2010

Non-standard semiparametric regression via BRugs

Jennifer Marley
Australian Bureau of Statistics

M. Wand
University of Wollongong, mwand@uow.edu.au

Publication Details
Non-standard semiparametric regression via BRugs

Abstract
We provide several illustrations of Bayesian semiparametric regression analyses in the BRugs package. BRugs facilitates use of the BUGS inference engine from the R computing environment and allows analyses to be managed using scripts. The examples are chosen to represent an array of non-standard situations, for which mixed model software is not viable. The situations include: the response variable being outside of the one-parameter exponential family, data subject to missingness, data subject to measurement error and parameters entering the model via an index.

Disciplines
Physical Sciences and Mathematics

Publication Details

This journal article is available at Research Online: http://ro.uow.edu.au/infopapers/3507
Non-Standard Semiparametric Regression via BRugs

Jennifer K. Marley
Australian Bureau of Statistics

Matt P. Wand
University of Wollongong

Abstract

We provide several illustrations of Bayesian semiparametric regression analyses in the BRugs package. BRugs facilitates use of the BUGS inference engine from the R computing environment and allows analyses to be managed using scripts. The examples are chosen to represent an array of non-standard situations, for which mixed model software is not viable. The situations include: the response variable being outside of the one-parameter exponential family, data subject to missingness, data subject to measurement error and parameters entering the model via an index.

Keywords: additive models, BUGS, hierarchical Bayesian models, Markov chain Monte Carlo, measurement error models, missing data, mixed models, penalized splines.

1. Introduction

Semiparametric regression is an enhancement of parametric regression that uses penalized spline basis functions to achieve greater flexibility. Many semiparametric regression models have useful formulations as hierarchical Bayesian models, with variance component parameters used to control the degrees of freedom of smooth functions; see, for example, Chapter 16 of Ruppert et al. (2003) and Brezger and Lang (2006). An immediate pay-off is that Markov chain Monte Carlo (MCMC) software for hierarchical Bayesian models can be used for fitting and inference. This fact has been exploited by, for example, Crainiceanu et al. (2005), Gurrin et al. (2005) and Zhao et al. (2006) in various semiparametric regression contexts via the WinBUGS Bayesian inference package (Lunn et al. 2000; Spiegelhalter et al. 2003b) – a Microsoft Windows interface to the BUGS inference engine (Spiegelhalter et al. 2003a)

In this article we focus on non-standard semiparametric regression situations. By ‘non-standard’ we mean variants of semiparametric regression that fall outside the conventional set-up in which the response distributions are in the one-parameter exponential family and all data are cleanly observed. Examples of non-standard situations include (1) overdispersed
count responses that warrant more flexible distributions such as those in the negative binomial family, (2) predictors being subject to measurement error, in which case a semiparametric regression model is augmented by a measurement error model, and (3) presence of an index inside a smooth function. Through a suite of examples we demonstrate how the BUGS inference engine is able to effectively deal with such non-standard situations. We access BUGS using the package BRugs (Ligges et al. 2009) in the R computing environment (R Development Core Team 2010). Employment of BRugs has the advantage that an entire analysis can be managed using a single R script. Because R is used at the front-end and back-end of the analysis, one can take advantage of R’s functionality for data input and pre-processing, as well as summary and graphical display. The package R2WinBUGS (Sturtz et al. 2005) has capabilities similar to those of BRugs.

Currently, BRugs only runs on the Windows operating system and communicates with the version of BUGS known as OpenBUGS (Thomas et al. 2006). The BRugs package is not available from the Comprehensive R Archive Network (CRAN), but located at the CRANextras repository: http://www.stats.ox.ac.uk/pub/RWin/src/contrib/.

The frequentist counterpart of this general approach to semiparametric regression involves mixed model representations of penalized splines. Ngo and Wand (2004) provided numerous examples of semiparametric analyses using the mixed model software modules PROC MIXED in SAS and lme() in S-PLUS. (The R version of lme() was not used in Ngo and Wand (2004), but the S-PLUS code given there requires only minor modification to run in R.) Unfortunately, conventional mixed model software does not cater for departures from standard situations. At the time of this writing, BUGS is essentially the only established software product that supports non-standard semiparametric regression analysis. Another possible contender is the alternative MCMC-based package, rjags (Plummer 2003, 2009), although we have not explored its use. The software package BayesX (Brezger et al. 2005) also supports Bayesian semiparametric regression, but it is not able to handle some of the non-standard situations treated here.

An undercurrent of the use of BUGS is the embedding of non-standard semiparametric regression within a graphical models framework. See Wand (2009) for more details on this viewpoint of semiparametric regression, in which graphical models, and associated inference engines, have great potential for streamlining complicated analyses.

Section 2 lays down infrastructure required for the rest of the paper. Each of the following sections deal with a specific non-standard semiparametric regression setting and illustrative example. In Section 9 we address the issue of formal chain diagnosis. Conclusions are summarized in Section 10

2. Preparatory infrastructure

Before embarking on the examples we lay out the mathematical infrastructure on which they are based.

2.1. Distributional notation

The density function of a random vector \( x \) is denoted by \( [x] \). The conditional density function of \( y \) given \( x \) is denoted by \( [y|x] \). If, for \( 1 \leq i \leq n \), \( y_i \) has distribution \( D_i \) and the \( y_i \) are
Table 1: Distributions used in the examples. The density function argument $x$ and parameters range over $\mathbb{R}$ unless otherwise specified.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Density function in $x$</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gamma</td>
<td>$x^{\alpha - 1}e^{-x/\beta}/\Gamma(\alpha)\beta^\alpha$, $x &gt; 0; \alpha, \beta &gt; 0$</td>
<td>Gamma($\alpha, \beta$)</td>
</tr>
<tr>
<td>$t$</td>
<td>$\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\nu}\sigma^2\Gamma(\nu/2)}\left[1 + \frac{(x-\mu)^2}{\nu\sigma^2}\right]^{-(\nu+1)/2}; \sigma, \nu &gt; 0$</td>
<td>$t(\mu, \sigma^2, \nu)$</td>
</tr>
<tr>
<td>Negative binomial</td>
<td>$\frac{\kappa^x\Gamma(x + \kappa)}{\Gamma(\kappa)(\kappa + \mu)\Gamma(x + 1)}; x = 0, 1 \ldots; \kappa, \mu &gt; 0$</td>
<td>Negative Binomial($\mu, \kappa$)</td>
</tr>
<tr>
<td>Half-Cauchy</td>
<td>$\frac{2A}{\pi(x^2 + A^2)}; x &gt; 0; A &gt; 0$</td>
<td>Half-Cauchy($A$)</td>
</tr>
</tbody>
</table>

2.2. Penalized splines

In many semiparametric regression contexts, nonparametric functional relationships are handled through the modelling mechanism:

$$f(x) = \beta_0 + \beta_1 x + \sum_{k=1}^{K} u_k z_k(x), \quad u_k \overset{\text{iid}}{\sim} N(0, \sigma_u^2).$$

Here $z_1, \ldots, z_K$ are a set of spline basis functions. Our default is the $z_k$ corresponding to O’Sullivan splines as described in Wand and Ormerod (2008), which provides a close approximation to smoothing splines. In some of our examples the $z_k$ need to be computed inside BUGS and computational procedures required for O’Sullivan splines (e.g. spectral decomposition) are not available. For this reason, we also use simple truncated line basis functions: $z_k(x) = (x - \kappa_k)_+$ for some knot sequence $\kappa_1, \ldots, \kappa_K$. The knots are usually taken to be equally spaced over the range of the predictor. For most function estimation situations, including all in the examples of this article, $K = 25$ is an adequate number of knots.

In penalized spline smoothing $\sigma_u$ plays the role of a smoothing parameter. However, a more meaningful measure of the amount of smoothing is effective degrees of freedom (Buja et al. 1989); sometimes shortened to degrees of freedom. For the Bayesian Gaussian response model

$$y_i \mid \beta_0, \beta_1, \sigma_u^2, \sigma_\varepsilon^2 \overset{\text{iid}}{\sim} N \left( \beta_0 + \beta_1 x + \sum_{k=1}^{K} u_k z_k(x), \sigma_\varepsilon^2 \right), \quad u_k \overset{\text{iid}}{\sim} N(0, \sigma_u^2), \quad \beta_0, \beta_1 \overset{\text{iid}}{\sim} N(0, \sigma_\beta^2),$$

independent then we write $y_i \overset{\text{iid}}{\sim} D_i$. 

Table 1 lists distributions used in the examples, and the parametrisation of their density functions.
the vector of fitted values,

\[
\begin{bmatrix}
\hat{f}(x_1) \\
\vdots \\
\hat{f}(x_n)
\end{bmatrix} = \begin{bmatrix}
\beta_0 + \beta_1 x_1 + \sum_{k=1}^{K} u_k z_k(x_1) \\
\vdots \\
\beta_0 + \beta_1 x_n + \sum_{k=1}^{K} u_k z_k(x_n)
\end{bmatrix},
\]

for given \( \sigma^2_u \) and \( \sigma^2_\varepsilon \), is

\[
S(\sigma^2_u, \sigma^2_\varepsilon) \equiv C[C^T C + \text{blockdiag}\{\sigma^{-2}_\beta I_2, (\sigma^2_\varepsilon/\sigma^2_u) I_K\}]^{-1} C^T,
\]

\( C \equiv [1 \ x_1 \ z_1(x_1) \ \cdots \ z_K(x_i)]_{1 \leq i \leq n} \) is the design matrix and \( I_d \) is the \( d \times d \) identity matrix. By analogy with effective degrees of freedom in frequentist contexts, an appropriate definition of degrees of freedom in this Bayesian context is

\[
df(\sigma^2_u, \sigma^2_\varepsilon) \equiv \text{tr}\{S(\sigma^2_u, \sigma^2_\varepsilon)\} = \text{tr}\left((C^T C + \text{blockdiag}\{\sigma^{-2}_\beta I_2, (\sigma^2_\varepsilon/\sigma^2_u) I_K\})^{-1} C^T C\right). \tag{1}
\]

For deviations from this simplest situation, such as those considered in Sections 4 and 6, the appropriate definition of effective degrees of freedom is somewhat thorny. However, we continue to use (1) as a reasonable measure of the amount of smoothing.

### 2.3. Standardization and default priors

As a general rule, we standardize all continuous variables of interest before commencing Bayesian analyses. This makes the priors scale invariant and can also lead to better behaviour of the MCMC. Standardization is especially important for the real data examples, where the measurements are recorded on several different scales. The examples involving simulated data are such that standardization makes little difference, so this step is omitted to keep the code simpler.

In each of the examples we do not have prior knowledge about the model parameters, so use non-informative priors. The default prior for a fixed effects parameter vector \( \beta \) is a diffuse Gaussian distribution:

\[
\beta \sim N(0, \sigma^2_\beta I).
\]

with \( \sigma^2_\beta = 10^8 \). The default prior for a standard deviation parameter \( \sigma \) is

\[
\sigma \sim \text{Half-Cauchy}(A).
\]

with \( A = 25 \). This is consistent with recommendations given in Gelman (2006) for achieving non-informativeness for variance parameters. **BUGS** does not offer direct specification of Half-Cauchy distributions. We get around this by using the result:

\[
X_1 \sim N(0, \sigma^2_1), \ X_2 \sim N(0, \sigma^2_2) \ \text{independently} \ \text{implies} \ |X_1/X_2| \sim \text{Half-Cauchy}(\sigma_2/\sigma_1).
\]

It follows that we can obtain a Half-Cauchy(25) variate by dividing a \( N(0, 625) \) random variate by a standard normal one, and taking absolute value. This fact is exploited in the **BUGS** code in each of the examples. However, since **BUGS** works with precision parameters \( \tau = 1/\sigma^2 \) the above translates to

\[
\tau = (N/D)^2, \ \text{with} \ N \sim N(0, 1), \ D \sim N(0, 625) \ \text{independently}.
\]
But because of the **BUGS** dispersion convention the \( N(0, 625) \) distribution is specified using \( \text{dnorm}(0, 0.0016) \) (since \( 0.0016 = 1/625 \)). Each of the **BUGS** scripts in the upcoming examples contain code for Half-Cauchy standard deviation prior specification. The examples to follow use only these single hyperparameter choices \( (\sigma_0^2 = 10^8 \text{ and } A = 25) \) for reasons of brevity. In practice, it is recommended that sensitivity checks be conducted on the hyperparameters.

### 2.4. Markov chain Monte Carlo defaults

All Bayesian models are fitted using standardized versions of continuous variables. Unless otherwise stated, Markov chain Monte Carlo examples use a burn-in period of 5000 iterations and then retain 5000 iterations. They are then thinned by a factor of 5, resulting in samples of size 1000 being retained for inference.

### 3. Negative binomial additive model

In this example we revisit the data set and model given in Thurston *et al.* (2000). The response variable data consists of adduct counts (\texttt{adductCount}) for 77 former smokers in a lung cancer study (source: Wiencke *et al.* 1999). Four predictors are available:

- \texttt{ageInit}: Age of smoking initiation.
- \texttt{yearsSmoking}: Number of years of smoking.
- \texttt{cigsPerDay}: Number of cigarettes smoked per day.
- \texttt{yearsSinceQuit}: Number of years since quitting.

As explained in Thurston *et al.* (2000), \texttt{adductCount} is over-dispersed and a Poisson additive model is not realistic. They make a case for a model of the form:

\[
\text{adductCount}_i \sim \text{Negative Binomial} \left\{ f_1(\text{ageInit}_i) + f_2(\text{yearsSmoking}_i) + f_3(\text{yearsSinceQuit}_i) + f_4(\text{cigsPerDay}_i), \kappa \right\}
\]  

(2)

where the \( f_j \)s are arbitrary smooth functions. Thurston *et al.* (2000) devised kernel methods to fit *negative binomial additive models* such as (2). Instead we take a hierarchical Bayesian approach with penalized spline modelling for the \( f_j \)s (Section 2.2).

Section 2.3 describes choice of priors for the fixed effects and standard deviation parameters. The prior for the shape parameter \( \kappa \) is motivated by the result:

\[
[y] \sim \text{Negative Binomial}(\mu, \kappa) \quad \text{if} \quad [y|g] \sim \text{Poisson}(g) \quad \text{and} \quad [g] \sim \text{Gamma}(\kappa, \mu/\kappa).
\]  

(3)

The coefficient of variation of the latent gamma variable \( g \) is

\[
\omega = \sqrt{\text{Var}(g)/E(g)} = \kappa^{-1/2}.
\]

This means that \( \kappa \) is the squared reciprocal of the standard deviation-like parameter \( \omega \). Based on the advice in Gelman (2006), this suggests use of

\[
[\omega] \sim \text{Half-Cauchy}(A) \quad \text{or} \quad [\omega] \sim \text{Uniform}(0, A).
\]
We settled on $\omega \sim \text{Half-Cauchy}(25)$; the same prior placed on other standard deviation parameters.

Even though BUGS has a command named dnegbin for specification of negative binomial distributions it has restrictions, such as the shape parameter being an integer. Therefore, we use the latent gamma random variable representation (3) for specification of a negative binomial node.

The resulting BUGS script is:

```r
for (i in 1:n)
{
  log(mu[i]) <- (beta0 + beta1*x1[i] + beta2*x2[i] + beta3*x3[i] + beta4*x4[i]
  + inprod(u1[],Z1[i,]) + inprod(u2[],Z2[i,])
  + inprod(u3[],Z3[i,]) + inprod(u4[],Z4[i,]))
  rateParm[i] <- kappa/mu[i] ; g[i] ~ dgamma(kappa,rateParm[i])
  y[i] <- dpois(g[i])
}
for (k in 1:numKnots1)
{
  u1[k] ~ dnorm(0,tauU1)
}
for (k in 1:numKnots2)
{
  u2[k] ~ dnorm(0,tauU2)
}
for (k in 1:numKnots3)
{
  u3[k] ~ dnorm(0,tauU3)
}
for (k in 1:numKnots4)
{
  u4[k] ~ dnorm(0,tauU4)
}

beta0 ~ dnorm(0,1.0E-8) ; beta1 ~ dnorm(0,1.0E-8)
beta2 ~ dnorm(0,1.0E-8) ; beta3 ~ dnorm(0,1.0E-8)
beta4 ~ dnorm(0,1.0E-8)
numerU1 ~ dnorm(0,1) ; denomU1 ~ dnorm(0,0.0016)
tauU1 <- pow(numerU1/denomU1,2)
numerU2 ~ dnorm(0,1) ; denomU2 ~ dnorm(0,0.0016)
tauU2 <- pow(numerU2/denomU2,2)
numerU3 ~ dnorm(0,1) ; denomU3 ~ dnorm(0,0.0016)
tauU3 <- pow(numerU3/denomU3,2)
numerU4 ~ dnorm(0,1) ; denomU4 ~ dnorm(0,0.0016)
tauU4 <- pow(numerU4/denomU4,2)
numerKappa ~ dnorm(0,1) ; denomKappa ~ dnorm(0,0.0016)
kappa <- pow(numerKappa/denomKappa,2)
```
Preliminary runs for this model applied to the adduct count data showed that chain convergence is quite slow. Section 9 contains some relevant chain diagnostics. This led us to use a burnin of size 100000, followed by 50000 retained iterations with a thinning factor of 50.

Figure 1 shows the fitted functions for model (2) and are consistent with those obtained in Thurston et al. (2000).

In Figure 2 we summarize the MCMC output, both graphically and numerically. The plots in the ‘lag 1’ column are the MCMC sample against its 1-lagged values and those in the ‘acf’ column are the sample autocorrelation function of the MCMC sample. These sets of plots allow for quick visual appreciation of the ‘stickiness’ of the chains. The 95% credible interval for $\kappa$ is (0.52, 0.97) which is consistent with over-dispersion (the Poisson distribution is a limiting case of the negative binomial as $\kappa \to \infty$). The credible intervals for the four $\sigma_u$...
parameters are away from zero, which is indicative of all effects being non-linear.

In Section 9 we carry out some formal convergence diagnostics for the third $\sigma_u$ parameter.

Figure 3 shows the behaviour of the MCMC for estimation of the $f_j$s at the quartiles of the corresponding predictor. All chains are seen to be well-behaved.

Recently, we became aware of the R package **gamlss** (Stasinopoulos and Rigby 2007). The **gamlss** documentation indicates support for negative binomial additive models, although we are yet to explore this in any depth.
Figure 3: Summary of MCMC-based inference for fitted functions in the negative binomial additive model. The columns are: missing predictor, trace plot of MCMC sample, plot of sample against 1-lagged sample, sample autocorrelation function, kernel estimates posterior density and basic numerical summaries.
4. Nonparametric regression with missingness

In this section we consider the simple nonparametric regression setting

\[ y_i = f(x_i) + \varepsilon_i, \quad \varepsilon_i \overset{\text{ind.}}{\sim} N(0, \sigma^2_\varepsilon), \quad 1 \leq i \leq n, \quad (4) \]

for a smooth function \( f \). Assume that the \( x_i \)s can be modelled as coming from a normal distribution with mean \( \mu_x \) and variance \( \sigma^2_x \), but are subject to missingness. An appropriate hierarchical Bayesian model for this situation is

\[
y_i | x_i, \beta, u, \sigma^2_\varepsilon \overset{\text{ind.}}{\sim} N \left( \beta_0 + \beta_1 x_i + \sum_{k=1}^{K} u_k z_k(x_i), \sigma^2_\varepsilon \right),
\]

\[
u | \sigma^2_u \sim N(0, \sigma^2_u I), \quad x_i | \mu_x, \sigma^2_x \sim N(\mu_x, \sigma^2_x),
\]

\[
\beta \sim N(0, 10^8 I), \quad \mu_x \sim N(0, 10^8),
\]

\[
\sigma_u \sim \text{Half-Cauchy}(25), \quad \sigma_\varepsilon \sim \text{Half-Cauchy}(25), \quad \sigma_x \sim \text{Half-Cauchy}(25).
\]

We illustrate \textsc{BRugs} fitting of (5) to simulated data with

\[ n = 300, \quad f(x) = \sin(4\pi x), \quad \mu_x = \frac{1}{2}, \quad \sigma^2_x = \frac{1}{36}, \quad \sigma^2_\varepsilon = 0.35 \quad (6) \]

and 20% of the \( x_i \)s missing completely at random.

Since the spline basis functions for the missing \( x_i \)s have to be computed inside \textsc{BUGS} we use simple truncated line spline basis functions:

\[ z_k(x) = (x - \kappa_k)^+ \quad \text{with} \quad \kappa_k = \{(K + 1 - k) \min(x_{i}^{\text{obs}}) + k \max(x_{i}^{\text{obs}})\} / (K + 1), \quad 1 \leq k \leq K \]

and \( K = 25 \). Here \( x_{i}^{\text{obs}} \) denotes the \( i \)-th observed \( x_i \) value. The relevant \textsc{BUGS} code is

```r
for (i in 1:nObs)
{
  muObs[i] <- beta0 + beta1*xObs[i] + inprod(u[],ZxObs[i,])
  yxObs[i] ~ dnorm(muObs[i],tauEps)
  xObs[i] ~ dnorm(muX,tauX)
}
for (i in 1:nMis)
{
  muMis[i] <- beta0 + beta1*xMis[i] + inprod(u[],ZxMis[i,])
  yxMis[i] ~ dnorm(muMis[i],tauEps)
  xMis[i] ~ dnorm(muX,tauX)
}
for (k in 1:numKnots)
{
  for (i in 1:nMis)
  {
    ZxMis[i,k] <- (xMis[i]-knots[k])*step(xMis[i]-knots[k])
  }
  u[k] ~ dnorm(0,tauU)
}
beta0 ~ dnorm(0,1.0E-8) ; beta1 ~ dnorm(0,1.0E-8)
```
Figure 4: MCMC-based estimate of $f$ in the missing predictor nonparametric regression model. The pale green shaded region corresponds to pointwise 95% credible sets. The orange points are those for which the $x$ values were missing and, while known from simulation, were hidden from the fitting procedure.

$$
\begin{align*}
\mu_X & \sim \text{dnorm}(0, 1.0E-8) \\
\text{numer}_X & \sim \text{dnorm}(0, 1) ; \ \text{denom}_X \sim \text{dnorm}(0, 0.0016) \\
\tau_X & \leftarrow \text{pow}(\text{numer}_X/\text{denom}_X, 2) \\
\text{numer}_U & \sim \text{dnorm}(0, 1) ; \ \text{denom}_U \sim \text{dnorm}(0, 0.0016) \\
\tau_U & \leftarrow \text{pow}(\text{numer}_U/\text{denom}_U, 2) \\
\text{numer}_\text{eps} & \sim \text{dnorm}(0, 1) ; \ \text{denom}_\text{eps} \sim \text{dnorm}(0, 0.0016) \\
\tau_\text{eps} & \leftarrow \text{pow}(\text{numer}_\text{eps}/\text{denom}_\text{eps}, 2)
\end{align*}
$$

where, for example, $y_{\text{obs}}[]$ is the vector of $y_i$ values that have an observed $x_i$ partner and $x_{\text{mis}}[]$ is the unobserved vector of missing $x$s.

Figure 4 shows the estimate of $f$ as well as pointwise 95% credible intervals. The missing data, known from simulation but hidden from the methodology, are shown as orange circles. The upper panels of Figure 5 summarizes the MCMC output produced by BUGS for the nodes $\mu_x$, $\sigma_x$ and $\sigma_e$, as well as $f$ evaluated at quartiles of $x$. The true values (6) from which the data were simulated are shown as vertical dashed lines in the posterior density plots. The middle panel monitors the effective degrees of freedom for estimation of $f$. The chains are seen to be reasonably well-behaved.
Figure 5: Summary of MCMC-based inference for parameters in the missing predictor non-parametric regression model. The columns are: parameter, trace plot of MCMC sample, plot of sample against 1-lagged sample, sample autocorrelation function, kernel estimates posterior density and basic numerical summaries. The vertical dashed lines in the density plots correspond to the true values of the parameters according to the simulation set-up.

Lastly, we study some of the output for the unobserved $x_i$s, which we denote here by $x_i^{\text{mis}}$. Five components were chosen at random and the MCMC summaries are shown in Figure 6. Interestingly, the posterior densities of some of the $x_i^{\text{mis}}$s are multimodal. This arises from the periodic nature of the underlying signal. Knowledge about the ordinate manifests in the posterior of the $x_i^{\text{mis}}$s as two or three clumps of probability mass corresponding, roughly, to horizontal slicing of $f$ at that ordinate.

In this example we have stayed with the simplest semiparametric regression setting to elucidate the missing data aspects. More elaborate semiparametric parametric models with missing data can be handled similarly.
5. Nonparametric regression with measurement error

In the previous section the $x_i$s in (5) were subject to missingness. Now suppose instead that they are subject to measurement error. Rather than observing $x_i$ we observe

$$w_i = x_i + z_i, \quad 1 \leq i \leq n,$$

(7)

where the $z_i \overset{\text{iid}}{\sim} N(0, \sigma_z^2)$ and independent of the $x_i$s. The contamination variance, $\sigma_z^2$, is assumed to be known.

This is an instance of nonparametric regression with measurement error. *Carroll et al. (2006)*
is a recent survey of this and related topics. A hierarchical Bayesian model for (4) and (7) is

\[ y_i | x_i, \beta, u, \sigma_e^2 \sim N \left( \beta_0 + \beta_1 x_i + \sum_{k=1}^{K} u_k z_k(x_i), \sigma_e^2 \right), \]

\[ u | \sigma_u^2 \sim N(0, \sigma_u^2 I), \quad x_i | \mu_x, \sigma_x^2 \sim N(\mu_x, \sigma_x^2), \quad w_i | x_i \sim N(x_i, \sigma_x^2), \]

\[ \beta \sim N(0, 10^8 I), \quad \mu_x \sim N(0, 10^8), \]

\[ \sigma_u \sim \text{Half-Cauchy}(25), \quad \sigma_e \sim \text{Half-Cauchy}(25), \quad \sigma_x \sim \text{Half-Cauchy}(25). \]

Models of type (8) were first formulated by Berry et al. (2002).

We illustrate BRugs fitting of (8) using the parameter settings given by (6) and with \( \sigma_z \) set to be 0.1. As for the missing data example spline basis functions have to be computed inside BUGS, so we used truncated line basis functions with knots

\[ \kappa_k = \{(K + 1 - k) \min(x_i) + k \max(x_i)\}/(K + 1), \quad 1 \leq k \leq K, \]

(which depend on the unobserved \( x_i \)'s) and \( K = 20 \). The BUGS code is:
Figure 8: Summary of MCMC-based inference for parameters in the nonparametric regression measurement error model. The columns are: parameter, trace plot of MCMC sample, plot of sample against 1-lagged sample, sample autocorrelation function, kernel estimates posterior density and basic numerical summaries. The vertical dashed lines in the density plots correspond to the true values of the parameters according to the simulation set-up. (These is no easily defined “true value” for the degrees of freedom for $f$, so the vertical line is omitted in this case.)

```r
for (i in 1:n)
{
  x[i] ~ dnorm(muX,tauX)
  w[i] ~ dnorm(x[i],tauZ)
  mu[i] <- beta0 + beta1*x[i] + inprod(u[],Z[i,])
  y[i] ~ dnorm(mu[i],tauEps)
}

for (k in 1:numKnots)
{
  knots[k] <- ((numKnots+1-k)*ranked(x[],1)+k*ranked(x[],n))/(numKnots+1)
  for (i in 1:n)
  {
    Z[i,k] <- (x[i]-knots[k])*step(x[i]-knots[k])
  }
  u[k] ~ dnorm(0,tauU)
}

beta0 ~ dnorm(0,1.0E-8) ; beta1 ~ dnorm(0,1.0E-8)
muX ~ dnorm(0,1.0E-8)
umerX ~ dnorm(0,1) ; denomX ~ dnorm(0,0.0016)
```
Non-Standard Semiparametric Regression via BRugs

tauX <- pow(numerX/denomX,2)
numerU ~ dnorm(0,1) ; denomU ~ dnorm(0,0.0016)
tauU <- pow(numerU/denomU,2)
numerEps ~ dnorm(0,1) ; denomEps ~ dnorm(0,0.0016)
tauEps <- pow(numerEps/denomEps,2)

Figure 7 shows the estimate of $f$ as well as pointwise 95% credible intervals. The orange circles are the unobserved $(x_i, y_i)$ pairs which, because this is a simulation study, are known. The curve estimate is seen to be quite reasonable despite having to adjust for contamination of the $x_i$s.

The upper panels of Figure 8 are the analogue of Figure 5 for the current measurement error example. Once again, the chains are seen to be reasonably well-behaved and true parameters are inside the 95% credible sets.

Even though BRugs and BUGS provide quite pleasing results for this example, it comes at a price in terms of computing time. While most of the other examples in this article take minutes to run on our computers, this example takes about one day.

6. Robust nonparametric regression via the $t$ distribution

Robustification of regression methodology has a large literature, some of which is surveyed in Rousseeuw and Leroy (1987). An attractive model-based approach is to use the $t$ distribution for the responses since, for low values of the degrees of freedom parameter, gross outliers occur with moderate probability. An early reference of $t$ distribution-based robust regression is Lange et al. (1989). Recently, Staudenmayer et al. (2009) described a penalized spline mixed model approach to nonparametric regression using the $t$ distribution. They took a maximum likelihood approach, with EM algorithm used for fitting. In this section we instead take a hierarchical Bayesian approach and show how BRugs can be used for effective fitting and inference.

Suppose that we observe regression data $(x_i, y_i), 1 \leq i \leq n$, but the $y_i$s are subject to occasional outlying values. Then an appropriate model is

$$y_i = f(x_i) + \varepsilon_i, \quad \varepsilon_i \overset{ind}{\sim} t(0, \sigma^2, \nu), \quad 1 \leq i \leq n,$$

for a smooth function $f$. The BUGS command dt supports specification of $t$ distributed nodes but not all three parameters can be treated stochastically. Instead we use the result

$$[y] \sim t(\mu, \sigma^2, \nu) \quad \text{if} \quad [y|g] \sim N(\mu, \sigma^2/g) \quad \text{and} \quad [g] \sim \Gamma(\nu^2/2, \nu^2/2).$$

Following Verdinelli and Wasserman (1991) we put a Beta prior on the reciprocal of $\nu$. Specifically

$$\nu \sim \text{Beta}(1.75, 2.5) \quad \text{where} \quad \nu = 1/\nu.$$ 

Their justification for this prior is that it corresponds to a fair amount of uncertainty about $\nu$ and there is not too much probability near normality. The full Bayesian penalized spline
model is

\[ y_i | \beta, \mathbf{u}, \sigma_\varepsilon^2, \nu \sim t \left( \beta_0 + \beta_1 x_i + \sum_{k=1}^{K} u_k z_k(x_i), \sigma_\varepsilon^2, \nu \right), \]

\[ \mathbf{u} | \sigma_u^2 \sim N(0, \sigma_u^2 \mathbf{I}), \quad \beta \sim N(0, 10^8 \mathbf{I}), \]

\[ \sigma_u \sim \text{Half-Cauchy}(25), \quad \sigma_\varepsilon \sim \text{Half-Cauchy}(25) \] and \( \nu \sim \text{Beta}(1.75, 2.5), \nu = 1/\nu. \)

The corresponding **BUGS** script is:

```R
for (i in 1:n)
{
    mu[i] <- beta0 + beta1*x[i] + inprod(u[],Z[i,])
    precis[i] <- g[i]*tauEps
    y[i] ~ dnorm(mu[i],precis[i])
    g[i] ~ dgamma(halfNu,halfNu)
}
for (k in 1:numKnots)
{
    u[k] ~ dnorm(0,tauU)
}
```

---

Figure 9: MCMC-based estimate of \( f \) in the \( t \) distribution nonparametric regression example. The pale green shaded region corresponds to pointwise 95% credible sets. Left panel: data, estimated and true curves. Right panel: Estimated and true curves only.
Figure 10: Summary of MCMC-based inference for parameters in the $t$ distribution nonparametric regression model. The columns are: parameter, trace plot of MCMC sample, plot of sample against 1-lagged sample, sample autocorrelation function, kernel estimates posterior density and basic numerical summaries. The vertical dashed lines in the density plots correspond to the true values of the parameters according to the simulation set-up. (These is no easily defined “true value” for the degrees of freedom for $f$, so the vertical line is omitted here.)

\[
\begin{align*}
\text{beta0} & \sim \text{dnorm}(0,1.0\times10^{-8}) ; \\
\text{beta1} & \sim \text{dnorm}(0,1.0\times10^{-8}) \\
\text{numerU} & \sim \text{dnorm}(0,1) ; \text{denomU} \sim \text{dnorm}(0,0.0016) \\
\tauU & \leftarrow \text{pow}(\text{numerU}/\text{denomU},2) \\
\text{numerEps} & \sim \text{dnorm}(0,1) ; \text{denomEps} \sim \text{dnorm}(0,0.0016) \\
\tauEps & \leftarrow \text{pow}(\text{numerEps}/\text{denomEps},2) \\
\text{recipNu} & \sim \text{dbeta}(1.75,2.5) \\
\text{halfNu} & \leftarrow 0.5/\text{recipNu} ; \nu \leftarrow 2*\text{halfNu}
\end{align*}
\]

Figure 9 provides an illustration of BRugs fitting via this script for simulated data with $n = 300$, $\sigma_\varepsilon = 0.35$, $\nu = 1.5$ and $f(x) = \sin(4\pi x)$ on the unit interval. The fitted curve is close to the true function and exhibits good resistance to the outliers.

Graphical assessment of the MCMC and inferential summaries are provided by Figure 10. Mixing is seen to be excellent for this example. In addition, the posterior density functions conform quite well with the true parameter values.
7. Generalized partially linear single index model

The generalized partially linear single index model was devised by Carroll et al. (1997) for flexible dependence of a categorical response variable on a single index. The binary response version of the model takes the general form

\[ \logit\{P(y_i = 1)\} = f(\alpha^\top x_i) + \beta^\top z_i \]

(12)

where the \(y_i\) are the responses, the \(x_i\) are vectors containing measurements on a set of continuous predictors and the \(z_i\) contain other covariates. The function \(f\) is a smooth, but otherwise arbitrary, function of the single index \(\alpha^\top x\). For identifiability we impose the constraint \(\|\alpha\| = 1\).

Carroll et al. (1997) applied a version of (12) to data on coronary heart disease (CHD) from the Framingham Heart Study (Kannel et al. 1986) with local polynomial smoothing used for estimation of \(f\). Here we take a hierarchical Bayesian approach with \(f\) estimated via penalized splines. The model is

\[ \logit\{P(CHD_i = 1)\} = f(\alpha_1 \text{age}_i + \alpha_2 \text{trBlood}_i + \alpha_3 \text{logChol}_i) + \beta \text{smoker}_i \]

where \(\text{age}\) denotes the patient’s age, \(\text{trBlood} = \log(\text{systolic blood pressure} - 25)\), \(\text{logChol}\) is the logarithm of cholesterol level and \(\text{smoker}\) is an indicator for the patient being a smoker. As done by Carroll et al. (1997) we worked with versions of \(\text{age}, \text{trBlood}\) and \(\text{logChol}\) that were first linearly transformed to the unit interval.

We impose the restriction \(\|\alpha\| = 1\) by working with spherical coordinates as follows:

\[ \alpha_1 = -\sin(\phi) \cos(\theta), \quad \alpha_2 = -\sin(\phi) \sin(\theta), \quad \alpha_3 = -\cos(\phi). \]

The prior distributions for \(\phi\) and \(\theta\) are each taken to be independent uniforms on \((0, \pi)\) and \((0, 2\pi)\) respectively. Since the single index depends on parameter values the basis functions for estimation of \(f\) have to be computed inside \textsc{Bugs}. For this reason we worked with the truncated line model

\[ f(s) = \beta_0 + \beta_1 s + \sum_{k=1}^{K} u_k (s - \kappa_k)_+, \quad u_k \text{ i.i.d. } N(0, \sigma_u^2) \]

with priors \(\beta_0, \beta_1 \sim N(0, 10^8)\) and \(\sigma_u \sim \text{Half-Cauchy}(25)\). The \textsc{Bugs} code is:

```r
for (i in 1:n) {
  s[i] <- alpha1*x1[i] + alpha2*x2[i] + alpha3*x3[i]
  f[i] <- beta0 + betas*s[i] + inprod(u[],Z[i,])
  logit(mu[i]) <- f[i] + betaz*z[i]
  y[i] ~ dbern(mu[i])
}
alpha1 <- (-sin(phi)*cos(theta))
alpha2 <- (-sin(phi)*sin(theta))
alpha3 <- (-cos(phi))
for (k in 1:numKnots)
  {  knot[k] <- ((numKnots+1-k)*ranked(s[],1)+k*ranked(s[],n))/(numKnots+1)
  ```
Figure 11: Summary of MCMC-based inference for parameters in the generalized partially linear single index model. The columns are: parameter, trace plot of MCMC sample, plot of sample against 1-lagged sample, sample autocorrelation function, kernel estimates posterior density and basic numerical summaries.

\[
u[k] \sim \text{dnorm}(0, \tau_U)
\]
for (i in 1:n)
{
    \[
    Z[i,k] <- (s[i]-knot[k])*\text{step}(s[i]-knot[k])
    \]
}

\[
\phi \sim \text{dunif}(\phi_{\text{Min}}, \phi_{\text{Max}}) \quad \theta \sim \text{dunif}(\theta_{\text{Min}}, \theta_{\text{Max}})
\]
\[
\beta_0 \sim \text{dnorm}(0, 1.0\times 10^{-8}) \quad \beta_i \sim \text{dnorm}(0, 1.0\times 10^{-8})
\]
\[
\beta_z \sim \text{dnorm}(0, 1.0\times 10^{-8})
\]
\[
\text{numerU} \sim \text{dnorm}(0, 1) \quad \text{denomU} \sim \text{dnorm}(0, 0.0016)
\]
\[
\tau_U \sim \text{pow}(\text{numerU}/\text{denomU}, 2)
\]
Figure 12: Fitted function for the single index $\alpha_1 \text{age} + \alpha_2 \text{trBlood} + \alpha_3 \log\text{Chol}$. The pale green band corresponds to pointwise 95% credible sets.

Figure 11 summarizes the MCMC output for each of the model parameters. The first four rows correspond to the regression coefficients $\alpha_1, \alpha_2, \alpha_3$ and $\beta$. Each are seen to be significant. In particular, smoking is seen to have an adverse effect.

The estimated function $f$ of the single index is plotted in Figure 12. It is monotonic, but has a pronounced non-linear shape. Carroll et al. (1997) obtained a similar estimate of $f$ using a local polynomial approach.

8. Generalized additive model with measurement error

As was done Section 5, we will augment a semiparametric model with a measurement error model. In this case there are several covariates and the response variable is a count, so a generalized additive model is appropriate.

The data consists of daily measurements on weather, air pollution and mortalities in Milan, Italy, over the period 1980–1989. The response variable is the total number of deaths ($\text{mort}$). There are four predictor variables available, the last of which may contain measurement error:

- **day**: The sequential day number.
- **temp**: The average temperature.
- **humid**: The average relative humidity.
- **TSP**: The natural logarithm of the measured total suspended particles.

As in the simulated example in Section 5, the $\text{TSP}_i$, for $1 \leq i \leq 3652$, are taken to be unbiased but more variable versions of the true measures of TSP ($\text{trueTSP}_i$) that we would have liked to have observed. That is:

$$\text{TSP}_i = \text{trueTSP}_i + z_i, \quad z_i \sim N(0, \sigma_z^2).$$
The following Poisson additive model, an appropriate choice due to the response being a count variable, is fit to the Milan mortality data via a hierarchical Bayesian approach:

\[ \text{mort}_i \sim \text{Poisson}\left[\exp\{f_1(\text{day}_i) + f_2(\text{temp}_i) + f_3(\text{humid}_i) + \beta_4 \text{trueTSP}_i\}\right]. \] (13)

The \(f_j\)s are arbitrary smooth functions modelled through the use of O’Sullivan splines as the basis functions. In addition:

\[ \text{TSP}_i \sim N(\text{trueTSP}_i, \sigma^2_z), \quad \text{trueTSP}_i \sim N(\mu_x, \sigma^2_x), \]
\[ \mu_x \sim N(0, 10^8), \quad \text{and} \quad \sigma^2_x \sim \text{Half-Cauchy}(25). \]

Note that \(\text{trueTSP}\) only enters model (13) parametrically. This is due to previous investigations showing that the observed \(\text{TSP}\) has a linear, if not a very close to linear, relationship with the response variable \(\text{mort}\). Hence, unlike the simulated example in the Section 5, spline basis functions calculated from the estimated \(\text{trueTSP}\) do not need to be computed inside \textsc{BUGS}. \textsc{Ganguli et al.} (2005) considered a model similar to (13).

The \textsc{BUGS} script for the current example is:

```plaintext
for (i in 1:n) {
    x4[i] ~ dnorm(muX4,tauX4)
    w4[i] ~ dnorm(x4[i],tauZ)
    log(mu[i]) <- (beta0 + beta1*x1[i] + beta2*x2[i] + beta3*x3[i]
                   + beta4*x4[i] + inprod(u1[1],Z1[i,1])
                   + inprod(u2[1],Z2[i,1]) + inprod(u3[1],Z3[i,1]))
    y[i] ~ dpois(mu[i])
}
for (k in 1:numKnots1) {
    u1[k] ~ dnorm(0,tauU1)
}
for (k in 1:numKnots2) {
    u2[k] ~ dnorm(0,tauU2)
}
for (k in 1:numKnots3) {
    u3[k] ~ dnorm(0,tauU3)
}
beta0 ~ dnorm(0,1.0E-8); beta1 ~ dnorm(0,1.0E-8)
beta2 ~ dnorm(0,1.0E-8); beta3 ~ dnorm(0,1.0E-8)
beta4 ~ dnorm(0,1.0E-8)
muX4 ~ dnorm(0,1.0E-8)
umerX4 ~ dnorm(0,1); denomX4 ~ dnorm(0,0.0016)
tauX4 <- pow(numerX4/denomX4,2)
numerU1 ~ dnorm(0,1); denomU1 ~ dnorm(0,0.0016)
tauU1 <- pow(numerU1/denomU1,2)
numerU2 ~ dnorm(0,1); denomU2 ~ dnorm(0,0.0016)
tauU2 <- pow(numerU2/denomU2,2)
numerU3 ~ dnorm(0,1); denomU3 ~ dnorm(0,0.0016)
tauU3 <- pow(numerU3/denomU3,2)
```
Figure 13: Fitted functions for the Poisson additive model with augmented measurement error model applied to the Milan mortality data. The shaded regions correspond to pointwise 95% credible intervals. In each panel the vertical axis corresponds to mortality.

With a choice of $\sigma^2_z = 0.1$, Figure 13 shows the resulting fitted functions for model (13) applied to the Milan mortality data. As $\sigma^2_z$ increases, the slope of the fitted function depicting the relationship between the response $\text{mort}$ and the estimated true measures of total suspended particles $\text{trueTSP}$ in the lower right panel in Figure 13 becomes steeper. The other fitted functions change very little with different values of $\sigma^2_z$.

A graphical and numerical summary of the MCMC output is provided in Figure 14. The credible intervals for the three $\sigma^2_u$ parameters do not include zero, meaning the three corresponding effects are nonlinear. This is in agreement with the plots in Figure 13.

Figure 15 shows the behaviour of the MCMC for the estimation of the $f_j$s at the three quartiles of the corresponding predictor and for the estimation of $\beta$ for the estimated $\text{trueTSP}$ at the three quartiles of the estimated $\text{trueTSP}$. All the chains can be seen to be well-behaved, indicating satisfactory convergence.

As was the case with the nonparametric regression with measurement error example in the previous section, these results take an incredible amount of computing time to produce. The analyst must be prepared to wait when the dataset is large, as for the Milan mortality dataset which has 3652 observations. On our computers, this example took about 6 days to run.
<table>
<thead>
<tr>
<th>parameter</th>
<th>trace</th>
<th>lag 1</th>
<th>acf</th>
<th>density</th>
<th>summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_u$ (consec.</td>
<td>![trace plot]</td>
<td>![scatter plot]</td>
<td>![acf plot]</td>
<td>![density plot]</td>
<td>posterior mean: 9.33, 95% credible interval: (6.1, 14)</td>
</tr>
<tr>
<td>day number)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_u$ (mean daily</td>
<td>![trace plot]</td>
<td>![scatter plot]</td>
<td>![acf plot]</td>
<td>![density plot]</td>
<td>posterior mean: 0.664, 95% credible interval: (0.387, 1.07)</td>
</tr>
<tr>
<td>temp.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_u$ (mean rel.</td>
<td>![trace plot]</td>
<td>![scatter plot]</td>
<td>![acf plot]</td>
<td>![density plot]</td>
<td>posterior mean: 0.0279, 95% credible interval: (0.005, 0.086)</td>
</tr>
<tr>
<td>humidity)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mu$ for true TSP</td>
<td>![trace plot]</td>
<td>![scatter plot]</td>
<td>![acf plot]</td>
<td>![density plot]</td>
<td>posterior mean: -6.71e-05, 95% credible interval: (-0.038, 0.034)</td>
</tr>
<tr>
<td>$\sigma$ for true TSP</td>
<td>![trace plot]</td>
<td>![scatter plot]</td>
<td>![acf plot]</td>
<td>![density plot]</td>
<td>posterior mean: 0.949, 95% credible interval: (0.924, 0.974)</td>
</tr>
</tbody>
</table>

Figure 14: Summary of the MCMC output for parameters in the Poisson additive model with augmented measurement error model. The columns are: missing predictor, trace plot of MCMC sample, plot of sample against 1-lagged sample, sample autocorrelation function, kernel estimates posterior density and basic numerical summaries.

9. Formal convergence diagnostics

An issue that we have left aside until now is formal diagnosis of convergence. Consider, for example, the standard deviation parameter corresponding to $f_3(\text{yearsSinceQuit})$ in the negative binomial additive model example of Section 3. We denote this parameter by $\sigma_{u3}$. The trace plot for this parameter in Figure 2 shows some upwards movement in the chain towards the end. Does this mean that the chain has not yet converged? Or is this movement in keeping with the posterior distribution of $\sigma_{u3}$? We now address this question using some established convergence diagnostics.

Several methods have been proposed for diagnosing chain convergence. Perhaps the most popular are those developed in Gelman and Rubin (1992) and Brooks and Gelman (1998) and have been become known as Brooks-Gelman-Rubin statistics. The essence of their approach is...
Figure 15: Summary of the MCMC output for parameters in the Poisson additive model with augmented measurement error model. The columns are: missing predictor, trace plot of MCMC sample, plot of sample against 1-lagged sample, sample autocorrelation function, kernel estimates posterior density and basic numerical summaries.

to run several chains and compare the combined chain behaviour with within-chain behaviour. Figure 16 illustrates use of the Brooks-Gelman-Rubin statistic $\hat{R}_{\text{interval}}$ (Brooks and Gelman 1998) based on 3 chains for $\sigma_{u3}$ with different starting values.

Convergence, according to this diagnostic, is seen to occur after about 30000 iterations. Since we used a burnin of 100000 iterations for all examples, Figure 16 provides some reassurance for the validity of the inference for $\sigma_{u3}$ given in Figure 2.

A complete Bayesian/MCMC analysis requires multiple chain runs and graphical checks of the type presented in Figure 16 for all key parameters in the model.
26

Non-Standard Semiparametric Regression via BRugs

Figure 16: Brooks-Gelman-Rubin convergence diagnosis for $\sigma_{u3}$ (standard deviation parameter controlling the amount of smoothing for $f_3(\text{yearsSinceQuit})$) in the negative binomial additive model example. Upper panel: trace plots of each chain. Middle panel: trace plots of the numerator and denominator of the $\hat{R}_{\text{interval}}$ statistic. Lower panel: trace plot of the $\hat{R}_{\text{interval}}$ statistic. To aid visualization, each trace plot shows the ordinate for every 40th iteration.

10. Conclusions

We have demonstrated that complex semiparametric regression scenarios can be handled using the BUGS inference engine. When combined with BRugs, analysis and inference for a particular problem can be managed using a single computer script. Aside from the time factor, there is no firm barrier on the complications that can be handled via this approach. We envisage this time factor become less of an issue as Bayesian inference technology and computing power continue to improve.
Disclaimer

Views expressed in this paper are those of the authors, and do not necessarily represent those of the Australian Bureau of Statistics. Where quoted, they should be attributed clearly to the authors.

Acknowledgments

We are grateful to Andrew Gelman for advice on some aspects of the Bayesian modelling. The work of Jennifer K. Marley was supported by a cadetship from the Australian Bureau of Statistics.

References


A. R and BRugs scripts

Scripts, functions and data sets for running each of the examples accompany this paper, and are available along with this paper. Table 2 provides a summary of relevant files. Each of the R script and function files have detailed comments in their preamble that explains their purpose.

<table>
<thead>
<tr>
<th>Example</th>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section 3</td>
<td>negBinAdd.R</td>
<td>R script</td>
</tr>
<tr>
<td></td>
<td>adduct.txt</td>
<td>data set</td>
</tr>
<tr>
<td>Section 4</td>
<td>npRegMis.R</td>
<td>R script</td>
</tr>
<tr>
<td>Section 5</td>
<td>npRegMeaErr.R</td>
<td>R script</td>
</tr>
<tr>
<td>Section 6</td>
<td>tNpReg.R</td>
<td>R script</td>
</tr>
<tr>
<td>Section 7</td>
<td>gplsim.R</td>
<td>R script</td>
</tr>
<tr>
<td></td>
<td>framingham.txt</td>
<td>data set</td>
</tr>
<tr>
<td>Section 8</td>
<td>meaErrGAM.R</td>
<td>R script</td>
</tr>
<tr>
<td></td>
<td>milanMort.txt</td>
<td>data set</td>
</tr>
<tr>
<td>General</td>
<td>BRugsMCMC.R</td>
<td>R function for obtaining MCMC samples</td>
</tr>
<tr>
<td></td>
<td>BGRinterval.R</td>
<td>R function for computing BGR diagnostics</td>
</tr>
<tr>
<td></td>
<td>ZOSull.R</td>
<td>R function for obtaining $Z$ matrix of O’Sullivan spline basis functions</td>
</tr>
<tr>
<td></td>
<td>summMCMC.R</td>
<td>R function for summarising MCMC output</td>
</tr>
<tr>
<td></td>
<td>wait.R</td>
<td>R function for allowing pausing in R scripts</td>
</tr>
</tbody>
</table>

Table 2: Files corresponding to the examples in this paper.

Affiliation:
Jennifer K. Marley
Australian Bureau of Statistics
45 Benjamin Way
Belconnen 2617, Australia
E-mail: jennifer.marley@abs.gov.au

Matt P. Wand
School of Mathematics and Applied Statistics
University of Wollongong
Wollongong 2522, Australia
E-mail: mwand@uow.edu.au
URL: http://www.uow.edu.au/~mwand/