Applied Bayesian Non- and Semi-parametric Inference using DPpackage

by Alejandro Jara

Introduction

In many practical situations, a parametric model cannot be expected to describe in an appropriate manner the chance mechanism generating an observed dataset, and unrealistic features of some common models could lead to unsatisfactory inferences. In these cases, we would like to relax parametric assumptions to allow greater modeling flexibility and robustness against misspecification of a parametric statistical model. In the Bayesian context such flexible inference is typically achieved by models with infinitely many parameters. These models are usually referred to as Bayesian Nonparametric (BNP) or Semiparametric (BSP) models depending on whether all or at least one of the parameters is infinity dimensional (Müller & Quintana, 2004).

While BSP and BNP methods are extremely powerful and have a wide range of applicability within several prominent domains of statistics, they are not as widely used as one might guess. At least part of the reason for this is the gap between the type of software that many applied users would like to have for fitting models and the software that is currently available. The most popular programs for Bayesian analysis, such as **BUGS** (Gilks et al., 1992), are generally unable to cope with nonparametric models. The variety of different BSP and BNP models is huge; thus, building for all of them a general software package which is easy to use, flexible, and efficient may be close to impossible in the near future.

This article is intended to introduce an R package, **DPpackage**, designed to help bridge the previously mentioned gap. Although its name is motivated by the most widely used prior on the space of the probability distributions, the Dirichlet Process (DP) (Ferguson, 1973), the package considers and will consider in the future other priors on functional spaces. Currently, **DPpackage** (version 1.0-5) allows the user to perform Bayesian inference via simulation from the posterior distributions for models considering DP, Dirichlet Process Mixtures (DPM), Polya Trees (PT), Mixtures of Triangular distributions, and Random Bernstein Polynomials priors. The package also includes generalized additive models considering penalized B-Splines. The rest of the article is organized as follows. We first discuss the general syntax and design philosophy of the package. Next, the main features of the package and some illustrative examples are presented. Comments on future developments conclude the article.

Design philosophy and general syntax

The design philosophy behind **DPpackage** is quite different from that of a general purpose language. The most important design goal has been the implementation of model-specific MCMC algorithms. A direct benefit of this approach is that the sampling algorithms can be made dramatically more efficient.

Fitting a model in **DPpackage** begins with a call to an R function that can be called, for instance, `DPmodel` or `PTmodel`. Here “model" denotes a descriptive name for the model being fitted. Typically, the model function will take a number of arguments that govern the behavior of the MCMC sampling algorithm. In addition, the model(s) formula(s), data, and prior parameters are passed to the model function as arguments. The common elements in any model function are:

i) `prior`: an object list which includes the values of the prior hyperparameters.

ii) `mcmc`: an object list which must include the integers `nburn` giving the number of burn-in scans, `nskip` giving the thinning interval, `nsave` giving the total number of scans to be saved, and `ndisplay` giving the number of saved scans to be displayed on screen: the function reports on the screen when every `ndisplay` scans have been carried out and returns the process's runtime in seconds. For some specific models, one or more tuning parameters for Metropolis steps may be needed and must be included in this list. The names of these tuning parameters are explained in each specific model description in the associated help files.

iii) `state`: an object list giving the current values of the parameters, when the analysis is the continuation of a previous analysis, or giving the starting values for a new Markov chain, which is useful for running multiple chains starting from different points.

iv) `status`: a logical variable indicating whether it is a new run (`TRUE`) or the continuation of a previous analysis (`FALSE`). In the latter case the current values of the parameters must be specified in the object state.

Inside the R model function the inputs to the model function are organized in a more useable form, the MCMC sampling is performed by calling a shared library written in a compiled language, and the posterior sample is summarized, labeled, assigned into an output list, and returned. The output list includes:

i) `state`: a list of objects containing the current values of the parameters.

ii) `save.state`: a list of objects containing the MCMC samples for the parameters. This list contains two matrices `randsave` and `thetasave` which contain the MCMC samples of the variables with random distribution (errors, random effects, etc.) and the parametric part of the model, respectively.

In order to exemplify the extraction of the output elements, consider the abstract model fit:

```
fit <- DPmodel(..., prior, mcmc,
                    state, status, ....)
```

The lists can be extracted using the following code:

```
fit$state
fit$save.state$randsave
fit$save.state$thetasave
```

Based on these output objects, it is possible to use, for instance, the **boa** (Smith, 2007) or the **coda** (Plummer et al., 2006) R packages to perform convergence diagnostics. For illustration, we consider the **coda** package here. It requires a matrix of posterior draws for relevant parameters to be saved as an `mcmc` object. As an illustration, let us assume that we have obtained `fit1`, `fit2`, and `fit3`, by independently running a model function three times, specifying different starting values each time. To compute the Gelman-Rubin convergence diagnostic statistic for the first parameter stored in the `thetasave` object, the following commands may be used,

```
library("coda")
chain1 <- mcmc(fit1$save.state$thetasave[,1])
chain2 <- mcmc(fit2$save.state$thetasave[,1])
chain3 <- mcmc(fit3$save.state$thetasave[,1])
coda.obj <- mcmc.list(chain1 = chain1,
                      chain2 = chain2,
                      chain3 = chain3)
gelman.diag(coda.obj, transform = TRUE)
```

where the fifth command saves the results as an object of class `mcmc.list`, and the sixth command computes the Gelman-Rubin statistic from these three chains.

Generic R functions such as `print`, `plot`, `summary`, and `anova` have methods to display the results of the **DPpackage** model fit. The function `print` displays the posterior means of the parameters in the model, and `summary` displays posterior summary statistics (mean, median, standard deviation, naive standard errors, and credibility intervals). By default, the function `summary` computes the 95% HPD intervals using the Monte Carlo method proposed by Chen & Shao (1999). Note that this approximation is valid when the true posterior distribution is symmetric. The user can display the order statistic estimator of the 95% credible interval by using the following code,

```
summary(fit, hpd=FALSE)
```

The `plot` function displays the trace plots and a kernel-based estimate of the posterior distribution for the model parameters. Similarly to `summary`, the `plot` function displays the 95% HPD regions in the density plot and the posterior mean. The same plot but considering the 95% credible region can be obtained by using,

```
plot(fit, hpd=FALSE)
```

The `anova` function computes simultaneous credible regions for a vector of parameters from the MCMC sample using the method described by Besag et al. (1995). The output of the `anova` function is an ANOVA-like table containing the pseudo-contour probabilities for each of the factors included in the linear part of the model.

Implemented Models

Currently **DPpackage** (version 1.0-5) contains functions to fit the following models:

i) Density estimation: `DPdensity`, `PTdensity`, `TDPdensity`, and `BDPdensity` using DPM of normals, Mixtures of Polya Trees (MPT), Triangular-Dirichlet, and Bernstein-Dirichlet priors, respectively. The first two functions allow uni- and multi-variate analysis.

ii) Nonparametric random effects distributions in mixed effects models: `DPlmm` and `DPMlmm`, using a DP/Mixtures of DP (MDP) and DPM of normals prior, respectively, for the linear mixed effects model. `DPglmm` and `DPMglmm`, using a DP/MDP and DPM of normals prior, respectively, for generalized linear mixed effects models. The families (links) implemented by these functions are binomial (logit, probit), poisson (log) and gamma (log). `DPolmm` and `DPMolmm`, using a DP/MDP and DPM of normals prior, respectively, for the ordinal-probit mixed effects model.

iii) Semiparametric IRT-type models: `DPrasch` and `FPTrasch`, using a DP/MDP and finite PT (FPT)/MFPT prior for the Rasch model with a binary distribution, respectively. DPraschpoisson and FPTraschpoisson, employing a Poisson distribution.

iv) Semiparametric meta-analysis models: DPmeta and DPMmeta for the random (mixed) effects meta-analysis models, using a DP/MDP and DPM of normals prior, respectively.

v) Binary regression with nonparametric link: CSDPbinary, using Newton et al. (1996)'s centrally standardized DP prior. DPbinary and FPTbinary, using a DP and a finite PT prior for the inverse of the link function, respectively.

vi) AFT model for interval-censored data: DPsurvint, using a MDP prior for the error distribution.

vii) ROC curve estimation: DProc, using DPM of normals.

viii) Median regression model: PTlm, using a median-0 MPT prior for the error distribution.

ix) Generalized additive models: PSgam, using penalized B-Splines.

Additional tools included in the package are DPelicit, to elicit the DP prior using the exact and approximated formulas for the mean and variance of the number of clusters given the total mass parameter and the number of subjects (see, Jara et al. 2007); and PsBF, to compute the Pseudo-Bayes factors for model comparison.

Examples

Bivariate Density Estimation

As an illustration of bivariate density estimation using DPM normals (DPdensity) and MPT models (PTdensity), part of the dataset in Chambers et al. (1983) is considered. Here, $n = 111$ bivariate observations $\mathbf{y}_i = (y_{i1}, y_{i2})^T$ on radiation y_{i1} and the cube root of ozone concentration y_{i2} are modeled. The original dataset has the additional variables wind speed and temperature. These were analyzed by Müller et al. (1996) and Hanson (2006).

The DPdensity function considers the multivariate extension of the univariate Dirichlet Process Mixture of Normals model discussed in Escobar & West (1995),

$$y_i \mid G \overset{iid}{\sim} \int N_k(\boldsymbol{\mu}, \boldsymbol{\Sigma})\, G(d\boldsymbol{\mu}, d\boldsymbol{\Sigma})$$

$$G \mid M, G_0 \sim DP(\alpha G_0)$$

$$G_0 \equiv N_k(\boldsymbol{\mu} \mid \boldsymbol{m}_1, \kappa_0^{-1}\boldsymbol{\Sigma}) IW_k(\boldsymbol{\Sigma} \mid \nu_1, \boldsymbol{\Psi}_1)$$

$$\alpha \sim \Gamma(a_0, b_0)$$

$$\boldsymbol{m}_1 \mid \boldsymbol{m}_2, \boldsymbol{S}_2 \sim N_k(\boldsymbol{m}_2, \boldsymbol{S}_2)$$

$$\kappa_0 \mid \tau_1, \tau_2 \sim \Gamma(\tau_1/2, \tau_2/2)$$

$$\boldsymbol{\Psi}_1 \mid \nu_2, \boldsymbol{\Psi}_2 \sim IW_k(\nu_2, \boldsymbol{\Psi}_2)$$

where $N_k(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ refers to a k-variate normal distribution with mean and covariance matrix $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, respectively, $IW_k(\nu, \boldsymbol{\Psi})$ refers to an inverted-Wishart distribution with shape and scale parameter ν and $\boldsymbol{\Psi}$, respectively, and $\Gamma(a, b)$ refers to a gamma distribution with shape and rate parameter, a and b, respectively. Note that the inverted-Wishart prior is parameterized such that its mean is given by $\frac{1}{\nu-k-1}\boldsymbol{\Psi}^{-1}$.

The PTdensity function considers a Mixture of multivariate Polya Trees model discussed in Hanson (2006),

$$y_i | G \overset{iid}{\sim} G, \tag{1}$$

$$G \mid \alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma}, M \sim PT^M(\Pi^{\boldsymbol{\mu},\boldsymbol{\Sigma}}, \mathcal{A}^{\alpha}), \tag{2}$$

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto |\boldsymbol{\Sigma}|^{-(d+1)/2}, \tag{3}$$

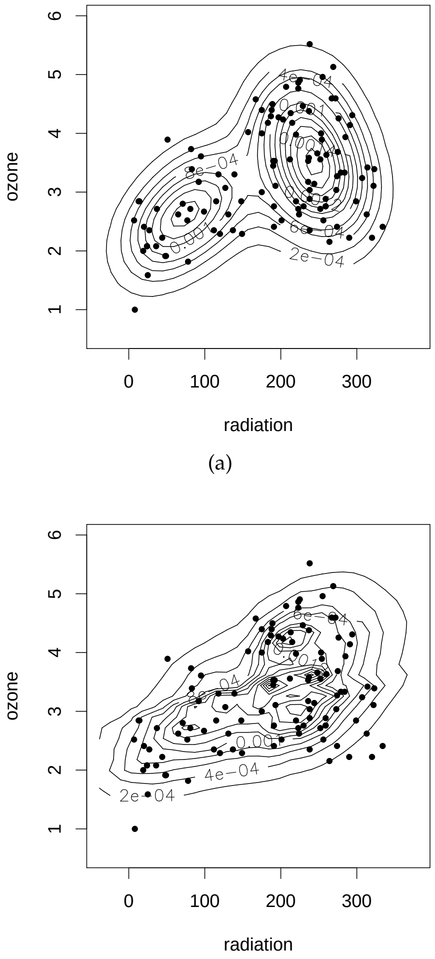
$$\alpha | a_0, b_0 \sim \Gamma(a_0, b_0), \tag{4}$$

where the PT prior is centered around a $N_k(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ distribution. To fit these models we used the following commands:

```
# Data
  data("airquality")
  attach(airquality)
  ozone <- Ozone**(1/3)
  radiation <- Solar.R

# Prior information
  priorDPM <- list(a0 = 1, b0 = 1/5,
    nu1 = 4, nu2 = 4,
    s2 = matrix(c(10000,0,0,1),ncol = 2),
    m2 = c(180,3),
    psiinv2 = matrix(c(1/10000,0,0,1),ncol = 2),
    tau1 = 0.01, tau2 = 0.01)

  priorMPT <- list(a0 = 5, b0 = 1, M = 4)

# MCMC parameters
  mcmcDPM <- list(nburn = 5000, nsave = 20000,
                  nskip = 20, ndisplay = 1000)

  mcmcMPT <- list(nburn = 5000, nsave = 20000,
                  nskip = 20, ndisplay = 1000,
                  tune1 = 0.025, tune2 = 1.1,
                  tune3 = 2.1)

# Fitting the models
  fitDPM <- DPdensity(y = cbind(radiation,ozone),
                prior = priorDPM,mcmc = mcmcDPM,
                state = NULL,status = TRUE,
                na.action = na.omit)
```

```
fitMPT <- PTdensity(
                y = cbind(radiation,ozone),
                prior = priorMPT,mcmc = mcmcMPT,
                state = NULL,status = TRUE,
                na.action = na.omit)
```

We illustrate the results from these analyses in Figure 1. This figure shows the contour plots of the posterior predictive density for each model.



(a)

(b)

Figure 1: Density estimate for the New York Air Quality Measurements dataset, using (a) `DPdensity` and (b) `PTdensity`, respectively.

Figure 1 clearly shows a departure from the normality assumption for these data. The results indicate the existence of at least two clusters of data. We refer to Hanson (2006) for more details and comparisons between these models.

Interval-Censored Data

The `DPsurvint` function implements the algorithm described by Hanson & Johnson (2004) for semiparametric accelerated failure time (AFT) models. We illustrate the function on a dataset involving time to cosmetic deterioration of the breast for women with stage 1 breast cancer who have undergone a lumpectomy, for two treatments, these being radiation, and radiation coupled with chemotherapy. Radiation is known to cause retraction of the breast, and there is some evidence that chemotherapy worsens this effect. There is interest in the cosmetic impact of the treatments because both are considered very effective in preventing recurrence of this early stage cancer.

The data come from a retrospective study of 46 patients who received radiation only and 48 who received radiation plus chemotherapy. Patients were observed typically every four to six months and at each observation a clinician recorded the level of breast retraction that had taken place since the last visit: none, moderate, or severe. The time-to-event considered was the time until moderate or severe breast retraction, and this time is interval censored between patient visits or right censored if no breast retraction was detected over the study period of 48 months. As the observed intervals were the result of pre-scheduled visits, an independent noninformative censoring process can be assumed. The data were analyzed by Hanson & Johnson (2004) and also given in Klein & Moeschberger (1997).

In the analysis of survival data with covariates, the semiparametric proportional hazards (PH) model is the most popular choice. It is flexible and easily fitted using standard software, at least for right-censored data. However, the assumption of proportional hazard functions may be violated and we may seek a proper alternative semiparametric model. One such model is the AFT model. Whereas the PH model assumes the covariates act multiplicatively on a baseline hazard function, the AFT model assumes that the covariates act multiplicatively on the argument of the baseline survival distribution, G, $P(T > t \mid \boldsymbol{x}) = G\left((t\exp\{\boldsymbol{x}_i^T\boldsymbol{\beta}\}, +\infty)\right)$, thus providing a model with a simple interpretation of the regression coefficients for practitioners.

Classical treatments of the semiparametric AFT model with interval-censored data were presented, for instance, in Lin & Zhang (1998). Note, however, that for semiparametric AFT models there is nothing comparable to a partial likelihood function. Therefore, the vector of regression coefficients and the baseline survival distribution must be estimated simultaneously, complicating matters enormously in the interval-censored case. The more recent classical approaches only provide inferences about the regression coefficients and not for the survival function.

In the Bayesian semiparametric context, Christensen & Johnson (1998) assigned a simple DP prior, centered in a single distribution, to baseline survival for nested interval-censored data. A marginal likelihood for the vector of regression coefficients $\boldsymbol{\beta}$ is maximized to provide a point estimate and resulting survival curves. However, this approach does not allow the computation of credible intervals for the parameters. Moreover, it may be difficult in practice to specify a single centering distribution for the DP prior and, once specified, a single centering distribution may affect inferences. To overcome these difficulties, a MDP prior can be considered. Under this approach, it is not very difficult to demonstrated that the computations involved for a full Bayesian solution are horrendous at best, even for the non-censored data problem. The analytic intractability of the Bayesian semiparametric AFT model has been overcome using MCMC methods by Hanson & Johnson (2004).

To test whether chemotherapy in addition to radiotherapy has an effect on time to breast retraction, an AFT model $T_i = \exp(-\mathbf{x}_i^T\boldsymbol{\beta})V_i$, $i = 1,\ldots,n$, was considered. We model the baseline distribution in the AFT model using a MDP prior centered in a standard parametric family, the lognormal distribution,

$$V_1,\ldots,V_n|G \overset{iid}{\sim} G,$$

$$G \mid \alpha,\mu,\sigma^2 \sim DP\left(\alpha G_0\right), \quad G_0 \equiv LN(\mu,\sigma^2),$$

$$\mu \mid m_0, s_0 \sim N\left(m_0, s_0\right),$$

$$\sigma^{-2} \mid \tau_1,\tau_2 \sim \Gamma\left(\tau_1/2,\tau_2/2\right),$$

$$\boldsymbol{\beta} \mid \boldsymbol{\beta}_0, \boldsymbol{S}_{\boldsymbol{\beta}_0} \sim N_p\left(\boldsymbol{\beta}_0, \boldsymbol{S}_{\boldsymbol{\beta}_0}\right),$$

where $LN\left(m,s^2\right)$ and $N\left(m,s^2\right)$ refer to a log-normal and normal distribution, respectively, with location m and scale parameter s^2. The precision parameter of the MDP prior was chosen to be $\alpha = 10$, allowing for moderate deviations from the log-normal family. We allow the parametric family to hold only approximately, and the resulting model is robust against mis-specification of the baseline survival distribution. The covariate of interest is $\text{trt}_i = 0$ if the ith patient had radiotherapy only and $\text{trt}_i = 1$ if the ith patient had radiotherapy and chemotherapy. The following commands were used to fit the model,

```
# Data
  data("deterioration")
  attach(deterioration)
  y <- cbind(left,right)

# MCMC parameters
  mcmc <- list(nburn = 20000, nsave = 10000,
               nskip = 20, ndisplay = 1000,
               tune = 0.25)

# Prior information
  prior <- list(alpha = 10, beta0 = rep(0,1),
             Sbeta0 = diag(100,1), m0 = 0,
             s0 = 1, tau1 = 0.01, tau2 = 0.01)

# Fitting the model
  fit <- DPsurvint(y ~ trt, prior = prior,
                   mcmc = mcmc, state = NULL,
                   status = TRUE)
```

In our analysis, the posterior mean and 95% HPD associated with trt was 0.50 (0.12, 0.82), indicating that including chemotherapy significantly reduces the time to deterioration. Figure 2 (page 22) displays posterior summary statistics for the parameters of interest. In this case, the output includes the log of the Conditional Predictive Ordinate (CPO) (see, Geisser & Eddy 1979) for each data point, the AFT regression coefficients, the parameters of the DP centering distribution, and the number of clusters.

Inferences about the survival curves can be obtained from the MCMC output. Indeed, given a sample of the parameters of size J, a sample of the survival curve for a given $\mathbf{x}$ can be drawn as follows: for the MCMC scan j of the posterior distribution, with $j = 1,\ldots,J$, we sample from $S^{(j)}(t|\mathbf{x},\text{data}) \sim Beta(a^{(j)}(t), b^{(j)}(t))$ where $a^{(j)}(t) = \alpha^{(j)}G_0^{(j)}\left((t\exp(\mathbf{x}^T\boldsymbol{\beta}^{(j)}),+\infty)\right) + \sum_{i=1}^n \delta_{V_i^{(j)}}\left((t\exp(\mathbf{x}^T\boldsymbol{\beta}^{(j)}),+\infty)\right)$, and $b^{(j)}(t) = \alpha^{(j)} + N - a^{(j)}(t)$. This approach is implemented in the function predict.DPsurvint. For user-specified values of the covariates, xnew, and the grid where the survival function is evaluated, grid, posterior information for the survival curves can be obtained using the following commands,

```
xnew <- matrix(c(0,1), nrow=2, ncol=1)
grid <- seq(0.01,70,1)
pred <- predict(fit, xnew=xnew, grid=grid)
plot(pred, all=FALSE, band=TRUE)
```

The resulting survival curves and point-wise 95% HPD intervals are given in Figure 3 (page 23).

Semiparametric Generalized Linear Mixed Model

Lesaffre & Spiessens (2001) analyzed data from a multicentre randomized comparison of two oral treatments for toe-nail infection (dermatophyte onychomycosis) involving two groups of 189 patients evaluated at seven visits; on week 0, 4, 8, 12, 24, 36, and 48. Onychomycosis, known popularly as toenail fungus, is a fairly common condition that not only can disfigure and sometimes destroy the nail but that also can lead to social and self-image issues for sufferers. Onychomycosis can be caused by several types of fungi known as dermatophytes, as well as by non-dermatophytic yeasts or molds. Dermatophyte onychomycosis corresponds to the type caused by dermatophytes. Here we are interested in the degree of onycholysis which expresses the degree of separation of the nail plate from the nail-bed and which was scored in four categories (0, absent; 1, mild; 2, moderate; 3, severe). These data were analyzed by Lesaffre & Spiessens (2001) using generalized estimating equations (GEE) and generalized linear mixed models (GLMM).

```
> summary(fit)
Bayesian Semiparametric AFT Regression Model

Call:
DPsurvint.default(formula = y ~ trt, prior = prior, mcmc = mcmc,
    state = state, status = TRUE)

Posterior Predictive Distributions (log):
   Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-4.5920  -2.3570  -1.4600  -1.6240  -0.7121  -0.1991

Regression coefficients:
     Mean      Median    Std. Dev.  Naive Std.Error  95%HPD-Low  95%HPD-Upp
trt  0.502282  0.513219  0.195521   0.001955         0.120880    0.820614

Baseline distribution:
        Mean      Median    Std. Dev.  Naive Std.Error  95%HPD-Low  95%HPD-Upp
mu      3.255374  3.255518  0.173132   0.001731         2.917770    3.589759
sigma2  1.021945  0.921764  0.469061   0.004691         0.366900    1.908676

Precision parameter:
          Mean      Median    Std. Dev.  Naive Std.Error  95%HPD-Low  95%HPD-Upp
ncluster  27.58880  28.00000   3.39630    0.03396          20.00000    33.00000

Acceptance Rate for Metropolis Step =  0.2637435

Number of Observations: 94
```

Figure 2: Posterior summary for the Breast Cancer Data fit using DPsurvint.

GLMM provide a popular framework for the analysis of longitudinal measures and clustered data. The models account for correlation among clustered observations by including random effects in the linear predictor component of the model. Although GLMM fitting is typically complex, standard random intercept and random intercept/slope models with normally distributed random effects can now be routinely fitted in standard software. Such models are quite flexible in accommodating heterogenous behavior, but they suffer from the same lack of robustness against departures from distributional assumptions as other statistical models based on Gaussian distributions.

A common strategy for guarding against such mis-specification is to build more flexible distributional assumptions for the random effects into the model. Following Lesaffre & Spiessens (2001), we consider a logistic mixed effects model to examine the probability of moderate or severe toenail separation $Y = 1$ versus the probability of absent or mild $Y = 0$, including as covariates treatment (trt) (0 or 1), time (t) (continuous), and time×treatment interaction,

$$\begin{aligned}\operatorname{logit}\left\{P\left(Y_{ij}=1 \mid \boldsymbol{\beta}, \theta_i\right)\right\} &= \theta_i + \beta_1 \mathrm{Trt}_i + \beta_2 \mathrm{Time}_{ij} + \\ &\quad \beta_3 \mathrm{Trt}_i \times \mathrm{Time}_{ij}.\end{aligned}$$

However, we replace the normality assumption of the random intercepts by using a DPM of normals prior (see, e.g., Müller et al. 2007),

$$\boldsymbol{\theta}_i \mid G \overset{iid}{\sim} G,$$

$$G \mid P, \boldsymbol{\Sigma}_k \sim \int N(\boldsymbol{m}, \boldsymbol{\Sigma}_k) P(d\boldsymbol{m}),$$

$$P \mid \alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma} \sim DP\left(\alpha N(\boldsymbol{\mu}, \boldsymbol{\Sigma})\right),$$

$$\boldsymbol{\beta} \sim N_p\left(\boldsymbol{\beta}_0, \boldsymbol{S}_{\boldsymbol{\beta}_0}\right),$$

$$\boldsymbol{\Sigma}_k \mid \nu_0, \boldsymbol{T} \sim IW_k\left(\nu_0, \boldsymbol{T}\right),$$

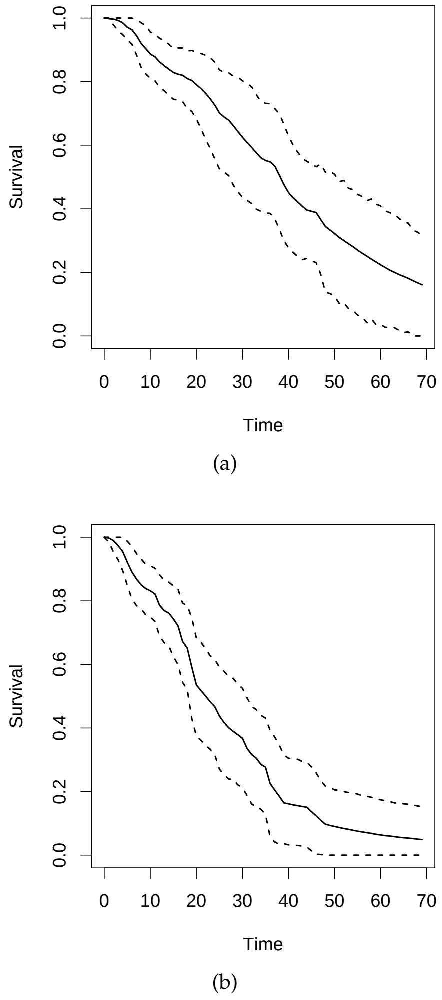
$$\boldsymbol{\mu} \mid \boldsymbol{m}_b, \boldsymbol{S}_b \sim N_q\left(\boldsymbol{m}_b, \boldsymbol{S}_b\right),$$

$$\boldsymbol{\Sigma} \mid \nu_0, \boldsymbol{T}_b \sim IW_k\left(\nu_b, \boldsymbol{T}_b\right),$$

$$\alpha \mid a_0, b_0 \sim \Gamma\left(a_0, b_0\right).$$

The semiparametric GLMM using DPM of normals model can be fitted using function DPMglmm and the following code,

```
# MCMC parameters
  mcmc <- list(nburn = 20000, nsave = 20000,
               nskip = 50, ndisplay = 1000)

# Prior information
  prior <- list(a0 = 2.01, b0 = 0.01,
                nu0 = 2.05, tinv = diag(0.02,1),
                nub = 2.05, tbinv = diag(0.02,1),
                mb = rep(0,1), Sb = diag(100,1),
                beta0 = rep(0,3),
           Sbeta0 = diag(100,3))

# Fitting the model
  fitDPM <- DPMglmm(fixed = infect~trt+
            time*trt, random = ~ 1|idnr,
            family = binomial(logit),
            prior = prior, mcmc = mcmc,
            state = NULL, status = TRUE)
```



(a)

(b)

Figure 3: Breast cancer data: Posterior probability of no evidence of breast retraction for (a) radiotherapy only and (b) radiotherapy plus chemotherapy, respectively.

Figure 4 shows the posterior estimate of the random effects distribution. The predictive density is overlaid on a plot of the posterior means of the random effects. The results clearly indicate departure from the normality assumption, suggesting the existence of groups of patients with respect to the resistance against the infection.

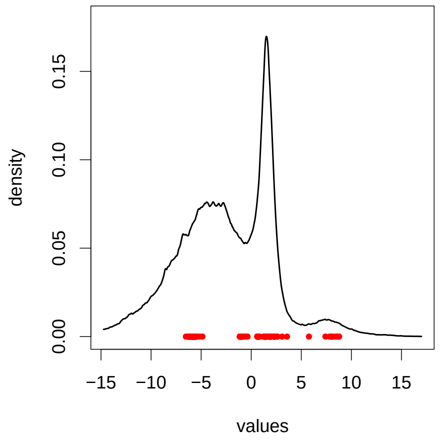


Figure 4: Toe-nail data: Random effects distribution and posterior means estimated using DPMglmm.

Figure 5 (page 24) reports summary statistics for the posterior distribution of the parameters of interest. It includes posterior means and 95% HPD intervals for the parameters along with two model performance measures: DIC and LPML. DIC is the deviance information criterion of Spiegelhalter et al. (2002). LPML is the log pseudo marginal likelihood of Geisser & Eddy (1979), which is a leave-one-out cross validatory measure based on predictive densities. A parametric analysis of these data (not shown), considering equivalent prior distributions, gave a DIC and LPML of 964.2 and -484.0, respectively. The results, therefore, indicate that the DPM version of the model outperforms the normal model using either the LPML or DIC statistic, suggesting that the semiparametric model is better both for explaining the observed data and from a predictive point of view.

Figure 5 (page 24) and the Pseudo Contour probabilities of the covariates in the model (see below) suggest a significant effect of time on the degree of toe-nail infection. As expected because of randomization of the patients to the treatment groups, no significant difference between the two treatment groups at baseline was observed. The results also suggest a non-significant difference in the evolution in time between the treatment groups, contradicting the results under the parametric normal model. The posterior mean (95% HPD interval) for β_3 (Trt × Time) under the normal assumption for the random effects was -0.138 (-0.271; -0.005). These results illustrate the consequences of the incorrect use of traditional model assumptions in the GLMM context.

```
> anova(fitDPM)
Table of Pseudo Contour Probabilities

Response: infect
         Df  PsCP
trt       1 0.512
time      1 <0.01 ***
```

```
> summary(fitDPM)

Bayesian semiparametric generalized linear mixed effect model

Call:
DPMglmm.default(fixed = infect ~ trt + time * trt, random = ~1 |
    idnr, family = binomial(logit), prior = prior, mcmc = mcmc,
    state = state, status = TRUE)

Posterior Predictive Distributions (log):
      Min.     1st Qu.      Median        Mean     3rd Qu.        Max.
-9.644e+00  -2.335e-01  -4.190e-02  -2.442e-01  -8.629e-03  -4.249e-05

Model's performance:
   Dbar    Dhat     pD      DIC     LPML
    753.0   603.6   149.4   902.5  -466.0

Regression coefficients:
             Mean       Median     Std. Dev.  Naive Std.Error  95%HPD-Low  95%HPD-Upp
(Intercept)  -2.508419  -2.440589   0.762218   0.005390        -4.122867   -1.091684
trt           0.300309   0.304453   0.478100   0.003381        -0.669604    1.242553
time         -0.392343  -0.390384   0.046101   0.000326        -0.482329   -0.302442
trt:time     -0.128891  -0.128570   0.072272   0.000511        -0.265813    0.018636

Kernel variance:
                   Mean       Median     Std. Dev.  Naive Std.Error  95%HPD-Low  95%HPD-Upp
sigma-(Intercept)  0.0318682  0.0130737  0.0966504  0.0006834        0.0009878   0.1069456

Baseline distribution:
                    Mean       Median     Std. Dev.  Naive Std.Error  95%HPD-Low  95%HPD-Upp
mub-(Intercept)     -2.624227  -2.558427   1.405269   0.009937        -5.621183    0.008855
sigmab-(Intercept)  26.579978  23.579114  13.640300   0.096451         7.714973   52.754246

Precision parameter:
          Mean      Median    Std. Dev.  Naive Std.Error  95%HPD-Low  95%HPD-Upp
ncluster   70.6021   64.0000   36.7421     0.2598          11.0000     143.0000
alpha      38.4925   25.7503   44.1123     0.3119           1.1589     112.1120

Acceptance Rate for Metropolis Steps =  0.8893615 0.9995698

Number of Observations: 1908
Number of Groups: 294
```

Figure 5: Posterior summary for the Toe-nail Data fit using DPMglmm.

```
trt:time  1 0.075 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01
'*' 0.05 '.' 0.1 ' ' 1
```

Finally, information about the posterior distribution of the subject-specific effects can be obtained by using the DPMrandom function as follows,

```
> DPMrandom(fitDPM)

Random effect information for the DP object:

Call:
DPMglmm.default(fixed = infect ~ trt +
    time * trt, random = ~1 |idnr,
    family = binomial(logit),
    prior = prior, mcmc = mcmc,
    state = state, status = TRUE)

Posterior mean of subject-specific components:

     (Intercept)
1      1.6239
        .
        .
383   2.0178
```

Summary and Future Developments

As the main obstacle for the practical use of BSP and BNP methods has been the lack of estimation tools, we presented an R package for fitting some frequently used models. Until the release of **DPpackage**, the two options for researchers who wished to fit a BSP or BNP model were to write their own code or to rely heavily on particular parametric approximations to some specific processes using the **BUGS** code given in Peter Congdon's books (see, e.g., Congdon 2001). **DPpackage** is geared primarily towards users who are not willing to bear the costs associated with both of these options.

Many improvements to the current status of the package can be made. For example, all **DPpackage** modeling functions compute CPOs for model comparison. However, only some of them compute the effective number of parameters pD and the deviance information criterion (DIC), as presented by Spiegelhalter et al. (2002). These and other model comparison criteria will be included for all the functions in future versions of **DPpackage**.

The implementation of more models, the development of general-purpose sampling functions, and the ability to run several Markov chains at once and to handle large dataset problems through the use of sparse matrix techniques, are areas of further improvement.

Acknowledgments

I wish to thank (in alphabetical order) Timothy Hanson, George Karabatsos, Emmanuel Lesaffre, Peter Müller, and Fernando Quintana for many valuable discussions, comments, and suggestions during the development of the package. Timothy Hanson and Fernando Quintana are also co-authors of the functions `PTdensity` and `FPTbinary`, and `BDPdensity`, respectively. I gratefully acknowledge the partial support of the KUL-PUC bilateral (Belgium-Chile) grant No BIL05/03 and of the IAP research network grant No P6/03 of the Belgian government (Belgian Science Policy). The author thanks Novartis, Belgium, for permission to use their dermatological data for statistical research.

Bibliography

J. Besag, P. Green, D. Higdon, and K. Mengersen. Bayesian computation and stochastic systems (with Discussion). *Statistical Science*, 10:3–66, 1995.

J. M. Chambers, S. Cleveland, and A. P. Tukey. Graphical Methods for Data Analysis. *Boston, USA: Duxbury*, 1983.

M. H. Chen and Q. M. Shao. Monte Carlo estimation of Bayesian credible and HPD intervals. *Journal of Computational and Graphical Statistics*, 8(1):69–92, 1999.

R. Christensen and W. O. Johnson. Modeling Accelerated Failure Time With a Dirichlet Process. *Biometrika*, 75:693–704, 1998.

P. Congdon. *Bayesian Statistical Modelling*. New York, USA: John Wiley and Sons, 2001.

M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.

T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1:209–230, 1973.

S. Geisser and W. Eddy. A predictive approach to model selection. *Journal of the American Statistical Association*, 74:153–160, 1979.

W. R. Gilks, A. Thomas, and D. J. Spiegelhalter. Software for Gibbs sampler *Computing Science and Statistics* 24:439–448, 1992.

T. Hanson. Inference for Mixtures of Finite Polya Tree Models. *Journal of the American Statistical Association*, 101:1548–1565, 2006.

T. Hanson and W. O. Johnson. A Bayesian Semiparametric AFT Model for Interval-Censored Data. *Journal of Computational and Graphical Statistics* 13(2):341–361, 2004.

A. Jara, M. J. Garcia-Zattera, and E. Lesaffre. A Dirichlet Process Mixture model for the analysis of correlated binary responses. *Computational Statistics and Data Analysis*, 51:5402–5415, 2007.

J. Klein and M. Moeschberger. *Survival Analysis*. New York, USA: Springer-Verlag, 1997.

E. Lesaffre and B. Spiessens. On the effect of the number of quadrature points in a logistic random-effects model: an example. *Applied Statistics* 50: 325–335, 2001.

G. Lin and C. Zhang. Linear Regression With Interval Censored Data. *The Annals of Statistics* 26:1306–1327, 1998.

P. Müller and F.A. Quintana. Nonparametric Bayesian Data Analysis. *Statistical Science* 19(1):95–110, 2004.

P. Müller, A. Erkanli, and M. West. Bayesian Curve Fitting Using Multivariate Normal Mixtures. *Biometrika*, 83:67–79, 1996.

P. Müller, F. A. Quintana, and G. Rosner. Semiparametric Bayesian Inference for Multilevel Repeated Measurement Data. *Biometrics*, 63(1):280–289, 2007.

M. A. Newton, C. Czado, and R. Chappell. Bayesian inference for semiparametric binary regression. *Journal of the American Statistical Association* 91:142–153, 1996.

M. Plummer, N. Best, K. Cowles, and K. Vines. CODA: Output analysis and diagnostics for MCMC. *R package version 0.12-1*, 2007.

B. J. Smith. boa: Bayesian Output Analysis Program (BOA) for MCMC. *R package version 1.1.6-1*, 2007.

S. D. Spiegelhalter, N. G.Best, B. P. Carlin, and A. Van der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B* 64:583–639, 2002.

Alejandro Jara
Biostatistical Centre
Catholic University of Leuven
Leuven, Belgium
Alejandro.JaraVallejos@med.kuleuven.be

An Introduction to gWidgets

by John Verzani

Introduction

CRAN has several different packages that interface R with toolkits for making graphical user interfaces (GUIs). For example, among others, there are **RGtk2** (Lawrence and Temple Lang, 2006), **rJava**, and **tcltk** (Dalgaard, 2001). These primarily provide a mapping between library calls for the toolkits and similarly named R functions. To use them effectively to create a GUI, one needs to learn quite a bit about the underlying toolkit. Not only does this add complication for many R users, it can also be tedious, as there are often several steps required to set up a basic widget. The **gWidgets** package adds another layer between the R user and these packages providing an abstract, simplified interface that tries to be as familiar to the R user as possible. By abstracting the toolkit it is possible to use the **gWidgets** interface with many different toolkits. Although, open to the criticism that such an approach can only provide a least-common-denominator user experience, we'll see that **gWidgets**, despite not being as feature-rich as any underlying toolkit, can be used to produce fairly complicated GUIs without having as steep a learning curve as the toolkits themselves.

As of this writing there are implementations for three toolkits, **RGtk2**, **tcltk**, and **rJava** (with progress on a port to **RwxWidgets**). The **gWidgetsRGtk2** package was the first and is the most complete. Whereas **gWidgetstcltk** package is not as complete, due to limitations of the base libraries, but it has many useful widgets implemented. Installation of these packages requires the base toolkit libraries be installed. For **gWidgetstcltk** these are bundled with the windows distribution, for others they may require a separate download.

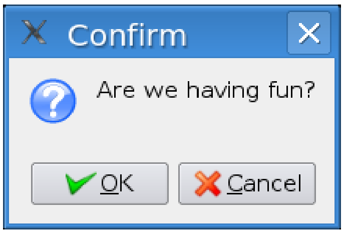
Dialogs

We begin by loading the package. Both the package and at least one toolkit implementation must be installed prior to this. If more than one toolkit implementation has been installed, you will be queried as to which one to use.

```
library("gWidgets")
```

The easiest GUI elements to create are the basic dialogs (Figure 1). These are useful for sending out quick messages, such as: [1]

```
gconfirm("Are we having fun?")
```



Are we having fun?

Figure 1: Simple dialog created by `gconfirm` using the **RGtk2** toolkit.

A basic dialog could be used to show error messages

```
options(error = function() {
  err = geterrmessage()
  gmessage(err, icon="error")
})
```

or, be an alternative to `file.choose`

```
source(gfile())
```

In **gWidgets**, these basic dialogs are modal, meaning the user must take some action before control of R is returned. The return value is a logical or string, as appropriate, and can be used as input to a further command. Modal dialogs can be confusing

[1] The code for these examples is available from http://www.math.csi.cuny.edu/pmg/gWidgets/rnews.R