Journal of Statistical Software

January 2010, Volume 32, Issue 12. http://www.jstatsoft.org/

Analyzing Temperature Effects on Mortality Within the R Environment: The Constrained Segmented Distributed Lag Parameterization

Vito M. R. Muggeo
Università di Palermo

Abstract

Here we present and discuss the R package **modTempEff** including a set of functions aimed at modelling temperature effects on mortality with time series data. The functions fit a particular log linear model which allows to capture the two main features of mortality-temperature relationships: nonlinearity and distributed lag effect. Penalized splines and segmented regression constitute the core of the modelling framework. We briefly review the model and illustrate the functions throughout a simulated dataset.

Keywords: temperature effects, segmented relationship, break point, P-splines, R.

1. Introduction

Health effects of air temperature are well-known. Some epidemiologic evidence may be found in Braga, Zanobetti, and Schwartz (2001) and Basu and Samet (2002) among the others. Temperature effects have been studied since a long time but in the last decades quantifying temperature effects has become quite important owing to greenhouse effect and consequent climatic changes (e.g., McGeehin and Mirabelli 2001).

The relationship between mortality and temperature is found to be V-shaped in the most of areas around the world: mortality reaches its minimum at some optimal value and increases as temperature gets colder or hotter. The temperature value where mortality reaches its minimum is sometimes referred as minimum mortality temperature and represents the threshold value beyond which mortality increases. Moreover it has been ascertained that the effect is not limited to the same day-exposure t, say, but it is extended to several next days $t+1, t+2, \ldots$ An in-depth analysis of temperature effects on mortality requires to account for the prolonged effects (the so-called *distributed lag* effect) and for nonlinearity (Armstrong

2006).

Muggeo (2008a) presents a unified framework to model the temperature effects on mortality. Let $\mathsf{E}[Y_t] = \mu_t$ be the expected number of deaths for day $t = 1, 2, \ldots, T$, z_t the temperature value, and $x_t^\top$ the vector of additional confounding explanatory variables, such as days of week, holidays, influenza epidemics, for instance. The proposed model assumes $Y_t \sim Pois(\mu_t)$ and

$$\log \mu_t = \boldsymbol{x}_t^\top \boldsymbol{\delta} + \sum_{l_1=0}^{L_1} \beta_{1l_1}(z_{t-l_1} - \psi_1)_- + \sum_{l_2=0}^{L_2} \beta_{2l_2}(z_{t-l_2} - \psi_2)_+. \tag{1}$$

where $(z - \psi_1)_- = (z - \psi)I(z < \psi)$ and $(z - \psi_2)_+ = (z - \psi_2)I(z > \psi_2)$ are two linear spline functions which allow to model the effects of low and high temperatures, respectively below the cold threshold ψ_1 and above the heat threshold ψ_2; L_1 and L_2 are the two maximum lag values selected to assess the delayed effects of cold and heat (typically 15 to 60); $\boldsymbol{x}_t^\top \boldsymbol{\delta}$ contains typical confounders sketched above; finally β_{1l_1} and β_{2l_2} describe the effect of temperature on the response.

More specifically, $\boldsymbol{\beta}_2 = (\beta_{20}, \beta_{21}, \ldots, \beta_{2l_2}, \ldots, \beta_{2L_2})^\top$ expresses the lag-specific log-relative risks for unit increase in temperature greater than the heat threshold ψ_2, namely the risk coming from $0, 1, \ldots, l_2, \ldots, L_2$ days before. Similarly $\boldsymbol{\beta}_1 = (\beta_{10}, \beta_{11}, \ldots, \beta_{1l_1}, \ldots, \beta_{1L_1})^\top$ reflects the lag specific risks of cold understood as temperature below the relevant threshold ψ_1. In short, $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ represent the distributed lag (DL) curve of cold and heat. Model (1) may be simplified by assuming a common threshold for cold and heat, $\psi_1 = \psi_2$, i.e.,

$$\log \mu_t = x_t^\top \delta + \sum_{l_1=0}^{L_1} \beta_{1l_1}(z_{t-l_1} - \psi)_- + \sum_{l_2=0}^{L_2} \beta_{2l_2}(z_{t-l_2} - \psi)_+. \tag{2}$$

Regardless the number of thresholds notice that lag-varying risks are allowed, while the breakpoints of the segmented relationship, i.e., the thresholds, are constrained to be the same across the lags; for this reason, we call the parameterization in models (1) and (2) the *constrained segmented distributed lag parameterization*, hereafter CSDL.

To obtain plausible and reasonable findings, the model also assumes that the DL curves are smooth functions. At this aim, the beta parameters are expressed by means of linear combinations of B-spline bases,

$$\boldsymbol{\beta}_1 = \boldsymbol{C}\boldsymbol{b}_1 \qquad \boldsymbol{\beta}_2 = \boldsymbol{H}\boldsymbol{b}_2 \tag{3}$$

where $\boldsymbol{C} = [C_1, \ldots, C_{P_1}]$ and $\boldsymbol{H} = [H_1, \ldots, H_{P_2}]$ are the two B-spline bases respectively of rank equal to P_1 and P_2 with relevant coefficients $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ (Eilers and Marx 1996; Wood 2006). To complete specification of the DL curves, a penalty term is imposed on the DL coefficients. The overall penalty $J(\boldsymbol{\lambda})$ is

$$J(\boldsymbol{\lambda}) = \lambda_1 \boldsymbol{b}_1^\top \boldsymbol{D}_1^\top \boldsymbol{D}_1 \boldsymbol{b}_1 + \lambda_2 \boldsymbol{b}_2^\top \boldsymbol{D}_2^\top \boldsymbol{D}_2 \boldsymbol{b}_2 + \omega_1 \boldsymbol{b}_1^\top \boldsymbol{C}^\top \boldsymbol{\Upsilon}_1 \boldsymbol{C} \boldsymbol{b}_1 + \omega_2 \boldsymbol{b}_2^\top \boldsymbol{H}^\top \boldsymbol{\Upsilon}_2 \boldsymbol{H} \boldsymbol{b}_2. \tag{4}$$

where $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are difference matrices (Eilers and Marx 1996), $\boldsymbol{\Upsilon}_1$ and $\boldsymbol{\Upsilon}_2$ are two diagonal known weight matrices (Muggeo 2008a). Therefore the DL coefficients are doubly penalised: a standard difference penalty ($\boldsymbol{b}_1^\top \boldsymbol{D}_1^\top \boldsymbol{D}_1 \boldsymbol{b}_1$ and $\boldsymbol{b}_2^\top \boldsymbol{D}_2^\top \boldsymbol{D}_2 \boldsymbol{b}_2$) on the spline coefficients to ensure smoothness over the whole lag range in the spirit of classical P-splines (Eilers and Marx 1996), and an additional varying ridge penalty affecting late DL coefficients to favour the DL curves

approaching to zero at longer lags. Therefore the penalised log-likelihood may be written as $\ell(\boldsymbol{\delta}, \boldsymbol{b}_1, \boldsymbol{b}_2) - J(\boldsymbol{\lambda})$ where $\ell(\cdot)$ is the Poisson log-likelihood.

The smoothing parameter $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \omega_1, \omega_2)^\top$ affects the estimate of all the model parameter, especially $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$, by regulating the smoothness of the DL curves via the spline coefficients $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$. To obtain values of the smoothing parameter $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \omega_1, \omega_2)^\top$, a reasonable approach is to minimise an empirical version of the expected mean square error: for known scale parameter (and specifically equal to one in the Poisson case) we consider the so-called un-biased risk estimator (or scaled AIC) given by (Wood 2006)

$$\text{UBRE} = \frac{1}{n}\{Dev + 2edf - n\}$$

in which $Dev = 2\sum y_i \log(y_i/\hat{\mu}_i)$ is the usual model deviance, and edf are the effective degrees of freedom computed as trace of the hat matrix. Additional measures are available, including the well-known AIC (Akaike information criterion) and BIC (Bayesian information criterion). Selection of $\boldsymbol{\lambda}$ may be carried out efficiently by the method proposed in Wood (2004) and implemented in his **mgcv** package by the function `gam.fit()`.

The estimation procedure which allows to bypass the problems related to the non-regularity of the segmented models (1) or (2) extends the previous work of Muggeo (2003) implemented in the R package **segmented** (Muggeo 2009), and it is described elsewhere (Muggeo 2008a). Details are omitted, but it is important to emphasise that estimation is performed iteratively in terms of the spline coefficients (rather than the DL coefficients) maximising a log-likelihood penalised for the (4), and supplying starting values only for the thresholds.

Poor clear-cut segmented relationships, due to short time series and/or a lot of zeroes in the observed counts and/or and many outliers, can make model estimation difficult; problematic convergence may suggest that the model being fitted is not supported by data (see model `o2` in the section 3). However limited experience on some datasets, shows that these computational troubles are quite unlike in typical time series and the model is successfully fitted most of times. At the convergence estimates of the thresholds and their standard errors are readily available from the model output, while the DL curves are easily obtained using the B-spline bases. For instance, for the cold DL curve we get

$$\hat{\boldsymbol{\beta}}_1 = \boldsymbol{C}\hat{\boldsymbol{b}}_1 \qquad \widehat{\mathsf{COV}}(\hat{\boldsymbol{\beta}}_1) = \boldsymbol{C}\widehat{\mathsf{COV}}(\hat{\boldsymbol{b}}_1)\boldsymbol{C}^\top,$$

and in the same way it is possible to obtain the estimates for the heat curve.

2. Overview of the package

The R package **modTempEff** includes functions to fit the constrained segmented distributed lag model to epidemiological time series of temperature and mortality. The package is written in R code (R Development Core Team 2009), and it is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=modTempEff`. The package depends on the packages **mgcv** and **splines**, and it includes the following functions:

- `tempeff(formula, data, fcontrol, etastart, drop.L, ...)`. This is the main function aimed at estimating the model.

- `csdl(z, psi, L, ridge, ndx, DL, diff.varying)`. This function is employed within the formula of `tempeff()` to set the temperature variable and the arguments necessary to fit a CSDL parameterization.
- `seas(x, ndx)`. This function allows to include in the linear predictor a nonparametric term for the long term trend and seasonality.
- `fit.control(tol, display, it.max, GLM, maxit.inner)`. Auxiliary function relevant to the fitting process.
- `print(x, digits, ...)`, `summary(object, spar, digits, ...)`, and `coef(object, which, L, ...)`. Methods to visualize and to extract the most relevant information of the fit.
- `anova(object, ..., dispersion, test)`. Method to perform model comparisons.
- `plot(x, which, var.bayes, add, delta.rr, level, ...)`. Method to plot the estimated DL curves for cold and heat.

`tempeff()` is used to specify the model: `formula` is the standard formula of the regression equation including confounders (e.g., days of week, influenza epidemics, ...) entering the model linearly, the temperature variable having a constrained segmented distributed lag relationship which has to be specified via the function `csdl()`, and the nonparametric term for long term trend and seasonality. `data` means the possible dataset where the variables are stored, and the control argument `fcontrol` refers to the some options of the fitting process returned `fit.control()`. Starting values may be supplied in `etastart`, and `drop.L` is an integer to specify whether the first `drop.L` observations have to be discarded before fitting. `drop.L` may be useful when several fitted models have to be compared and the same number of observations in each model is desirable, as explained below. The three dots `...` accept arguments to be passed to `csdl()` as discussed below.

Actually `tempeff()` is based on `tempeff.fit()` which is not designed to be called from the user; in turn, `tempeff.fit()` uses `gam.fit()` from the **mgcv** package and `splineDesign()` from the **splines** package, both included in the R base distribution. `tempeff()` returns objects of class '`modTempEff`' for which some methods exist as described below.

The function `csdl()` is employed to include in the model a variable having a *csdl* relationship with the response; this variable, specified via its first argument `z`, typically represents the mean or maximum daily temperature or sometimes the 'apparent' temperature accounting for humidity and pressure. The arguments `psi` and `L` are mandatory; one or two starting values have to supplied in `psi` depending on the number of the breakpoints to be estimated, while `L` defines the maximum lags within which to assess the effect of cold and heat, see L_1 and L_2 in formulas (1) and (2); Of course, the first $max(L_1, L_2)$ observations are removed when a CSDL is included. The optional arguments `ndx`, `DL` and `diff.varying` regulate smoothing of DL curves. `ndx` requires two integers (default to `round(L/3)`) to specify the 'apparent' dimension of the B-spline bases for cold and heat (P_1 and P_2 of formula (3)). The user may impose a global difference penalty on the spline coefficients (`DL = FALSE`, default) or on the DL coefficients themselves (`DL = TRUE`): empirical evidence has shown that the two options are unlike to lead different results. The argument `diff.varying` (default to `FALSE`) enables the user to specify a varying difference penalty, in the form $\sum_l (\beta_l - \beta_{l-1})^2 \delta_l$ with

δ_l being a monotonic function of lag l which penalises against large values of differences of DL coefficients. Some simulations have shown this varying difference penalty is substantially unnecessary and even not advised in practice, provided that a varying ridge penalty is used. The additional varying ridge penalties are specified via the argument `ridge` which defaults to `NULL` indicating no ridge penalty. Otherwise `ridge` is a length-two named list of characters written as a function of `l`; for instance two quadratic ridge penalties for both cold and heat may be set via `ridge = list(cold = "l^2", heat = "l^2")`.

The function `seas()` allows to model the long term trend and seasonality in a non parametric way: a (usually rich) B-spline of rank `ndx` is used with a standard second-order difference penalty to prevent undersmoothing.

`fit.control()` allows to control the estimating algorithm, for instance via `tol` to regulate the tolerance value at which the algorithm stops, `display` to print the iterative process, and `it.max` to set the maximum number of the (outer) iterations of the algorithm. Each outer iteration comprises a few inner iterations managed by `maxit.inner` and `GLM` which defaults to `FALSE`. When `GLM = TRUE`, at each iteration an unpenalised GLM is fitted via `glm.fit()`, otherwise `gam.fit()` from **mgcv** is used. `GLM = TRUE` speeds up computations since the smoothing parameter is estimated only at the final iteration, and therefore it may be helpful with very large datasets when `gam.fit()` is unpractical; however some experience suggests to use `GLM = FALSE` to prevent premature convergence to non-optimal solutions.

Finally the methods `print`, `summary`, `coef`, `anova` and `plot` allows to visualize, extract and display the most important information of the fit; in particular `coef` returns `L`+1 coefficients of DL curves for cold and/or heat (depending on `which`), and `plot` portrays the fitted DL curves for cold and/or heat effects (depending on `which`) with pointwise confidence intervals at level `level`.

3. Fitting the model in R

We illustrate the aforementioned functions on a simulated dataset, including daily time series of natural mortality and temperature for five years ($T = 1825$). We load the package via

```
R> library("modTempEff")

Loading required package: mgcv
This is mgcv  1.5-6 . For overview type `help("mgcv-package")'.
Loading required package: splines
```

Notice that `modTempEff` loads the packages **splines** and **mgcv**; the former is employed to build the B-spline bases of formula (3) via the function `splineDesign()`, and the latter is requested to perform 'selection' of the smoothing parameter $\boldsymbol{\lambda}$ and model estimation via the function `gam.fit()`.

The typical dataset employed in the analysis of temperature (or air pollution) effect on health, comprises daily times series of the 'health variable' (mortality counts, all causes or cause-specific) and meteorological/environmental variables. The dataset also includes variables corresponding to day, year, month, and day-of-week. The data being analysed have a similar appearance

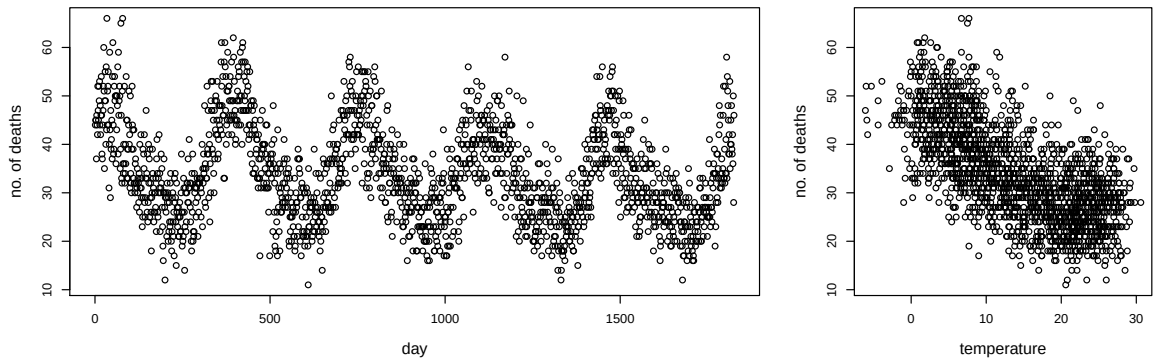

Figure 1: Daily time series of death counts (left) and mortality-temperature scatterplot (right).

```
R> data("dataDeathTemp")
R> head(dataDeathTemp)
```

```
  dec1 mtemp month year day dweek decNS dec2
1   44   0.3     1 1985   1     2    40   33
2   45  -2.2     1 1985   2     3    41   46
3   45  -1.6     1 1985   3     4    44   34
4   37  -0.5     1 1985   4     5    34   32
5   46  -2.2     1 1985   5     6    45   32
6   44  -4.4     1 1985   6     7    50   38
```

The variables `mtemp` and `dec1` are basic for our analysis, as they represent the daily time series of mean temperature and death counts; `dec1` is actually simulated using estimates coming from a real data analysis. `month`, `year`, `day`, and `wday` are 'seasonal' variables respectively for month (12 level categorical variable), year (5 level categorical variable), day (integer $t = 1, 2, \ldots, 1825$) and day of week (7 level categorical variable). Although additional variables, such as humidity or pressure, may be present in the dataset in practice, we neglect them for simplicity; the rationale of discussed methods and relevant code are unchanged.

Figure 1 shows the daily time series of mortality counts and a raw scatterplot of mortality against temperature. Note the classical seasonal pattern in the daily series and the V-shaped relationship mortality-temperature. These figures may be obtained easily via

```
R> layout(matrix(c(1, 1, 2), ncol = 3))
R> with(dataDeathTemp, plot(dec1, xlab = "day", ylab = "no. of deaths"))
R> with(dataDeathTemp, plot(mtemp, dec1, xlab = "temperature",
+    ylab = "no. of deaths"))
```

The interest centers on the temperature effects controlling for confounders, such as seasonality and days-of week, to be included in the regression equation. A starting model could consider the categorical variables year, month and day of week to control for seasonality, while the

temperature effects could be modeled via equation (2) with maximum lags $L_1 = L_2 = 60$; we assume two B-spline bases of rank given by `round(L)/3` (default) and no ridge penalty, i.e., the default `ridge = NULL`; the starting value for the threshold is set to 20 as suggested by the right plot in Figure 1. The option `display = TRUE` in `fit.control()` allows to monitor the estimation process by printing at each iteration the deviance and the current estimate of the threshold.

```
R> o <- tempeff(dec1 ~ day + factor(dweek) + factor(year) + factor(month) +
+    csdl(mtemp, psi = 20, L = c(60, 60)), data = dataDeathTemp,
+    fcontrol = fit.control(display = TRUE))

0   2080.571  ---- without 'csdl' variable
1   1818.663 19.847
2   1818.124 19.844
.....
10  1817.923 19.733
11  1817.918 19.733
12  1817.918 19.732
```

Convergence is attained in twelve iterations and the fitted model is stored in the object `o`.

Following a recently consolidated approach in the analysis of mortality time series, we may improve the fit by including a smoother for seasonality rather than parametric terms for month, year and day; at this aim, we use penalised splines by modifying properly the formula,

```
R> o.noRidge <- update(o, . ~ . - day - factor(year) - factor(month) +
+    seas(day, 30), fcontrol = fit.control(display = FALSE))
```

The smoother used to model the long term trend and seasonality of the observed series is a 'classical' P-spline, that is B-splines with a difference penalty on the coefficients. It should be emphasized that `ndx`, like `ndx` argument in `csdl()`, controls the rank, i.e., the size of the basis used for the penalized spline; the rank or 'apparent' dimension is `ndx`+3 since third degree splines are employed, but the 'actual' dimension, i.e., the effective degrees of freedom (*edf*), are obtained as trace of the hat matrix, and they are typically much less than the corresponding basis size. Note that in the analysis of epidemiological time series, P-splines fitted by direct maximisation of the penalised log-likelihood should be preferred to the alternative nonparametric smoothers fitted by backfitting. The pitfall with the backfitting lies on the so-called 'concurvity' (i.e., a sort of nonparametric collinearity) which leads to biased estimates for the model parameters, especially for the cold effect (Ramsay, Burnett, and Krewski 2003; Muggeo 2004).

A raw inspection of the fitted models via the `print` method may be useful to assess the different fits,

```
R> o

Model Summary (n = 1765):
AIC = 11271.33   BIC = 11486.79   ubre = 0.07457   dev = 1817.918
```

```
Degrees of freedom:
     Model  Cold  Heat Seasonality
edf  39.35  2.92 12.43          NA
rank 70.00 23.00 23.00          NA
```

```
R> o.noRidge
```

```
Model Summary (n = 1765):
AIC = 11255.99   BIC = 11520.07   ubre = 0.06588   dev = 1784.825
Degrees of freedom:
     Model   Cold  Heat Seasonality
edf  48.22  2.841 12.63       25.75
rank 86.00 23.000 23.00       33.00
```

The print method returns the usual model residual deviance with the AIC, BIC, UBRE and also some information on the number of parameters of the model and of any B-spline employed. The rank is the apparent dimension of the bases, i.e., the number of basis functions equal to the number of column of the matrix. The effective degrees of freedom (*edf*) measure the actual model dimension which is reduced owing to penalty. Overall, model `o` (with a design matrix having 70 columns) uses $edf = 39.35$, and model `o.noRidge` (design matrix having 86 columns) with a nonparametric term for seasonality exploits $edf = 48.22$. The DL curves of cold and heat are substantially based on the same *edf*. AIC and UBRE tend to prefer `o.noRidge` with respect to `o`, but the BIC is better for model `o`; we can try to improve the model (such that each likelihood-based criterion is better), by also imposing the fitted DL curves to follow a more plausible biological pattern. Following arguments reported in Muggeo (2008a) and briefly sketched above, we could try to include a ridge penalty to allow the DL curves to approach to zero more rapidly. We set a linear varying ridge penalty on the DL coefficients i.e., $\boldsymbol{\Upsilon}_1 = \boldsymbol{\Upsilon}_2 = \text{diag}(0, 1, 2, 3, \ldots, 60)$ such that the varying ridge wiggliness measures become $\sum_{l_1=0}^{60} \beta_{1l_1}^2 l_1$ and $\sum_{l_2=0}^{60} \beta_{2l_2}^2 l_2$, respectively for cold and heat. The argument `ridge` of `csdl()` has to be employed at this aim, and a natural call makes use of `csdl(mtemp, 20, c(60, 60), ridge = list(cold = "l", heat = "l"))` in the formula of `tempeff()`. However the `...` in `tempeff()` accept arguments to be passed to `csdl()`; therefore we can simply type

```
R> o.Ridge.l <- update(o.noRidge, ridge = list(cold = "l", heat = "l"))
```

and note the formula reads correctly as

```
R> formula(o.Ridge.l)
```

```
dec1 ~ factor(dweek) + seas(day, 30) + csdl(mtemp, psi = 20,
    L = c(60, 60), ridge = list(cold = "l", heat = "l"))
```

Each argument given in `tempeff()` via the `...` is passed to `csdl()` by overwriting its possible previous value; this feature may be useful for the user interested in fitting and comparing different models, for instance by replacing the temperature variable ('apparent' rather than

'ambient' temperature, say) and/or by modifying the starting values for the breakpoint and/or the number of lags `L`.

The effect of the varying ridge penalty is to shrink the late DL coefficients throughout zero. However the amount of shrinkage depends on the weights (main diagonals of $\boldsymbol{\Upsilon}_1$ and $\boldsymbol{\Upsilon}_2$) and on the smoothing parameters ω_1 and ω_2. While smoothing parameters are not modifiable as they are estimated by data, it is possible to increase weights to strengthen the effect of shrinkage. Quadratic or cubic weights lead to results similar to ones returned by a linear ridge (model `o.Ridge.l`) and relevant results are not shown. On the other hand a varying ridge penalty with quartic weights, such as $\sum_{l_1=0}^{60} \beta_{1l_1}^2 l_1^4$ and $\sum_{l_2=0}^{60} \beta_{2l_2}^2 l_2^4$, leads to noteworthy outcome,

```
R> o.Ridge.l4 <- update(o.noRidge, ridge = list(cold = "l^4", heat = "l^4"))
R> o.Ridge.l4

Model Summary (n = 1765):
AIC = 11249.56   BIC = 11445.93   ubre = 0.06223   dev = 1803.121
Degrees of freedom:
     Model   Cold   Heat Seasonality
edf  35.86  1.064  5.269       22.53
rank 86.00 23.000 23.000       33.00
```

Now each likelihood-based criterion (including the BIC) is better than the previous ones; we can use the anova method to compare the different fits using the Mallows' Cp statistic which is closely related to AIC,

```
R> anova(o.noRidge, o.Ridge.l, o.Ridge.l4, test = "Cp")

Analysis of Deviance Table

Model 1: dec1 ~ factor(dweek) + csdl(mtemp, psi = 20, L = c(60, 60)) +
    seas(day, 30)
Model 2: dec1 ~ factor(dweek) + seas(day, 30) + csdl(mtemp, psi = 20,
    L = c(60, 60), ridge = list(cold = "l", heat = "l"))
Model 3: dec1 ~ factor(dweek) + seas(day, 30) + csdl(mtemp, psi = 20,
    L = c(60, 60), ridge = list(cold = "l^4", heat = "l^4"))
  Resid. Df Resid. Dev      Df Deviance     Cp
1    1716.8     1784.8                  1881.3
2    1720.0     1791.3 -3.1849  -6.4826 1881.4
3    1729.1     1803.1 -9.1791 -11.8132 1874.8
```

As expected, the deviance is lower for more complex models (higher *edf*), however both AIC and BIC are lower for the model `o.Ridge.l4` which uses less than 12.36 degrees of freedom as compared with the model with no ridge penalty. In conclusion, P-splines for seasonality and a quartic ridge penalty for the DL curves should be preferred. Of course, different combinations of varying ridge penalty patterns might be used for cold and heat, and comparisons could be made via statistical criteria and/or substantive assessments. We do not include these comparisons or a discussion in the present paper.

We can have a deeper glance of the 'selected' model via `summary()`,

```
R> summary(o.Ridge.14)

Model:  tempeff(formula = dec1 ~ factor(dweek) + csdl(mtemp, psi = 20,
    L = c(60, 60)) + seas(day, 30), data = dataDeathTemp,
    fcontrol = fit.control(display = FALSE),
    ridge = list(cold = "l^4", heat = "l^4"))

Seasonality (smooth):  edf = 22.53 (rank = 33 ; log(spar) = 3.146)

Fit summary (model edf = 35.86; n = 1765):
AIC = 11249.56  BIC = 11445.93  ubre = 0.06223  dev = 1803.121

Net effects of  mtemp (based on edf = 7.33):
           Est  SE.freq SE.bayes rank   edf  L
Cold  0.023091 0.001763 0.003129   23 1.064 60
Heat -0.004343 0.003942 0.005652   23 5.269 60

log(spar) for smooth DL curves:
lambda.Cold lambda.Heat  omega.Cold  omega.Heat
   22.11456    10.03786     1.83512     0.07019

Threshold:
  Est SE.freq SE.bayes
 19.5   0.668    0.684
V variable(s):
         coef        tvalue
-0.0001325005 -0.0001984776
```

Most of the printed information are rather self-explaining, although some points are noteworthy. The V variable shows the estimate and relevant t value of a re-parameterization of the threshold; at the convergence such values should be small. We suggest to warn about fits with large values of coefficients of the V variable.

The reported net effect of temperature is the sum of the lag specific log relative risks for cold and heat. Such synthesis measure is aimed at quantifying the overall effect of cold and heat effects after accounting for possible 'harvesting'. The harvesting occurs when a positive association at short lags (positive lag-specific DL coefficients, typically within seven days) is followed by negative association at longer lags (negative lag-specific DL coefficients) which should suggest a 'deficit' of mortality. From an epidemiological point of view, this would emphasise that the effect of temperature is 'only' to anticipate the deaths by some days, probably affecting more vulnerable people, elderly or suffering persons (e.g., Hajat, Armstrong, Gouveia, and Wilkinson 2005). For the estimates of the net effect and of the threshold, two standard errors are computed. The 'frequentist' ones (`SE.freq`) are based on a sandwich formula involving penalised and unpenalised information matrix assuming fixed the smoothing parameter; the 'bayesian' standard errors (`SE.bayes`) also account, to some extend, for the smoothing parameters and therefore should be preferred as featured by better coverage properties, see Wood (2006) for details. Threshold estimate is also reported along with corresponding standard errors (bayesian and frequentist). Note the breakpoint is actually

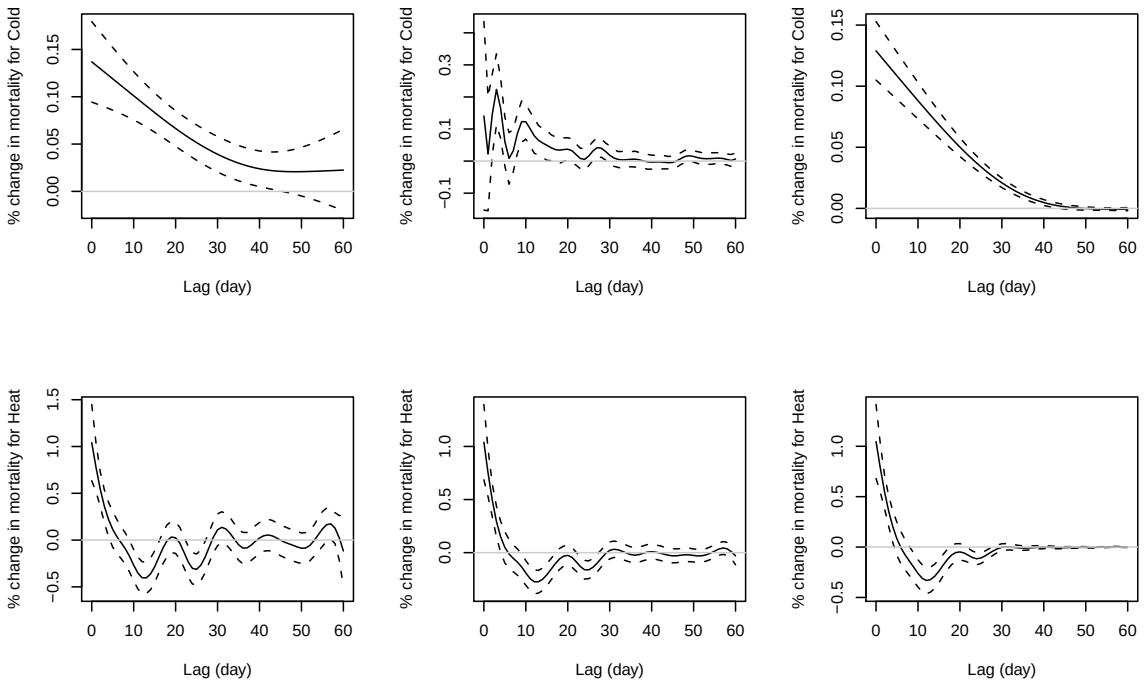

Figure 2: Smoothed Distributed Lag curves for cold (top) and heat (bottom) from three different models. From left to right: only global penalty (`o.noRidge`), global plus linear ridge penalty (`o.Ridge.l`), and global plus quartic ridge penalty (`o.Ridge.l4`).

estimated, and therefore it is included in the overall *df* of the CSDL parameterization given by *df*(cold) plus *df*(heat) plus 1 breakpoint (for model `o.Ridge.l4` it is $1.064+5.269+1 = 7.33$). Lag-specific log relative risk may be extracted via the 'coef' method,

```
R> coef(o.Ridge.l4, L = 7)
```

```
        cold     heat
lag0 0.00129  0.01041
lag1 0.00125  0.00786
lag2 0.00121  0.00554
lag3 0.00117  0.00357
lag4 0.00112  0.00201
lag5 0.00108  0.00079
lag6 0.00104 -0.00012
```

where `L` specifies the number of coefficients to be returned.

It is instructive to compare the fitted DL curves from the three aforementioned models. Figure 2 emphasises the nice end of the additional varying penalty. Plots on the left side show the fitted DL curves using only a global difference penalty. This output is substantially the one of the approach by Zanobetti, Wand, Schwartz, and Ryan (2000), although they deal

with a linear (non-segmented) relationship using different basis and penalty. Note, however, the DL estimated curve (and its standard errors) does not approach to zero at late lags. This implies, for instance, that the estimate of the 'net' effect (sum of lag-specific effects) and corresponding standard error might depend on choice of maximum lag L.

While a simple difference penalty ensures smoothness over the lags, the varying ridge penalty allows the DL curve estimate to approach to zero. Unlike the only difference penalty, the additional varying ridge shrink the DL coefficients and their standard errors towards zero. DL coefficients near to zero at longer lags are biologically plausible since they assume a vanishing effect as lag increases. Moreover this feature makes choice of maximum lag a minor issue.

The plots of DL curves are obtained via the `plot` method

```
R> par(mfcol = c(2, 3))
R> plot(o.noRidge, new = FALSE)
R> plot(o.Ridge.l, new = FALSE)
R> plot(o.Ridge.l4, new = FALSE)
```

Notice the argument `new = FALSE` has been set to display the plot on the current device; otherwise the default value `new = TRUE` would have opened a new device. Additional arguments for the plot method can be used to specify which DL has to be drawn (cold, heat or both of them), the level of the pointwise confidence intervals and which standard errors should be used (frequentist or bayesian). Note when the '`modTempEff`' object has been called without a CSDL term, `plot.modTempEff()` still works by drawing the fitted nonparametric term for seasonality, provided that it has been included in the model. This method also works for fits obtained with fixed (i.e., not estimated) breakpoints via `fcontrol = fit.control(it.max = 0)`.

We conclude the illustration of the code by fitting model (1), namely two different thresholds for cold and heat. The only difference concerns the `psi` argument which now requires two starting values. Thus,

```
R> o2 <- tempeff(dec1 ~ day + factor(dweek) + factor(year) + factor(month) +
+       csdl(mtemp, psi = c(10, 20), L = c(60, 60)),
+       data = dataDeathTemp, fcontrol = fit.control(display = TRUE))

0   2080.571  ---- without 'csdl' variable
1   1835.399 11.818 19.137
2   1827.852 12.197 19.830
3   1828.362 13.126 19.485
.....
19  1816.649 18.027 20.326
20  1816.989 19.363 20.398
Warning message:
max number of iterations attained
```

The algorithm does not converge after 20 iterations; in general, we could also increase the number of iterations or modify the starting values, but usually this does not change the result, see Muggeo (2008b) for a discussion about non convergence in segmented regression.

Broadly speaking, we can interpret such non-convergence as over-fitting, namely the fitted model is not supported by data and a 'bath-type' relationship (see equation (1)) is unlike. However it is always possible to inspect the fit to have a deeper assessment of the results,

```
R> summary(o2)

Model:  tempeff(formula = dec1 ~ day + factor(dweek) + factor(year) +
    factor(month) + csdl(mtemp, psi = c(10, 20), L = c(60, 60)),
    data = dataDeathTemp, fcontrol = fit.control(display = TRUE))

Seasonality (smooth):  NA

Fit summary (model edf = 40.07; n = 1765):
AIC = 11271.85  BIC = 11491.28  ubre = 0.07486  dev = 1816.989

Net effects of  mtemp (based on edf = 17.07):
           Est  SE.freq SE.bayes rank    edf  L
Cold  0.036031 0.003646 0.003666   23  2.916 60
Heat -0.008224 0.012516 0.012539   23 12.155 60

log(spar) for smooth DL curves:
lambda.Cold lambda.Heat
      20.32        9.74

Threshold:
      Est SE.freq SE.bayes
psi1 19.4   1.489    1.497
psi2 20.4   0.663    0.666
V variable(s):
     coef1      coef2    tvalue1    tvalue2
1.33571366 0.07128425 0.89700879 0.10748561
```

There are several indications to discard this two-breakpoints model. First, point estimates of the threshold are very close each other, with corresponding confidence intervals strongly overlapped. Second, and more importantly, the AIC, BIC and UBRE are somewhat higher. Similar convergence problems occur when we try to estimate two breakpoints in the previously 'selected' model (`o.Ridge.l4`) with a nonparametric term for seasonality and an additional varying ridge penalty to smooth the DL curves.

```
R> o2.Ridge.l4 <- update(o.Ridge.l4, psi = c(10, 20),
+       fcontrol = fit.control(it.max = 30))

Warning message:
max number of iterations attained

R> o2.Ridge.l4
```

```
Model Summary (n = 1765):
AIC = 11250.39   BIC = 11463.90   ubre = 0.0627   dev = 1797.685
Degrees of freedom:
     Model   Cold   Heat Seasonality
edf  38.99  1.138  5.546       24.31
rank 87.00 23.000 23.000       33.00
```

Due to the additional breakpoint to be estimated, note the model rank is 87, one more than the one of `o.Ridge.l4`; however AIC, BIC and UBRE do not improve.

We do not discuss further the selection between one- or two-breakpoints models, and following results reported in Tiwari, Cronin, Davis, Feuer, Yu, and Chib (2005), we suggest of using the BIC; at this aim the anova method includes the option `test="BIC"`,

```
R> anova(o.Ridge.l4, o2.Ridge.l4, test = "BIC")

Analysis of Deviance Table

Model 1: dec1 ~ factor(dweek) + seas(day, 30) + csdl(mtemp, psi = 20,
    L = c(60, 60), ridge = list(cold = "l^4", heat = "l^4"))
Model 2: dec1 ~ factor(dweek) + seas(day, 30) + csdl(mtemp, psi = c(10,
    20), L = c(60, 60), ridge = list(cold = "l^4", heat = "l^4"))
  Resid. Df Resid. Dev     Df Deviance    BIC
1    1729.1     1803.1                 2071.2
2    1726.0     1797.7 3.1308   5.4357 2089.2
```

Note the BIC returned by `anova.modTempEff` is actually computed as $Dev + \log(n) \cdot edf$ which is numerically different from the ones by the print and summary methods, $-2\ell + \log(n) \cdot edf$; however findings from model comparisons are unchanged.

The dataset shipped with the package also includes two additional simulated response counts: `decNS` which is not associated with `mtemp`, and `dec2` which is associated via a CSDL parameterization with two breakpoints. The user may try to fit models with these responses and to assess different results.

4. Conclusion

We have discussed the practical implementation of a log-linear regression model to analyse the temperature effects on mortality with (daily) time series data. The model is estimated via penalised log-likelihood by means of the efficient `gam.fit()` function in the **mgcv** package. Estimates of distributed lag effect of the cold and heat, and threshold values are returned, along with additional linear parameters and a smoothing term to account for long term trend and seasonality.

There are several sides where the model and the package may be improved, specifically with regard to the effect of air pollution, e.g., particulate matter or ozone. Currently the pollutant may enter the model linearly in the `formula` of `tempeff()`, however it would be interesting to model it via an additional distributed lag effect with its proper maximum lag, size of the spline basis, and smoothing parameter to be estimated from data. Modelling pollutant

via a separate DL does not pose particular problems and its implementation appears rather practicable: this feature could be included in the next release of the package. A much more challenging improvement would be to model the synergic effect of temperature and pollutant via two bivariate DL curves, cold-by-pollutant and heat-by-pollutant. The idea has been discussed in Muggeo (2007) assuming a fixed breakpoint ψ, but further investigation is needed to modify the model framework and the estimating procedure when the breakpoints have to be estimated. Another possible and non-straightforward extension of the package concerns the so-called case-crossover studies where each event day is matched to several control days according to a specified design (e.g., Janes, Sheppard, and Lumley 2005). The constrained segmented distributed lag parameterization may be still applied in theory, but the regression model to fit is not longer a log-linear model for count response but a conditional logit model with a binary response applied to an opportunely augmented dataset. Therefore fitting the constrained segmented distributed lag parameterization would rely on a different function, perhaps the `clogit()` from package **survival** (Therneau and Lumley 2009). More generally, the present package may be employed to model data from different fields; if model (1) or (2) hold and the response variable belongs to exponential family, **modTempEff** may be customized by modifying the `family` argument of the call to `gam.fit()`.

However in its current implementation, at time of writing the model has been successfully employed in the analysis of temperature and mortality in Santiago and Palermo, two cities with different climatic conditions (Muggeo and Hajat 2009), and it is hoped that the package may be helpful for researchers involved in epidemiological studies of mortality and temperature.

Acknowledgments

I would like to thank the anonymous referees and the Editor Achim Zeileis for their constructive comments on the manuscript and the package itself. This work was partially supported by grant 'Fondi di Ateneo (ex 60%) 2007' prot. ORPA07J7R8.

References

Armstrong B (2006). "Models for the Relationship Between Ambient Temperature and Daily Mortality." *Epidemiology*, **17**, 624–631.

Basu R, Samet JM (2002). "Relation Between Elevated Ambient Temperature and Mortality: A Review of the Epidemiologic Evidence." *Epidemiological Reviews*, **24**, 190–202.

Braga AL, Zanobetti A, Schwartz J (2001). "The Time Course of Weather Related Deaths." *Epidemiology*, **12**, 662–667.

Eilers PHC, Marx BD (1996). "Flexible Smoothing with B-Splines and Penalties." *Statistical Science*, **11**, 89–121.

Hajat S, Armstrong BG, Gouveia N, Wilkinson P (2005). "Mortality Displacement of Heat Related Deaths: a Comparison of Delhi, Sao Paulo, and London." *Epidemiology*, **16**, 613–620.

Janes H, Sheppard L, Lumley T (2005). "Case-Crossover Analyses of Air Pollution Exposure Data: Referent Selection Strategies and Their Implications for Bias." *Epidemiology*, **16**, 717–726.

McGeehin M, Mirabelli M (2001). "The Potential Impacts of Climate Variability and Change on Temperature-Related Morbidity and Mortality in the United States." *Environmental Health Perspectives*, **109**, 185–9.

Muggeo VMR (2003). "Estimating Regression Models with Unknown Break-Points." *Statistics in Medicine*, **22**, 3055–3071.

Muggeo VMR (2004). "A Note on Temperature Effect Estimate in Mortality Time Series Analysis (Letter to Editor)." *International Journal of Epidemiology*, **33**, 1151–1153.

Muggeo VMR (2007). "Bivariate Distributed Lag Models for the Analysis of Temperature-by-Pollutant Interaction Effect on Mortality." *Environmetrics*, **18**, 231– 243.

Muggeo VMR (2008a). "Modeling Temperature Effects on Mortality: Multiple Segmented Relationships with Common Break Points." *Biostatistics*, **9**, 613–620.

Muggeo VMR (2008b). "**segmented**: An R Package to Fit Regression Models with Broken-Line Relationships." *R News*, **8**(1), 20–25. URL http://CRAN.R-project.org/doc/Rnews/.

Muggeo VMR (2009). ***segmented**: Segmented Relationships in Regression Models*. R package version 0.2-6, URL http://CRAN.R-project.org/package=segmented.

Muggeo VMR, Hajat S (2009). "Modelling the Nonlinear Multiple-Lag Effects of Ambient Temperature on Mortality in Santiago and Palermo: A Constrained Segmented Distributed Lag Approach." *Occupational and Environmental Medicine*, **66**, 584–591.

Ramsay TO, Burnett RT, Krewski D (2003). "The Effect of Concurvity in Generalized Additive Models Linking Mortality to Ambient Particulate Matter." *Epidemiology*, **14**, 18–23.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Therneau T, Lumley T (2009). ***survival**: Survival Analysis Including Penalised Likelihood*. R package version 2.35-7, URL http://CRAN.R-project.org/package=survival.

Tiwari RC, Cronin KA, Davis W, Feuer EJ, Yu B, Chib S (2005). "Bayesian Model Selection for Join Point Regression with Application to Age-Adjusted Cancer Rates." *Applied Statistics*, **54**, 919–939.

Wood SN (2004). "Stable and Efficient Multiple Smoothing Parameter Estimation for Generalized Additive Models." *Journal of American Statistical Association*, **99**, 637–686.

Wood SN (2006). *Generalized Additive Models: An Introduction with R*. Chapman & Hall, Boca Raton, Florida.

Zanobetti A, Wand MP, Schwartz J, Ryan LM (2000). "Generalized Additive Distributed Lag Models: Quantifying Mortality Displacement." *Biostatistics*, **1**, 279–292.

Affiliation:

Vito M. R. Muggeo
Dipartimento Scienze Statistiche e Matematiche 'S. Vianelli'
Università di Palermo
90128 Palermo, Italy
E-mail: vito.muggeo@unipa.it
URL: http://dssm.unipa.it/vmuggeo/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 32, Issue 12 *Submitted:* 2008-12-16
January 2010 *Accepted:* 2009-12-11