

Practical 5. Evidence synthesis and decision models

Tuesday, 21 June 2022

[Lecture 5](#)
[PDF version](#)

1. Aggregated level data

The file [EvSynth.R](#) is a script that can be used to perform the Bayesian analysis of the influenza model, described in class. This can in turn be used to also extend the analysis to the cost-effectiveness evaluation of prophylaxis.

1. First, open the file [EvSynth.txt](#), which contains the BUGS translation of the model. Make sure you understand the structure and the underlying assumptions.
2. Following the script, load the data available from the published studies on the effectiveness of prophylaxis with NIs and the incidence of influenza, and load the data on costs and the other relevant variables for the model.
3. Following the script, set up the data list, the pointer to the file containing the model code and the parameters list.
4. Following the script, write a function that creates random initial values to be associated to the parameters of the model.

i NB: the R notation `alpha=rnorm(S,0,1)` creates `S` random draws from a Normal of mean 0 and standard deviation 1.

Are all the necessary nodes initialised by the function `inits`? If not, add the missing nodes (together with a suitable random initialisation) to the function. (*Hint: you need to check the model code and that all the unobserved random quantities are initialised*).

5. Following the script, set up the number of iterations, burn-in and thinning to be used and run the MCMC simulation calling `R2OpenBUGS` from within R. You can tweak with the thinning parameter by increasing (or decreasing) it to assess its impact to model convergence.
6. Following the script, use the R command `print` to display the summary statistics of the MCMC run.

i NB: the arguments `digits=3` and `intervals=c(0.025, 0.975)` instruct R to show up to 3 significant digits in the table and to display the 95% central credible intervals, based on the 2.5% and the 97.5% percentiles of the posterior distributions.)

7. Using the table displayed by R with the summary for the posterior distributions of the monitored notes, comment on convergence, specifically with reference to the \hat{R} statistics and the effective sample size.
8. Following the code, produce a traceplot for the node `p1` and comment on convergence.

i NB: in R, the command `plot` is used to produce graphs. The number and type of arguments that it can take vary depending on the type of plot you are trying to make. In this case, you are selecting the first 500 simulations for the node `p1`. These are stored in the element `sims.list$p1` of the object `es` — the R notation `[1:500]` indicates that you want to select the values from the first to the five-hundredth — these are the values associated with the first chain. The additional option `ylabel=p1` instructs R to use the string `p1` as label for the y-axis of the graph. To add another layer to an existing plot, you can use the R command `points`. In this case, you give a similar argument, except you are requiring the values from 501 to 1000 of the element `sims.list$p1` — these are the simulations for the second chain)

9. Try and produce a trace (history) plot for other nodes.
10. Now, still following the script, prepare to run the economic analysis of this model. First you need to `attach` the object `es` to your current R session. Then you need to combine the model output (in terms of the MCMC simulations) so that suitable measures of costs and effectiveness are created.
11. Following the script, load the package `BCEA` and perform the basic cost-effectiveness analysis.
12. If you wish to, you can replicate the MCMC analysis by running `R2OpenBUGS` directly (i.e. without interfacing it with R). The files [script.txt](#), [data.txt](#), [inits1.txt](#) and [inits2.txt](#) contain the script to run the entire MCMC analysis from `R2OpenBUGS`, the data and two sets of initial values.

