We can have:

- Hidden Markov models
- State Space models
- Other types of latent class model

But does your data match your ambitions?

- All models can be specified
- Relatively few are identifiable

Before you go. . .

- Feedback on the course would be extremely welcome!
  - https://www.survey-xact.dk/LinkCollector?key=RKMUENCXS11N
  - I will send a reminder email later today with (the same) survey link

Before you go. . .

- Feedback on the course would be extremely welcome!
  - https://www.survey-xact.dk/LinkCollector?key=RKMUENCXS11N
  - I will send a reminder email later today with (the same) survey link
- Remember to keep an eye on the COST action website:
  - http://harmony-net.eu
  - Physical training schools are being run in September and accepting sign-ups now!

Practical session 7

Points to consider

1. When is there a benefit to adding imperfect test characteristics?
2. When is there no real benefit?