

UNIVERSITY OF TECHNOLOGY SYDNEY  
School of Mathematical and Physical Sciences  
**37457 Advanced Bayesian Methods**

## LABORATORY 3

**Goals:** The goals of Laboratory 3 are:

- Reinforce the ideas of penalized splines for flexible non-linear regression analysis.
- Explain how to carry out generalized additive model analyses in R. Both non-Bayesian and Bayesian approaches are treated. Even the non-Bayesian approach is much faster, at the end of the laboratory we describe advantages of the Bayesian approach.

### 1. Downloading Required Files

- (a) Decide on a folder in which you will store the files for this laboratory. It is best if you put them all in the same folder. A simple option is to put them on the Desktop folder, but please take note of where you store them for later.
- (b) Open a web-browser and go to the web-page:  
`matt-wand.utsacademics.info/37457.html`
- (c) Scroll down to the heading **Files for Laboratory 3** and download each of the files underneath this heading into the same folder. There are 6 files in total with names:
  - `penSplinesDetails.R`
  - `ozoneCalif.txt`
  - `ozoneCalifViaStan.R`
  - `BostonMortgages.txt`
  - `BostonMortgagesViaStan.R`
  - `penSplGEV.R`
- (d) The remainder of this laboratory assumes that these the filenames are **exactly** as listed here. If the downloading process corrupts the filenames in any way then you need to change the filenames to be these.
- (e) Check to see if `rstan` is installed on the computer that you are using. If not then you will need to install `rstan` for this laboratory. The instructions are on the subject web-site under the heading **Advice About Laptop Preparation For Computer Laboratories**.
- (f) Make sure that the package `HRW` is installed on the computer that you are using. A quick check is to issue the command `library(HRW)` and see whether or not an error is produced. If the package is not present then the command:  
`install.packages("HRW")`  
is required for this laboratory.

### 2. Some Penalized Spline Underpinnings

- (a) This task involves running the R script `penSplinesDetails.R`.

- (b) Start an R session and make sure that the current working directory corresponds to the directory where `penSplinesDetails.R` is stored on your computer.
- (c) `source("penSplinesDetails.R")`  
This runs a self-paced tutorial on some details on penalized splines, which are a building block of generalized additive models and other advanced non-linear regression models.

### 3. Generalized Additive Model Analysis for Data From a Californian Ozone Study

- (a) Start an R session and make sure that the file `ozoneCalif.txt` is in the current working directory.
- (b) `ozoneCalif <- read.table("ozoneCalif.txt", header=TRUE)`
- (c) `pairs(ozoneCalif)`

The last command produces all pairwise scatterplot between the 4 variables in the `ozoneCalif` data frame. The first row is particularly relevant since it shows the (possibly non-linear) relationship between the main variable on interest, ozone level, and three potential meteorological predictor variables.

```
library(mgcv)
fitGAM <- gam(ozone ~ s(inversion.base.height)
               + s(daggett.pressure.gradient)
               + s(inversion.base.temp),
               data = ozoneCalif)
```

This loads the R package `mgcv` and fits a non-Bayesian generalized additive model to the data of the form:

$$y_i \stackrel{\text{ind.}}{\sim} N\left(f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}), \sigma_\varepsilon^2\right), \quad 1 \leq i \leq n.$$

where  $\stackrel{\text{ind.}}{\sim}$  stands for “distributed independently as”.

- (d) `gam.check(fitGAM)`

The output from this last command shows that the residuals are in keeping with the normality assumption and the number of spline basis is reasonable. If there were concerns about the latter than a re-issuing of the above fit command with e.g. `s(inversion.base.height, k=25)` will result in a larger spline basis function being used.

- (e) `plot(fitGAM, pages=1, shade=TRUE, shade.col="palegreen")`  
The resulting plots indicate pronounced non-linear forms of the  $f_1$ ,  $f_2$  and  $f_3$  functions.
- (f) So far in this part of Laboratory 3 we have done a non-Bayesian generalized additive model analysis. We will finish with a Bayesian version using the Stan Bayesian inference engine. This connects generalized additive models with other parts of the subject. However, there are also advantages of the Bayesian inference engine approach that are listed at the end of this laboratory. A *disadvantage* that we will see is the slowness. Depending on the application area this may or not be acceptable. In this application the data collection took a huge amount of time compared to Markov chain Monte Carlo-based analysis.

Make sure that the file `ozoneCalifViaStan.R` is in the current working directory of your R session.

Issue the command `source("ozoneCalifViaStan.R")`

Depending on the capacity of your computer, this will take a few minutes to run. The script eventually produces two plots:

- Markov chain Monte Carlo summary graphics for the effective degrees of freedom of each of  $f_1$ ,  $f_2$  and  $f_3$ . This concept is explained in Section 2.6 of the Penalized Spline notes.
  - Bayesian estimates of  $f_1$ ,  $f_2$  and  $f_3$  with pointwise 95% credible intervals. The non-linear effects are similar to those obtained using the `mgcv` package.
- (g) Perhaps later look through the code in `ozoneCalifViaStan.R`, especially that involving Stan model specification, to see how a Bayesian generalized additive model is programmed. Note that the function `ZOSull()` within the `HRW` package is used to set up spline basis function design matrices for each predictor variable.

#### 4. Generalized Additive Model Analysis for Mortgage Data from Boston, U.S.A.

- (a) Start an R session and make sure that the file `BostonMortgages.txt` is in the current working directory.
- (b) 

```
BostonMortgages <- read.table("BostonMortgages.txt", header=TRUE)
library(mgcv)
BostonMortgages$denyBinary <- as.numeric(BostonMortgages$deny == "yes")
fitGAM <- gam(denyBinary ~ s(dir) + s(lvr) + black + pbcr + self
              + single + factor(ccs),
              data = BostonMortgages, family="binomial")
```

In this case the response is binary (rather than continuous in the previous example). The previous command fits a *logistic* generalized additive model of the form:

$$y_i \stackrel{\text{ind.}}{\sim} \text{Bernoulli}\left(1/[1+\exp\{-(f_1(x_{1i})+f_2(x_{2i})+\beta_3 x_{3i}+\dots+\beta_{11}x_{11i})\}]\right), \quad 1 \leq i \leq n.$$

where

$$y_i = \begin{cases} 1 & \text{if the } i\text{th mortgage application is denied,} \\ 0 & \text{otherwise.} \end{cases}$$

Also,  $x_{1i}$  and  $x_{2i}$  are data on continuous financial variables named:

“debt payments to income ratio” and “loan size to property value ratio”

and  $x_{3i}, \dots, x_{11i}$  are binary indicator variables such as

$$x_{3i} = \begin{cases} 1 & \text{if the } i\text{th applicant is black,} \\ 0 & \text{otherwise.} \end{cases}$$

- (c) `summary(fitGAM)`

This leads to a table that shows the significance of each predictor. Note that we are doing *non-Bayesian* inference here and the last column corresponds to p-values. We see, for example, that being single significantly elevates the probability of a mortgage being denied after controlling for each of the other variables.

- (d) `par(mfrow=c(1,2))`

```
plot(fitGAM, shade=TRUE, shade.col="palegreen", select=1, xlim=c(0,1))
plot(fitGAM, shade=TRUE, shade.col="palegreen", select=2, xlim=c(0,1))
```

This shows that the two continuous predictors have interesting non-linear effects of the probability of denial.

- (e) Make sure that the file `BostonMortgagesViaStan.R` is in the current working directory of your R session.

Issue the command `source("BostonMortgagesViaStan.R")`

Depending on the capacity of your computer, this will take a few minutes to run. The script eventually produces two plots:

- Markov chain Monte Carlo summary graphics for  $\beta_3, \dots, \beta_{11}$  and the standard deviation parameters  $\sigma_{u1}$  and  $\sigma_{u2}$  that control spline penalization for the estimates of  $f_1$  and  $f_2$ . For ease of interpretation, the first column has the description of the indicator variable corresponding to each  $\beta_j$ .
  - Bayesian estimates of  $f_1$  and  $f_2$  with pointwise 95% credible intervals. The non-linear effects are similar to those obtained using the `mgcv` package (although here we plot them on the probability scale).
- (f) `ORblackVSnonblackMCMC <- exp(beta3MCMC)`  
`hist(ORblackVSnonblackMCMC, col="gold")`  
`mean(ORblackVSnonblackMCMC)`  
`quantile(ORblackVSnonblackMCMC, c(0.025, 0.975))`

These last commands correspond to Bayesian inference for the *odds ratio* corresponding to

$$\frac{\text{odds of a black applicant being denied a mortgage}}{\text{odds of a non-black applicant being denied a mortgage}}$$

after controlling for other factors corresponding to the  $x_{1i}$ ,  $x_{2i}$  and  $x_{4i}, \dots, x_{11i}$  predictors. The *odds* of an applicant being denied a mortgage is defined by

$$\frac{\text{probability denied}}{\text{probability not denied}}.$$

Odds ratios are a commonly used quantity in medical statistics and epidemiology since they are readily obtained from logistic regression coefficients. For a binary (i.e. “yes”/“no”-type) predictor the corresponding odds ratio being significantly greater than 1 (or significantly below 1 if the ratio is reversed) is interpreted as evidence of the binary predictor being statistically significant.

- (g) **OPTIONAL SUBMISSION TASK LABORATORY 3**  
**(PROBABLY TO BE DONE LATER)**

Open the file `BostonMortgagesViaStan.R` in an editor and change the two lines:

```
nWarm <- 200           # Length of warmup.  
nKept <- 200           # Size of the kept sample.
```

to

```
nWarm <- 1000          # Length of warmup.  
nKept <- 1000          # Size of the kept sample.
```

Save the file and then repeat steps (d) and (e) with these larger Markov chain Monte Carlo sample sizes.

The mortgage data correspond to applicants in the city of Boston, which is the capital of the American state of Massachusetts. Suppose that you are working as a statistician for a social justice section within the Massachusetts state government. Based on the analysis done in this part of Laboratory 3 write a half page memorandum to the head of the section on whether a more thorough investigation into possible racial discrimination by Massachusetts banks should be launched.

**NOTE WELL: Due to the loss of the Week 1 class, the lecturer has made this hand-in task optional. You will receive full credit for Laboratory 3 if you work through as many of the tasks that time permits during the second hour of the 18th October class. Any submissions, which should be done by sending an e-mail message to `matt.wand@uts.edu.au` will be read, and feedback provided.**

- (h) Perhaps later look through the code in `ozoneCalifViaStan.R`, especially that involving Stan model specification, to see how a Bayesian logistic additive model is programmed.

## 5. Arbitrarily Specified Response Distributions

Most of the examples in this short-course involve common response distributions such as the Normal and Bernoulli distributions. In this exercise we explain how arbitrarily specified response distributions can be handled. Provided that the density function has a closed form expression then the

`target +=` command in `rstan` can be used to handle *any* response distribution. Our illustration here involves penalized spline regression with the responses having a *generalized extreme value (GEV)* distribution. The basic form of such a model is

$$y_i | f(x_i) \stackrel{\text{ind.}}{\sim} \text{GEV}(f(x_i), \sigma, \xi)$$

where  $f$  is the usual penalized spline model and the GEV notation is as follows:

$$x \sim \text{GEV}(\mu, \sigma, \xi)$$

and only if the density function of  $x$  is

$$p(x) = \begin{cases} \exp \left[ -\left( \frac{x - \mu}{\sigma} \right) - \exp \left\{ -\left( \frac{x - \mu}{\sigma} \right) \right\} \right] / \sigma, & \xi = 0, \\ \left\{ 1 + \xi \left( \frac{x - \mu}{\sigma} \right) \right\}^{-1/\xi - 1} \exp \left[ - \left\{ 1 + \xi \left( \frac{x - \mu}{\sigma} \right) \right\}^{-1/\xi} \right] / \sigma, & \xi \neq 0. \end{cases}$$

In the  $\xi = 0$  case the above definition of  $p(x)$  applies over all  $-\infty < x < \infty$ . In the  $\xi \neq 0$  case, the expression only applies for  $1 + \xi \left( \frac{x - \mu}{\sigma} \right) > 0$  and the density function is zero otherwise. This distribution is not supported in the current release of `rstan`.

- (a) This task involves running the R script `penSplGEV.R`.
- (b) In `penSplGEV.R` the model is specified for use by `rstan` via the code below. The main new part is the coding of the logarithm of the response density function using the `target +=` command. (**Note:** You do not need to type this code – just browse it!)

```

"data
{
  int<lower=1> n;                int<lower=1> ncZ;
  vector[n] y;                  matrix[n,2] X;
  matrix[n,ncZ] Z;
  real<lower=0> sigmabeta;       real<lower=0> Au;
  real<lower=0> Aeps;           real<lower=0> sigmaxi;
}
parameters
{
  vector[2] beta;               vector[ncZ] u;
  real<lower=0> sigmaeps ;      real<lower=0> sigmau ;
  real xi ;
}
transformed parameters
{
  vector[n] arg;                vector[n] supparg;
  vector[n] logdens;
  arg <- (y - X*beta - Z*u)/sigmaeps;
  supparg <- 1 + xi*arg;
  for (i in 1:n)
  {
    if (xi==0)
      logdens[i] <- -arg[i] - exp(-arg[i]) - log(sigmaeps);
    if (xi!=0)
    {
      if (supparg[i]>0)
      {
        logdens[i] <- -(1/xi)-1*log(1 + xi*(arg[i]))
                      - pow((1 + xi*arg[i]), -(1/xi))
                      - log(sigmaeps);
      }
    }
  }
}
model
{
  for (i in 1:n) target += logdens[i];
  beta ~ normal(0,sigmabeta);    sigmaeps ~ cauchy(0,Aeps);
  sigmau ~ cauchy(0,Au);        xi ~ normal(0,sigmaxi);
}"

```

- (c) Start an R session and make sure that the current working directory corresponds to the directory where `penSplGEV.R` is stored on your computer and issue the command:

```
source("penSplGEV.R")
```

The script will take a few minutes to run. While you are waiting, you may wish to study the code more deeply.

Eventually you will get some plots showing fitting penalized spline and summaries of the MCMC samples of key quantities.

## Advantages of the Bayesian Approach to Generalized Additive Model Fitting

As we have seen in Laboratory 3 there are at least two ways to fit generalized additive models in R:

- using non-Bayesian fitting and inference via the package `mgcv`.
- using a Bayesian inference engine via the package `rstan`.

The non-Bayesian approach is quicker to use and produces answers much more quickly. This leads to the question: why use the Bayesian inference engine approach? Some advantages of the latter approach are as follows:

- For binary response data and other non-Gaussian generalized additive models the accuracy of the statistical inference can be poor, compared with Markov chain Monte Carlo-based inference. For example, advertised 95% confidence intervals may not really be 95% in terms of actual coverage percentage.
- The standard generalized additive models have a limited number of distribution families for the response distribution. As we saw with the `penSplGEV.R` script, the Bayesian inference engine approach allows one to specify arbitrary response distributions. A similar limitation of standard regression software is the restriction that the variance is constant (known as *heteroscedasticity*). As we will see in Laboratory 4, the Bayesian inference engine approach can alleviate this assumption and allow the variance to also be a function.
- All of the examples given so far are such that the data are “clean”. Many real data sets are plagued by problems such as missingness. Bayesian inference engines have the capacity to fit generalized additive models that incorporate missing data models.

