Analysing Spatial Data in R: Worked examples: (Bayesian) disease mapping II

Virgilio Gómez-Rubio

Department of Epidemiology and Public Heath
Imperial College London
London, UK

31 August 2007
Bayesian Disease mapping

- Bayesian Estimation in Disease Mapping has been one of the leading topics in spatial statistics in the last 20 years
- Bayesian Hierarchical Models can be used to model complex data structures
- The Bayesian approach offers an easy approach to the estimation of complex models via Markov Chain Monte Carlo
- Spatial analysis of routinely collected health data is standard practice nowadays
- Spatio-temporal models can be used
- Waller & Gotway (2004) and Banerjee et al. (2003) account for a comprehensive summary on spatial models
Bayesian Inference

- Bayesian Inference is based on estimating the probability density of the parameters $\theta$ in the model after observing the data, i.e., their posterior distributions: $p(\theta|y)$
- $p(\theta|y)$ is usually difficult to derive:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int_\theta p(y|\theta)p(\theta)} \propto p(y|\theta)p(\theta)$$

- $p(y|\theta)$ is the likelihood of the model, which reflects the relationship between the data and the parameters
- $p(\theta)$ is the prior distribution of the parameters, which reflects the initial information on the parameters
- Usually, $p(\theta|y)$ is computed by simulation using Markov Chain Monte Carlo techniques
- WinBUGS in a generic software to fit a wide range of models. It uses the Gibbs sampler for that.
Benefits of Bayesian Inference

- Suitable framework to deal with a large number of problems
- Priors can be used to account for initial information (for example, spatial dependence)
- If no prior information is available, vague (or non-informative) priors can be used so that the posterior distribution will only depend on the data and the model.
- Multilevel models can be used: Bayesian Hierarchical Models
- Complex effects, such as spatial and/or temporal dependence, can be modeled easily
- When the posterior distribution is not in a closed form, different simulation techniques can be used to approximate them.
- Missing values are treated similarly as the parameters in the model
Markov Chain Monte Carlo/Gibbs sampler

MCMC aims at simulating a series of values for the parameters in the model, so that, in the end, these values will be draws from the posterior distribution.

- Assign initial values to every parameter in the model (and missing values)
- At every step, Gibbs sampler simulates from the full conditional distribution:

\[ p(\theta_i|\theta_{-i}, y) \]

- After a burn in period, the simulated values are draws from the posterior \( p(\theta|y) \)
- Convergence of the simulated values should be assessed
WinBUGS

- BUGS stands for *Bayesian inference Using Gibbs Sampler*
- Developed at the MRC and Imperial College London
- Provides a generic language to Bayesian Hierarchical models
- Models can be specified graphically as well
- Several utilities to assess the convergence of the chain and display results
- GeoBUGS is an extension to deal with spatial models and maps
- PkBUGS is another extension to deal with Pharmacokinetics models
- A developer interface has been included so that the user can extend the range of functions available
- OpenBUGS is the *open source* version of WinBUGS
Calling WinBUGS from \( R \)

- Packages \texttt{R2WinBUGS} and \texttt{BRUGS} can call WinBUGS and OpenBUGS from \( R \).
- \texttt{R2WinBUGS} calls WinBUGS using the scripting language and then reads the output log file.
- \texttt{BRUGS} is an interface to the actual OpenBUGS (NOT WinBUGS) routines.
- \texttt{R2WinBUGS} can run on several platforms (Windows, Linux/Unix, Mac).
- Other alternatives to call WinBUGS externally in different ways are available at [http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/remote14.shtml](http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/remote14.shtml)
We need...

- Model specification (using the BUGS language)
- Mortality Data (in a list)
- Spatial data describing the neighbourhood structure, in a specific format
- Initial values of the parameters
- Optionally, we may want to export the map information to be used within WinBUGS
Bayesian Spatial Modelling

\( O_i \sim \text{Poisson}(\mu_i) \)
\( \mu_i = \theta_i E_i \)

\[
\log(\theta_i) = \alpha + \beta X_i + u_i + v_i
\]
\( u_i \sim \text{Normal}(0, \sigma_u^2) \)
\( v_i | v_{-i} \sim \text{Normal}(\sum_{j \sim i} v_j / n_i, \sigma_v^2 / n_i) \)

\[
f(\alpha) \propto 1 \quad f(\beta) \propto 1
\]
\( \sigma_u^2 \sim \text{Gamma}^{-1}(0.001, 0.001) \)
\( \sigma_v^2 \sim \text{Gamma}^{-1}(0.001, 0.001) \)

Model specification using the BUGS language

model{
  for(i in 1:N)
  {
    O[i] ~ dpois(mu[i])
    mu[i]<-theta[i] * E[i]
    log(theta[i]) <- alpha + beta[1]*PCTAGE65P[i] +
                   beta[2]*PCTOWNHOME[i] + beta[3]*AVGIDIST[i] + u[i] + v[i]

    u[i] ~ dnorm(0, precu)

    SMR[i]<- O[i] / E[i]
    prob[i]<-step(theta[i]-1)
  }
  v[1:N]~car.normal(adj[], weights[], num[], precv)

  alpha~dflat()
  for(i in 1:3) {beta[i] ~dflat()}
  precu~dgamma(0.001, 0.001)
  precv~dgamma(0.001, 0.001)

  sigmau<-1/precu
  sigmav<-1/precv
}
Preparing data...

1.- Read maps

> library(maptools)
> nymap <- readShapePoly("NY8_utm18")

2.- Create list of observed, expected

> nymap$EXP <- nymap$POP8 * sum(nymap$Cases)/sum(nymap$POP8)

3.- Create adjacency matrix

> library(spdep)
> nynb <- poly2nb(nymap)

4.- Create weights

> nyWBweights <- nb2WB(nynb)

> d <- c(list(O = nymap$Cases, E = nymap$EXP), N = 281,
+       list(PCTAGE65P = nymap$PCTAGE65P, PCTOWNHOME = nymap$PCTOWNHOME,
+             AVGIDIST = nymap$AVGIDIST))
> inits1 <- list(alpha = 1, beta = c(0, 0, 0), u = rep(0, 281),
+                v = rep(0, 281), precu = 1, precv = 1)
> inits2 <- list(alpha = 10, beta = c(1, 1, 1), u = rep(1, 281),
+                v = rep(1, 281), precu = 0.1, precv = 0.1)
Calling WinBUGS using \texttt{R2WinBUGS}

5.- Call WinBUGS

\begin{verbatim}
> library(R2WinBUGS)
> mfile <- paste(getwd(), "/model.txt", sep = "", collapse = "")
> tdir <- paste(getwd(), "/NYoutput", sep = "", collapse = "")
> dir.create(tdir)
> res <- bugs(data = c(d, nyWBweights), inits = list(inits1, inits2),
+ parameters.to.save = c("u", "v", "theta",
+ "prob", "sigmau", "sigmav"), model.file = mfile,
+ working.directory = tdir, n.thin = 3, n.chains = 2,
+ n.iter = 6000, n.burnin = 3000)
\end{verbatim}

6.- Add results to map object

\begin{verbatim}
> nymap$prob <- res$mean$prob
> nymap$theta <- res$mean$theta
> nymap$u <- res$mean$u
> nymap$v <- res$mean$v
> logfile <- paste(getwd(), "/NYoutput/log.txt", sep = "",
+ collapse = "")
> reslog <- bugs.log(file = logfile)
\end{verbatim}
Mapping the results

Smoothed Relative Risks

Probability map
Exporting the data to work directly with WinBUGS

1. Export the maps with spdep

   > sp2WB(map = nymap, file = "NY_WB.txt")

2. Import map with WB first, and "reboot"

3. Use bugs.data (from R2WinBUGS) to create the files with data and initial values

   > bugs.data(d)
   > file.rename("data.txt", "dataNY.txt")
   > bugs.data(nyWBweights)
   > file.rename("data.txt", "data-spatialNY.txt")
   > bugs.data(inits1)
   > file.rename("data.txt", "inits1NY.txt")
   > bugs.data(inits2)
   > file.rename("data.txt", "inits2NY.txt")
Running WinBUGS directly

1. Open all needed files in WinBUGS
2. *Check* the model syntax
3. *Load* data (health and spatial)
4. *Compile* the model
5. *Load* initial values
6. *Run* the model (burn in period)
7. *Monitor* parameters of interest and DIC
8. *Rerun* the model
9. *Assess* convergence of the simulations
10. *Show* summary statistics of the parameters of the model
11. *Display* results on a map
Further references

- OpenBUGS: [http://mathstat.helsinki.fi/openbugs/](http://mathstat.helsinki.fi/openbugs/)
- R programming language: [http://www.r-project.org](http://www.r-project.org)