Practical Data Analysis with JAGS using R

Department of Biostatistics
Institute of Public Health, University of Copenhagen
Sunday 19th August, 2012
INLA-problems

Compiled Sunday 19th August, 2012, 16:33
from: C:/Bendix/undervis/Bayes/Cph.2012/pracs/lji.tex

Lyle Gurrin Centre for MEGA Epidemiology
School of Population Health, University of Melbourne
Carlton, Victoria, Australia
lgurrin@unimelb.edu.au
http://www.sph.unimelb.edu.au/about/contact/allstaff/gurrin

Bendix Carstensen Steno Diabetes Center, Gentofte
& Department of Biostatistics, University of Copenhagen
bxc@steno.dk
http://www.biostat.ku.dk/~bxc

Søren Højsgaard Department of Mathematical Sciences
Aalborg University
sorenh@math.aau.dk
http://people.math.aau.dk/~sorenh/

Claus Ekstrøm Department of Biostatistics, Institute of Public Health,
University of Southern Denmark
cekstrom@health.sdu.dk
http://www.sdu.dk/staff/cekstrom

Contents

Chapter 1

INLA-problem

1.1 Fetal growth

We are interested in describing how head circumference varies by the transformed gestational age, but also in describing how growth of the head circumference varies between fetuses. First we read the data:

```
> fetal <- read.csv("http://BendixCarstensen.com/Bayes/Cph-2012/data/fetal.csv",header=TRUE)
> head( fetal, 11 )
```

```
   id  hc    ga   tga
1   1 211 23.00 16.83
2   1 274 28.43 19.00
3   1 314 33.43 20.39
4   1 330 38.43 21.20
5   2 141 17.71 14.05
6   2 199 22.86 16.76
7   2 266 27.86 18.81
8   2 297 31.29 19.87
9   2 313 34.57 20.63
10  2 321 36.57 20.97
11  3 205 23.43 17.03
```

`ga` is the gestational age, `tga` is a transformation of it which we shall use as covariate, in the following model formulation termed t. The response is head circumference of the fetus, `hc`, in the model description termed y.

For the sake of model fitting we will center the transformed gestational age around 19.5, corresponding to a gestational age of 30 weeks:

```
> fetal$tga <- fetal$tga - 19.5
```

1.2 REML modelling

The relevant model is a *linear mixed model* with a random intercept and a random slope term for the measurement y_{ft} on fetus f at time t:

$$y_{ft} = (\beta_0 + u_{0f}) + (\beta_1 + u_{1f})t + e_{ft}$$
$$(u_{0f}, u_{1f}) \sim \mathcal{N}(0, \Sigma), \quad e_{ft} \sim \mathcal{N}(0, \sigma)$$

Now set up and estimate in this model using e.g. `lmer` from the `lme4`.

```
> library( lme4 )
> m0 <- lmer( hc ~ tga + (tga|id), data=fetal )
> summary(m0)


Linear mixed model fit by REML
Formula: hc ~ tga + (tga | id)
   Data: fetal
   AIC   BIC logLik deviance REMLdev
 23340 23376 -11664    23324   23328
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 id       (Intercept) 74.4922  8.6309
          tga          1.2152  1.1024   0.747
 Residual             73.7748  8.5892
Number of obs: 3097, groups: id, 706

Fixed effects:
             Estimate Std. Error t value
(Intercept) 286.27482    0.37207   769.4
tga          26.48473    0.07786   340.2

Correlation of Fixed Effects:
    (Intr)
tga 0.536
```

Thus we see that the default is to produce a set of correlated random effects, which of course is the only sensible thing to do.

1.3 JAGS

We can also set this model up in `JAGS` the usual way; first we specify data:

```
> fetal.dat <- list( id = as.integer( factor(fetal$id) ),
+                    hc = fetal$hc,
+                   tga = fetal$tga,
+                     N = nrow(fetal),
+                     I = length( unique(fetal$id) ) )
```

In particular we need to specify a variance-covariance for the random effects, which is done by specifying a Wishart prior, which takes a 2×2-matrix as input, which we specify in a `data` section of the JAGS program:

```
> cat("
+ # Fixing data to be used in model definition
+ data
+    {
+    zero[1] <- 0
+    zero[2] <- 0
+    R[1,1] <- 0.1
+    R[1,2] <- 0
+    R[2,1] <- 0
+    R[2,2] <- 0.5
+    }
+ # Then define model
+ model
+    {
+    # Intercept and slope for each person, including random effects
+      for( i in 1:I )
+      {
```

```
+       u[i,1:2] ~ dmnorm(zero,Omega.u)
+       }
+
+     # Define model for each observational unit
+       for( j in 1:N )
+       {
+       mu[j] <- ( beta[1] + u[id[j],1] ) +
+                ( beta[2] + u[id[j],2] ) * ( tga[j]-18 )
+       hc[j] ~ dnorm( mu[j], tau.e )
+       }
+
+     #-------------------------------------------------------------
+     # Priors:
+
+     # Fixed intercept and slope
+       beta[1] ~ dnorm(0.0,1.0E-5)
+       beta[2] ~ dnorm(0.0,1.0E-5)
+
+     # Residual variance
+       tau.e <- pow(sigma.e,-2)
+     sigma.e  ~ dunif(0,100)
+
+     # Define prior for the variance-covariance matrix of the random effects
+       Sigma.u <- inverse(Omega.u)
+       Omega.u  ~ dwish( R, 2 )
+     }",
+       file="fetal.jag" )
```

Then we extract the relevant variances from the `lmer` object and use variante of these as starting values for the MCMC chains

```
> ( sigma.e <- attr(VarCorr(m0),"sc") )
```

```
[1] 8.589226
```

```
> ( Omega.u <- solve( VarCorr(m0)$id ) )
```

```
            (Intercept)        tga
(Intercept)  0.03036677 -0.1775887
tga         -0.17758872  1.8614539
```

```
> ( beta    <- fixef( m0 ) )
```

```
(Intercept)         tga
  286.27482    26.48473
```

```
> fetal.ini <- list( list( sigma.e = sigma.e/3,
+                          Omega.u = Omega.u/3,
+                             beta = beta   /3 ),
+                    list( sigma.e = sigma.e*3,
+                          Omega.u = Omega.u*3,
+                             beta = beta   *3 ),
+                    list( sigma.e = sigma.e/3,
+                          Omega.u = Omega.u*3,
+                             beta = beta   /3 ),
+                    list( sigma.e = sigma.e*3,
+                          Omega.u = Omega.u/3,
+                             beta = beta   *3 ) )
```

Finally, we can the whole thing going:

```
> library( rjags )
> system.time(
+ fetal.mod <- jags.model( file = "fetal.jag",
+                           data = fetal.dat,
+                       n.chains = 4,
+                          inits = fetal.ini,
+                        n.adapt = 10000 )
+             )


Compiling data graph
   Resolving undeclared variables
   Allocating nodes
   Initializing
   Reading data back into data table
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
   Graph Size: 19203

Initializing model

   user  system elapsed
 445.48    0.03  460.42


> system.time(
+ fetal.res <- coda.samples( fetal.mod,
+                                  var = c("beta","sigma.e","Sigma.u"),
+                               n.iter = 10000,
+                                 thin = 20 ) )


   user  system elapsed
 456.17    0.00  474.11


> str( fetal.res )


List of 4
 $ : mcmc [1:500, 1:7] 143243 142238 133752 146925 137496 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:7] "Sigma.u[1,1]" "Sigma.u[2,1]" "Sigma.u[1,2]" "Sigma.u[2,2]" ...
  ..- attr(*, "mcpar")= num [1:3] 10020 20000 20
 $ : mcmc [1:500, 1:7] 791 856 877 839 647 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:7] "Sigma.u[1,1]" "Sigma.u[2,1]" "Sigma.u[1,2]" "Sigma.u[2,2]" ...
  ..- attr(*, "mcpar")= num [1:3] 10020 20000 20
 $ : mcmc [1:500, 1:7] 195182 192375 185925 176922 190153 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:7] "Sigma.u[1,1]" "Sigma.u[2,1]" "Sigma.u[1,2]" "Sigma.u[2,2]" ...
  ..- attr(*, "mcpar")= num [1:3] 10020 20000 20
 $ : mcmc [1:500, 1:7] 8414 7281 7099 8527 7855 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:7] "Sigma.u[1,1]" "Sigma.u[2,1]" "Sigma.u[1,2]" "Sigma.u[2,2]" ...
  ..- attr(*, "mcpar")= num [1:3] 10020 20000 20
 - attr(*, "class")= chr "mcmc.list"
```

```
> summary( fetal.res )


Iterations = 10020:20000
Thinning interval = 20
Number of chains = 4
Sample size per chain = 500

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                  Mean        SD  Naive SE Time-series SE
Sigma.u[1,1] 22083.161 4.341e+04 9.707e+02      2.546e+03
Sigma.u[2,1]  1260.033 2.076e+03 4.642e+01      1.258e+02
Sigma.u[1,2]  1260.033 2.076e+03 4.642e+01      1.258e+02
Sigma.u[2,2]   128.595 2.023e+02 4.525e+00      1.114e+01
beta[1]        706.129 1.351e+02 3.020e+00             NA
beta[2]         26.841 1.132e+01 2.531e-01             NA
sigma.e          8.593 1.429e-01 3.196e-03      3.630e-03

2. Quantiles for each variable:

                 2.5%     25%     50%      75%     97.5%
Sigma.u[1,1] 556.2469 707.085 861.653 7203.935 1.506e+05
Sigma.u[2,1]  21.2227  28.297  35.604 1696.226 7.261e+03
Sigma.u[1,2]  21.2227  28.297  35.604 1696.226 7.261e+03
Sigma.u[2,2]   0.8607   1.200   1.551  215.191 7.293e+02
beta[1]      372.9883 719.351 762.438  765.018 8.405e+02
beta[2]        7.5651  24.381  26.459   26.580 5.359e+01
sigma.e        8.3274   8.493   8.592    8.688 8.884e+00
```

1.4 INLA

The INLA version of the model with uncorrelated random incept and slope (which is a totally daft model) looks like this:

```
> library( INLA )
> fetal <- transform(fetal, tgac=tga)
> system.time(
+ im1 <- inla( hc ~ tga + f(id) + f(tgac), data=fetal )
+           )


   user  system elapsed
   0.55    0.47   26.23


> summary( im1 )


Call:
"inla(formula = hc ~ tga + f(id) + f(tgac), data = fetal)"

Time used:
 Pre-processing    Running inla Post-processing           Total
       0.734375       24.875000        0.500000       26.109375

Fixed effects:
                 mean        sd 0.025quant  0.5quant 0.975quant kld
(Intercept) 286.85413 0.5476637  285.78102 286.85374  287.92974   0
```

```
tga          26.53264 0.1661865   26.20633  26.53275   26.85845   0

Random effects:
Name          Model                  Max KLD
id   IID model
tgac   IID model

Model hyperparameters:
                                        mean      sd        0.025quant
Precision for the Gaussian observations 0.0167223 0.0003827 0.0159047
Precision for id                        0.0221556 0.0057376 0.0146714
Precision for tgac                      0.0526882 0.0087575 0.0373498
                                        0.5quant  0.975quant
Precision for the Gaussian observations 0.0167538 0.0173925
Precision for id                        0.0208955 0.0346340
Precision for tgac                      0.0520810 0.0716678

Expected number of effective parameters(std dev): 679.37(4.893)
Number of equivalent replicates : 4.559

Marginal Likelihood:  -11563.88
Warning: Interpret the marginal likelihood with care if the prior model is improper.
```

```
> names( im1 )
```

```
 [1] "names.fixed"                  "summary.fixed"
 [3] "marginals.fixed"              "summary.lincomb"
 [5] "marginals.lincomb"            "size.lincomb"
 [7] "summary.lincomb.derived"      "marginals.lincomb.derived"
 [9] "size.lincomb.derived"         "mlik"
[11] "cpo"                          "model.random"
[13] "summary.random"               "marginals.random"
[15] "size.random"                  "summary.linear.predictor"
[17] "marginals.linear.predictor"   "summary.fitted.values"
[19] "marginals.fitted.values"      "size.linear.predictor"
[21] "summary.hyperpar"             "marginals.hyperpar"
[23] "internal.summary.hyperpar"    "internal.marginals.hyperpar"
[25] "si"                           "total.offset"
[27] "model.spde2.blc"              "summary.spde2.blc"
[29] "marginals.spde2.blc"          "size.spde2.blc"
[31] "misc"                         "dic"
[33] "mode"                         "neffp"
[35] "joint.hyper"                  "nhyper"
[37] "version"                      "Q"
[39] "graph"                        "cpu.used"
[41] "control.compute"              "control.predictor"
[43] "control.lincomb"              "control.data"
[45] "control.inla"                 "control.results"
[47] "control.fixed"                "control.mode"
[49] "control.expert"               "call"
[51] "family"                       "data"
[53] "formula"                      "inla.call"
[55] "silent"                       "model.matrix"
[57] ".control.defaults"            ".internal"
```

When we want the model with correlated random effects we need the replication trick, we have `Nb` babies:

```
> Nb <- length( unique( fetal$id) )
> N <- nrow(fetal)
> fetal$xid <- fetal$id + max( fetal$id )
```

```
> system.time(
+ im2 <- inla( hc ~ 1 + tga +
+                    f(  id, model = "iid2d", n=2*N ) +
+                    f( xid, tga, copy="id" ),
+               data = fetal )
+              )

   user  system elapsed
   3.50    0.67  305.64

> summary( im2 )

Call:
c("inla(formula = hc ~ 1 + tga + f(id, model = \"iid2d\", n = 2 * ",  "    N) + f(xid, tga, copy = \

Time used:
 Pre-processing    Running inla Post-processing           Total
       0.671875      301.218750        3.500000      305.390625

Fixed effects:
                  mean         sd 0.025quant  0.5quant 0.975quant kld
(Intercept) 286.50406 0.23368219  286.04580 286.50406  286.96243   0
tga          26.53311 0.08864301   26.35935  26.53309   26.70707   0

Random effects:
Name          Model                  Max KLD
id   IID2D model
xid   Copy

Model hyperparameters:
                                          mean       sd         0.025quant
Precision for the Gaussian observations  0.0072312  0.0001931  0.0068583
Precision for id (component 1)           5.7686242  3.6562786  1.3351444
Precision for id (component 2)           4.1634951  2.9475689  0.7125060
Rho1:2 for id                           -0.1292505  0.3332889 -0.6839918
                                         0.5quant   0.975quant
Precision for the Gaussian observations  0.0072290  0.0076161
Precision for id (component 1)           4.9441955 15.0617344
Precision for id (component 2)           3.4698604 11.6804273
Rho1:2 for id                           -0.1608835  0.5639297

Expected number of effective parameters(std dev): 40.13(18.08)
Number of equivalent replicates : 77.18

Marginal Likelihood:  -70553.68
Warning: Interpret the marginal likelihood with care if the prior model is improper.
```

The quantities to comapre ate the precisions from the REML fit:

```
> fixef( m0 )

(Intercept)         tga
  286.27482    26.48473

> 1/attr(VarCorr(m0),"sc")^2

[1] 0.01355476

> solve( VarCorr(m0)$id )

            (Intercept)        tga
(Intercept)  0.03036677 -0.1775887
tga         -0.17758872  1.8614539
```

so even though we got the means right, the covariance structe is not what we expected…