MortalitySmooth: An R Package for Smoothing Poisson Counts with P-Splines

Carlo G. Camarda

Max Planck Institute for Demographic Research

Abstract

The **MortalitySmooth** package provides a framework for smoothing count data in both one- and two-dimensional settings. Although general in its purposes, the package is specifically tailored to demographers, actuaries, epidemiologists, and geneticists who may be interested in using a practical tool for smoothing mortality data over ages and/or years. The total number of deaths over a specified age- and year-interval is assumed to be Poisson-distributed, and P-splines and generalized linear array models are employed as a suitable regression methodology. Extra-Poisson variation can also be accommodated.

Structured in an S3 object orientation system, **MortalitySmooth** has two main functions which fit the data and define two classes of objects: `Mort1Dsmooth` and `Mort2Dsmooth`. The methods for these classes (`print`, `summary`, `plot`, `predict`, and `residuals`) are also included. These features make it easy for users to extract and manipulate the outputs. In addition, a collection of mortality data is provided.

This paper provides an overview of the design, aims, and principles of **MortalitySmooth**, as well as strategies for applying it and extending its use.

Keywords: smoothing, mortality, P-splines, Poisson data, overdispersion.

1. Introduction

In recent decades, advanced smoothing methods have been developed for analyzing and modeling data when parametric forms for describing functions are unreasonably rigid. Developers of statistical software have since implemented most of these methodologies, thereby broadening the user-base beyond the statistical circle.

These smoothing methods can be applied to demography in general, and to mortality analysis in particular. Mortality changes are crucial issues in various fields, and understanding mortality development is a key factor in actuarial studies, as well as in epidemiology and genetics.

Because demography studies often use data on whole populations, mortality studies are commonly performed at an aggregate level of analysis. Moreover, mortality dynamics essentially develop over both age and period in an eventual two-dimensional frame. Over the last two centuries, researchers have sought to reduce the dimensionality of the mortality data using both simple parametric functions and overparameterized models. See Forfar, McCutcheon, and Wilkie (1988) and Girosi and King (2008) and the references given there.

Because mortality developments generally display regular patterns, using smoothing approaches is a more natural choice for analyzing mortality changes than imposing a model structure. In this paper, we present a specific smoothing approach, P-splines, and an R package, which is tailored to model aggregate mortality data in both one- and two-dimensional settings: **MortalitySmooth** (Camarda 2012).

Several methodologies have been proposed for smoothing both one- and two-dimensional data structures. Smoothing methods for two-dimensional problems have, for example, been proposed by Cleveland and Devlin (1988), who used a generalization of the "loess" methodology; and by Dierckx (1993) and De Boor (2001), who employed a two-dimensional regression basis as the Kronecker product of B-splines. Gu and Wahba (1993) and Wood (2003) fitted surfaces with thin plate splines. Other researchers have demonstrated that two-dimensional data structures can be embedded in the framework of generalized additive models (GAMs, Hastie and Tibshirani 1990).

Packages and functions for fitting such approaches are already available in R (R Development Core Team 2012). The function `loess` in the **stats** package can fit locally polynomial surfaces, while the **locfit** package (Loader 2010) can fit local regression and likelihood models. Multivariate functions for estimating splines are provided by the **gss** package (Gu 2012). Finally, GAMs can be fitted using the **mgcv** package developed by Wood (2006).

An appealing methodology for smoothing data was developed using two-dimensional regression splines; specifically, B-splines with penalties, known as P-splines. Eilers and Marx (1996) have dealt with unidimensional regression, while extensions for bivariate regression have been presented in Eilers and Marx (2002b), Currie, Durban, and Eilers (2004, 2006) and Eilers, Currie, and Durban (2006). This approach can deal with aggregate data, which may be assumed to be Poisson-distributed, although overdispersion can also be accommodated. Specifically, an important property of P-splines is that they are based on linear regression and generalized linear models (GLMs). This property makes it possible to avoid backfitting and knot selection schemes, as described in Friedman and Silverman (1989) and Kooperberg and Stone (1991, 1992), and also allows for the easy computation of diagnostics. In addition, this approach provides compact results useful for prediction, the computation of standard errors, and fast cross-validation. Without going into details, in the following we will outline the reasons why P-splines are suitable for use in mortality research. P-splines can be also fitted within the R package **mgcv**, but in this paper we introduce a user-friendly tool for employing them in the context of mortality analysis. In addition, we incorporated the generalized linear array models (GLAMs) regression methodology, which can speed up the estimation procedure by orders of magnitude relative to a direct evaluation (Currie *et al.* 2006).

The paper is structured as follows. A description of one- and two-dimensional P-splines is provided in Section 2. Here we give a brief but thorough introduction to the method, and provide key references for additional information. Information about the selection of the smoothing parameters and about the case of overdispersed counts is also provided. From

Section 3 onwards, we present the package itself. First, we demonstrate how users can select the data that are included in the package. Examples that show the potential of the package for smoothing counts over one- and two-dimensions are given in Section 4. Although most of the package's features can be used in either setting, we describe how to extrapolate, construct confidence intervals, and account for overdispersion in the unidimensional case. In Section 4.2, we illustrate interpolation and residuals in the 2D setting. The paper concludes with a critical discussion of the methodology, its performance in the package, and its possible extensions.

2. Introduction to P-splines

While we can easily generalize this approach for any given Poisson-distributed data, here we will formulate the model for mortality data. For a given population, at age i during year j, the total number of deaths, Y_{ij}, follows a Poisson distribution, $Y_{ij} \sim \mathcal{P}(E_{ij} \cdot \mu_{ij})$ (Keiding 1990). The expected values are thus the product of exposures E_{ij} and the hazard (or force of mortality) μ_{ij}. In large populations, the size of the exposure population is typically estimated by averaging the population sizes at the beginning and at the end of the year. A fully non-parametric estimate of this hazard can be obtained by computing the actual death rates at the respective age and year: $\hat{m}_{ij} = Y_{ij}/E_{ij}$.

For ease of manipulation and presentation, mortality data are commonly prepared as rectangular arrays. For each age and calendar year, we have the number of deaths and the number of exposures arranged in $m \times n$ matrices $\boldsymbol{Y}$ and $\boldsymbol{E}$, respectively. The rows are indexed by age, and the columns are indexed by year. Mortality data can be also viewed in one dimension, such as age (year), by fixing a specific column-year (row-age). In this case, we will have a vector of death counts, $\boldsymbol{y}$, and exposures, $\boldsymbol{e}$.

MortalitySmooth employs P-splines for smoothing this type of data. Eilers and Marx (1996) developed a method which combines (fixed-knot) B-splines with a roughness penalty. Currie *et al.* (2004) extended the initial idea to modeling and forecasting mortality, and a comprehensive study of the methodology was presented in Currie *et al.* (2006).

B-splines are bell-shaped curves composed of smoothly joined polynomial pieces of degree q. The points on the horizontal axis at which the pieces merge together are called "knots." We will use equally spaced knots. For details on B-splines and related algorithms, see De Boor (1978) and Dierckx (1993).

In the **MortalitySmooth** package, the function `MortSmooth_bbase` creates internally an equally-spaced B-splines basis over the abscissa of the data. The function reproduces an algorithm presented by Eilers and Marx (2010) using differences of truncated power functions. In the section "Splines and knots," Eilers and Marx (2010) considered the possibility that small errors might be made when using this algorithm, and produced small negative entries in the basis. However, these mistakes are typically very small as a result of computational advances in present-day computers, and, in their example, the largest negative entry was of the order of 10^{-10} which is practically irrelevant in a mortality analysis.

A B-splines basis of degree $q = 3$ with 10 internal knots evaluated over 100 random points in $[0, 1]$ is given by

```
R> library("MortalitySmooth")
R> x <- runif(100)
```

```
R> B3 <- MortSmooth_bbase(x = x, xl = min(x), xr = max(x),
+    ndx = 10, deg = 3)
```

The number of columns of a B-splines basis, k, is the sum of the the number of internal knots and the associated polynomial degree q. In our example: $k = 10 + 3 = 13$.

2.1. P-splines in one dimension for Poisson data

In the one-dimensional case, equally-spaced B-splines can be used as a regression basis: $\boldsymbol{B} \in \mathbb{R}^{m \times k}$. The coefficients, $\boldsymbol{a}$, will be associated with each B-spline. We then model our Poisson data as follows:

$$\boldsymbol{\eta} = \ln(E(\boldsymbol{y})) = \ln(\boldsymbol{e}) + \ln(\boldsymbol{\mu}) = \ln(\boldsymbol{e}) + \boldsymbol{B}\boldsymbol{a}\,, \tag{1}$$

where $\boldsymbol{\eta}$ is the linear predictor, and, when dealing with Poisson data, a logarithm is used as a link-function. The logarithm of the exposures, $\ln(\boldsymbol{e})$, is commonly called an offset.

Following the approach outlined by Eilers and Marx (1996), we can choose a relatively large number of B-splines with an additional penalty, $\boldsymbol{P}$, on the regression coefficients. Whereas the large basis would result in over-fitting, the penalty forces the coefficients to vary more smoothly.

A penalized version of the iteratively reweighted least squares (IRWLS) algorithm yields estimates of the coefficients $\boldsymbol{a}$:

$$(\boldsymbol{B}^\top \tilde{\boldsymbol{W}} \boldsymbol{B} + \boldsymbol{P})\tilde{\boldsymbol{a}} = \boldsymbol{B}^\top \tilde{\boldsymbol{W}} \tilde{\boldsymbol{z}}\,, \tag{2}$$

where $\tilde{\boldsymbol{z}} = (\boldsymbol{y} - \boldsymbol{e}\tilde{\boldsymbol{\mu}})/(\boldsymbol{e}\tilde{\boldsymbol{\mu}}) + \boldsymbol{B}\tilde{\boldsymbol{a}}$, which is defined as a working dependent variable. $\boldsymbol{B}$ is again the regression matrix, $\tilde{\boldsymbol{\mu}}$ and $\tilde{\boldsymbol{a}}$ denote current approximations to the solution, and $\tilde{\boldsymbol{W}}$ is a diagonal matrix of weights. In the case of Poisson errors, as in Equation 3.7 in Currie *et al.* (2004), $\tilde{\boldsymbol{W}} = \text{diag}(\boldsymbol{e}\tilde{\boldsymbol{\mu}})$.

The only departure from the standard procedure for fitting GLM with B-splines as regressors is the modification of $\boldsymbol{B}^\top \tilde{\boldsymbol{W}} \boldsymbol{B}$ by $\boldsymbol{P}$. This penalty is weighted by a positive regularization parameter λ, and is given by

$$\boldsymbol{P} = \lambda \, \boldsymbol{D}_d^\top \, \boldsymbol{D}_d\,. \tag{3}$$

The matrix $\boldsymbol{D}_d$ constructs dth order differences of the coefficients: $\boldsymbol{D}_d \, \boldsymbol{a} = \Delta^d \boldsymbol{a}\,.$

MortalitySmooth computes $\boldsymbol{D}_d$ as the (repeated) differencing of the identity matrix. For example, $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are constructed as follows, when $k = 5$:

```
R> D1 <- diff(diag(5), diff = 1)
R> D2 <- diff(diag(5), diff = 2)
```

By changing λ, the smoothness can be tuned. Hence, the parameter λ controls the trade-off between smoothness and model fidelity. A higher λ will lead to a higher penalty term, and, consequently, to smoother fitted values. Conversely, with a small λ, the unpenalized term gains importance and the smooth curve will be closer to the data.

It is noteworthy that, as pointed by Eilers and Marx (2002a), (i) the number of equally spaced knots does not matter much, provided enough of them are chosen to ensure greater flexibility than is needed, and that (ii) the choice of the degree of the polynomials q for constructing the B-splines is almost irrelevant in the case of P-spline models.

2.2. P-spline models for two-dimension mortality data

In order to model mortality data in arrays, we will seek to construct a basis for a two-dimensional regression with local support analogous to $\boldsymbol{B}$ in Equation 2. A detailed description of this generalization can be found in Eilers and Marx (2002b); Currie *et al.* (2004, 2006); Eilers *et al.* (2006).

For the purposes of regression, we assume that the data are arranged as a column vector; that is, $\boldsymbol{y} = \text{vec}(\boldsymbol{Y})$. Accordingly, we arrange the matrices of exposures $\boldsymbol{E}$ so that $\boldsymbol{e} = \text{vec}(\boldsymbol{E})$. Let $\boldsymbol{B}_a$, $m \times k_a$ be the regression matrix of B-splines based on age $\boldsymbol{x}_a$, and, similarly, let $\boldsymbol{B}_y$, $n \times k_y$ be the regression matrix of the explanatory variable for year $\boldsymbol{x}_y$. The regression matrix for our two-dimensional model is the Kronecker product

$$\boldsymbol{B} = \boldsymbol{B}_y \otimes \boldsymbol{B}_a \,. \tag{4}$$

As in the unidimensional case, the number of columns in $\boldsymbol{B}_a$ and $\boldsymbol{B}_y$ is related to the number of knots chosen for the B-splines. Following the same concept as in the unidimensional case, we will use a relatively large number of equally spaced B-splines over both domains.

In a regression setting, the matrix $\boldsymbol{B}$ has an associated vector of regression coefficients $\boldsymbol{a}$ of length $k_a k_y$. Therefore, the model can be written as

$$\boldsymbol{\eta} = \ln(E(\boldsymbol{y})) = \ln(\boldsymbol{e}) + \ln(\boldsymbol{\mu}) = \ln(\boldsymbol{e}) + (\boldsymbol{B}_y \otimes \boldsymbol{B}_a)\boldsymbol{a} = \ln(\boldsymbol{e}) + \boldsymbol{B}\boldsymbol{a}. \tag{5}$$

If we arrange the elements of $\boldsymbol{a}$ in a $k_a \times k_y$ matrix $\boldsymbol{A}$, where $\boldsymbol{a} = \text{vec}(\boldsymbol{A})$, Equation 5 can be written as

$$\ln(E(\boldsymbol{Y})) = \ln(\boldsymbol{E}) + \ln(\boldsymbol{M}) = \ln(\boldsymbol{E}) + \boldsymbol{B}_a \boldsymbol{A} \boldsymbol{B}_y^\top \,. \tag{6}$$

From the definition of the Kronecker product, we can independently penalize the coefficients over the rows and columns of $\boldsymbol{A}$. Let the difference matrices, $\boldsymbol{D}_a$ and $\boldsymbol{D}_y$, act on the columns and rows of $\boldsymbol{A}$, respectively. The penalized IRWLS in Equation 2 can still be used with the following two-dimensional penalty:

$$\boldsymbol{P} = \lambda_a(\boldsymbol{I}_{k_y} \otimes \boldsymbol{D}_a^\top \boldsymbol{D}_a) + \lambda_y(\boldsymbol{D}_y^\top \boldsymbol{D}_y \otimes \boldsymbol{I}_{k_a}) \,, \tag{7}$$

where λ_a and λ_y are the smoothing parameters used for age and year, respectively. $\boldsymbol{I}_{k_a}$ and $\boldsymbol{I}_{k_y}$ are identity matrices of dimension k_a and k_y, respectively. More details can be found in Currie *et al.* (2004).

As in the unidimensional setting, B-splines provide enough flexibility to capture surface trends. The additional penalty reduces the effective dimension, leading to a rather parsimonious model with a smoothed, fitted surface. Another advantage of using two-dimensional P-splines is that doing so allows us to choose different smoothing parameters over ages and years, which makes the model very flexible.

2.3. A generalized linear array model approach

In theory, we can use Equation 2 to estimate $\boldsymbol{a}$ in a two-dimensional setting as well. This is possible when the problems are of moderate size. But with mortality data, the problems can be relatively large. For instance, with 90 ages and 70 years, and assuming $\boldsymbol{B}_a$ has $90/5 = 18$ and $\boldsymbol{B}_y$ has $70/5 = 14$ columns, the regression matrix would be $(m \cdot n) \times (k_a \cdot k_y) = 6300 \times 252$.

Even the computation of $\boldsymbol{B}^\top \tilde{\boldsymbol{W}} \boldsymbol{B}$ is challenging in the context of two dimensional smoothing and, in general, the penalized IRWLS algorithm quickly runs into storage and computational difficulties.

In order to overcome these issues, Eilers *et al.* (2006) and Currie *et al.* (2006) propose an algorithm that takes advantage of the special structure of both the data as an rectangular array, and the model matrix as a tensor product. They call their model the generalized linear array model, or GLAM.

This concept can be illustrated in the computation of $\boldsymbol{\mu} = \mathrm{vec}(\boldsymbol{M})$ in two dimensions, as in Equation 6, in which the matrix of force of mortality is constructed without using the large Kronecker product basis. In this manner, we save both space and time.

In a similar fashion, the inner product for the two-dimensional setting in Equation 2 can be obtained efficiently.

We define the function $G(\boldsymbol{X})$ as the row tensor function for a $n \times c$ matrix $\boldsymbol{X}$:

$$G(\boldsymbol{X}) = [\boldsymbol{X} \otimes \mathbf{1}^\top] \cdot [\mathbf{1}^\top \otimes \boldsymbol{X}], \quad n \times c^2,$$

where $\mathbf{1}$ is a vector of 1s of length c.

Following Equation 2.6 in Currie *et al.* (2006), the inner products in Equation 2 for a two-dimensional setting can then be written as

$$(\boldsymbol{B}_y \otimes \boldsymbol{B}_a)^\top \boldsymbol{W}(\boldsymbol{B}_y \otimes \boldsymbol{B}_a) \equiv G(\boldsymbol{B}_a)^\top \, \breve{\boldsymbol{W}} \, G(\boldsymbol{B}_y), \tag{8}$$

where $\breve{\boldsymbol{W}}$ is the $m \times n$ matrix of weights, i.e., $\mathrm{vec}(\breve{\boldsymbol{W}}) = \mathrm{diag}(\boldsymbol{W})$.

Equation 8 has two important computational properties. On the right-hand side, we avoid the computation of the potentially large matrix $\boldsymbol{B} = \boldsymbol{B}_y \otimes \boldsymbol{B}_a$. Moreover, the number of multiplications on the right-hand side is much smaller than the number on the left-hand side.

Section 4 in Currie *et al.* (2006) provides details of the rearranging and the re-dimensioning of the right-hand side that are required to give the left-hand side exactly. This procedure is done internally in **MortalitySmooth**. In the following, we present a small example in which the computing times and the object sizes are compared for the direct and the GLAM algorithms: the left-hand and the right-hand sides of Equation 8.

First we create artificial axes and associated B-spline bases:

```
R> a <- 1:50
R> m <- length(a)
R> y <- 1:100
R> n <- length(y)
R> Ba <- MortSmooth_bbase(x = a, xl = min(x), xr = max(x), ndx = 7, deg = 3)
R> ka <- ncol(Ba)
R> By <- MortSmooth_bbase(x = y, xl = min(y), xr = max(y), ndx =17, deg = 3)
R> ky <- ncol(By)
R> w <- runif(m * n)
```

The last line creates a vector of pseudo-weights. The direct computation of the inner product in Equation 2 could be as follows, with the related computing time:

```
R> system.time(tBWB1 <- t(B) %*% (w * B))
```

```
   user  system elapsed
  0.184   0.012   0.195
```

The GLAM algorithm requires the construction of each tensor product of the right-hand side of Equation 8, as well as the arrangement of the weights in $\breve{W}$. Moreover, in **MortalitySmooth** the function `MortSmooth_BWB` recovers the left-hand side by re-dimensioning these elements, which is a very efficient operation.

```
R> Ba1 <- kronecker(matrix(1, ncol = ka, nrow = 1), Ba)
R> Ba2 <- kronecker(Ba, matrix(1, ncol = ka, nrow = 1))
R> Ga <- Ba1 * Ba2
R> By1 <- kronecker(matrix(1, ncol = ky, nrow = 1), By)
R> By2 <- kronecker(By, matrix(1, ncol = ky, nrow = 1))
R> Gy <- By1 * By2
R> Wbreve <- matrix(w, m, n)
R> system.time(tBWB2 <- MortSmooth_BWB(Ga, Gy, ka, ky, Wbreve))
```

```
   user  system elapsed
  0.004   0.000   0.005
```

The CPU times reported were estimated on a 2.0 GHz portable PC with an Intel Core 2 Duo T6400 processor and 4 GB of RAM. The GLAM algorithm clearly outperforms the plain procedure in terms of computing speed. Furthermore, we extracted and compared the sizes of the objects used in each approach, and found that the GLAM approach greatly reduces memory requirements by orders of magnitude:

```
R> size1 <- object.size(list(B, w))
R> size2 <- object.size(list(Ga, Gy, ka, ky, Wbreve))
R> size1/size2
```

```
20.0775316139601 bytes
```

2.4. Smoothing parameter selection

In a P-spline approach, the trade-off between parsimony and accuracy is clearly driven by the choice of the smoothing parameter(s). We need to balance the ability to reproduce observed data and to achieve small effective dimensions.

In a GLM framework, a common measure of discrepancy is the deviance. For Poisson counts, it takes the following form:

$$\text{Dev} = 2 \sum \left\{ \boldsymbol{y} \cdot \ln\left(\frac{\boldsymbol{y}}{\hat{\boldsymbol{y}}}\right) \right\} \tag{9}$$

On the other side, we define the effective dimension, or the degrees of freedom of the model to be

$$\text{ED} = \text{tr}(\boldsymbol{H}). \tag{10}$$

We refer the reader to Hastie and Tibshirani (1990, p. 52) for a fuller treatment of this problem. The hat-matrix $\boldsymbol{H}$ can be computed from the estimated linearized smoothing problem in Equation 2:

$$\boldsymbol{H} = \boldsymbol{B}(\boldsymbol{B}^\top \hat{\boldsymbol{W}} \boldsymbol{B} + \boldsymbol{P})^{-1} \boldsymbol{B}^\top \hat{\boldsymbol{W}} \,, \tag{11}$$

where $\hat{\boldsymbol{W}}$ contains the weights of the last iteration after convergence. By fixing a value of λ or a combination (λ_a, λ_y), both deviance and the effective dimension can be computed in both a one-dimensional and in a two-dimensional setting.

In a P-spline framework, we need some way of choosing an "optimal" value(s) for the smoothing parameter(s) which can balance bias and variance in the model construction. Eilers and Marx (1996, 2002a) have suggested using the Akaike information criterion (AIC, Akaike 1973). The expression for AIC is given by

$$\text{AIC}(\lambda) = \text{Dev} + 2 \cdot \text{ED} \,. \tag{12}$$

Alternatively, one can use the Bayesian information criterion (BIC, Schwarz 1978), which penalizes model complexity more heavily than AIC. In a two-dimensional setting,

$$\text{BIC}(\lambda_a, \lambda_y) = \text{Dev} + \ln(m \cdot n) \cdot \text{ED} \,. \tag{13}$$

Once the smoothing parameter(s) has been selected, the system of equations described in (2) has a unique solution. As has already been pointed out by Currie *et al.* (2004, p. 285), a stiffer fit, which is given by the BIC, is preferred when modeling mortality data with P-splines. Therefore, **MortalitySmooth** will have this criterion as the default option.

In practice, based on a given information criterion, two functions are employed for searching for the optimal smoothing parameter in the one-dimensional setting, and the (λ_a, λ_y) combination used in the two dimensional setting: `Mort1Dsmooth_optimize` and `Mort2Dsmooth_optimize`.

Instead of conducting a plain grid-search over a wide range(s) of possible smoothing parameter(s), these functions make use of `cleversearch` from the R package **svcm** (Heim 2009). This approach efficiently explores a one(two)-dimensional space of the smoothing parameter(s), optimizing each smoothing parameter in turn, moving at most one grid step up or down.

The smoothing parameter selection is made in two separate steps, and, in a unidimensional setting, two arguments control this optimization procedure: `TOL2` and `RANGE`. The former argument controls the difference between the two adjacent smoothing parameters in the grid search, and by default is set to 0.5 on a logarithmic scale. `RANGE` contains the range of smoothing parameters in which the grid search is applied, and is set by default to $(10^{-4}, 10^6)$, taken on the log-scale.

First, we search over the complete `RANGE` using a rough grid (four times `TOL2`) and the median of `RANGE` as an initial value of λ. Next, the procedure searches in the restricted range around the sub-optimal smoothing parameter, using a finer grid defined by `TOL2`. This procedure allows us to find a precise smoothing parameter in an efficient manner: we do not explore the full range of possible λ values, moving at most one grid step up or down. Furthermore, the two-step routine reduces the risk of finding a sub-optimal smoothing parameter. In two-dimensional cases, the controlling arguments are defined over both domains.

The default ranges and tolerance levels have been proven to be adequate for many applications in mortality research. Nevertheless, two issues could arise when working with objective information criteria: (1) the information criterion may not present a clear minimum, and (2) the information criterion may not provide reasonable outcomes. The optimization procedure in **MortalitySmooth** attempts to reduce the risk associated with this first problem by evaluating the selected criterion over many values of the smoothing parameter(s). In the latter case, neither BIC nor AIC might be suitable, mainly because the data present features that do not fit with the assumptions of these information criteria, such as strong seasonal effects. As a valid alternative, the user could design a personal grid search, or compute a specific information criterion that is more in accordance with the data in hand.

2.5. Overdispersed Poisson data

We frequently find that the Poisson assumption, which was mentioned above, can be relatively strong in demographic and actuarial data, and that, in particular situations, the presence of overdispersion cannot be neglected. Breslow (1984) and Cameron and Trivedi (1986) are among those who have recognized that counts may display extra-Poisson variation, or overdispersion, relative to a Poisson model.

However, the P-spline methods can be modified and adapted to this new situation. Replacing the classic Poisson assumption, we allow the variance to be adjusted as a function of the mean. Specifically, we have the variance proportional to the mean. The additional dispersion parameter will then be the constant of proportionality. In the formula, we have for the unidimensional case

$$\text{Var}(\boldsymbol{y}) = \psi^2 \, v(\boldsymbol{e\mu}) \,.$$

where $v(\cdot)$ is the variance function and ψ^2 is the dispersion parameter. For the simple Poisson case, we would have $v(\boldsymbol{e\mu}) = \boldsymbol{e\mu}$ and $\psi^2 = 1$.

Using this generalization, the penalized IRWLS algorithm is modified and a quasi-likelihood estimation is performed. The classic works in this area are by McCullagh and Nelder (1989) and Cameron and Trivedi (1998). For a recent treatment from a P-splines perspective, we refer the reader to Djeundje and Currie (2011).

It is worth pointing out here that, regardless of whether maximizing the penalized likelihood with B-splines basis leads to the system of equations in (2), the penalized quasi-likelihood version is given by

$$(\boldsymbol{B}^\top \tilde{\boldsymbol{W}}_{\psi^2} \boldsymbol{B} + \boldsymbol{P}) \tilde{\boldsymbol{a}} = \boldsymbol{B}^\top \tilde{\boldsymbol{W}}_{\psi^2} \tilde{\boldsymbol{z}} \tag{14}$$

where the diagonal weight matrix is scaled by the dispersion parameter ψ^2

$$\tilde{\boldsymbol{W}}_{\psi^2} = \tilde{\boldsymbol{W}}/\psi^2 \,.$$

Since the working variable $\tilde{\boldsymbol{z}}$ is not modified by ψ^2, the overdispersion parameter will have an effect on the estimated coefficients $\boldsymbol{a}$. For instance, with overdispered data, i.e., $\psi^2 > 1$, the resulting fitted values will be smoother.

In a one-dimensional setting and for a fixed smoothing parameter, the dispersion parameter can be computed as follows:

$$\psi^2 = \frac{\text{Dev}}{m - \text{ED}} \,, \tag{15}$$

but an information criterion is needed for estimating the smoothing parameter. Replacing the log-likelihood in both AIC and BIC with the quasi-likelihood, we obtain:

$$\text{AIC} = \frac{\text{Dev}}{\psi^2} + 2 \cdot \text{ED} \qquad\qquad \text{BIC} = \frac{\text{Dev}}{\psi^2} + \ln(m) \cdot \text{ED} \,. \qquad (16)$$

For a two-dimensional setting, m is substituted with $m \cdot n$ in both Equation 15 and Equation 16. It is easy to see that, the larger the ψ^2, the smaller the first term in both AIC and BIC is. Such a result clearly leads to smoother outcomes.

In **MortalitySmooth**, overdispersion can be accommodated. In this case, the smoothing parameter(s) is first optimized with $\psi^2 = 1$, and then the dispersion parameter is computed by Equation 15. Given the updated ψ^2, a criterion from Equation 16 is used for re-optimizing the smoothing parameter(s). This sequence of steps continues until convergence. Details of this iterative procedure are given in Section 3 of Djeundje and Currie (2011).

3. Data selection

As was mentioned above, P-splines are suitable for smoothing any kind of Poisson-distributed data. However, **MortalitySmooth** is specifically tailored to demographers and actuaries conducting mortality research. The package therefore provides examples and some data from this area. The user can supply external data, as well, provided they are conformable.

Data from four countries (Denmark, Japan, Sweden, and Switzerland) are contained in the object `HMDdata`. The data are taken from the Human Mortality Database (2011). Matrices of age-specific population, deaths, exposures, and death rates are provided.

For a given country, type of data, and sex, the data have the same number of rows (ages 0 to 110), whereas the ranges of years are subject to availability. `HMDdata` are accessible either manually or simply through using the function `selectHMDdata`. In the following, we present the latter approach to selecting the death rates of Danish females from ages 50 to 100, and from 1950 to 2009:

```
R> x <- 50:100
R> y <- 1950:2009
R> mydata <- selectHMDdata(country = "Denmark", data = "Rates",
+    sex = "Females", ages = x, years = y)
```

We create an `HMDdata` object: an indexed matrix with its own attributes. Within the package, these object `mydata` can be directly plotted as follow:

```
R> mydata

          1950        1951        1952
50 0.005733326 0.005169963 0.005274262 [...]
51 0.005738661 0.005425875 0.004788330 [...]
[...]
attr(,"country-data-sex")
[1] "Denmark-Rates-Females"
attr(,"class")
[1] "HMDdata"
```
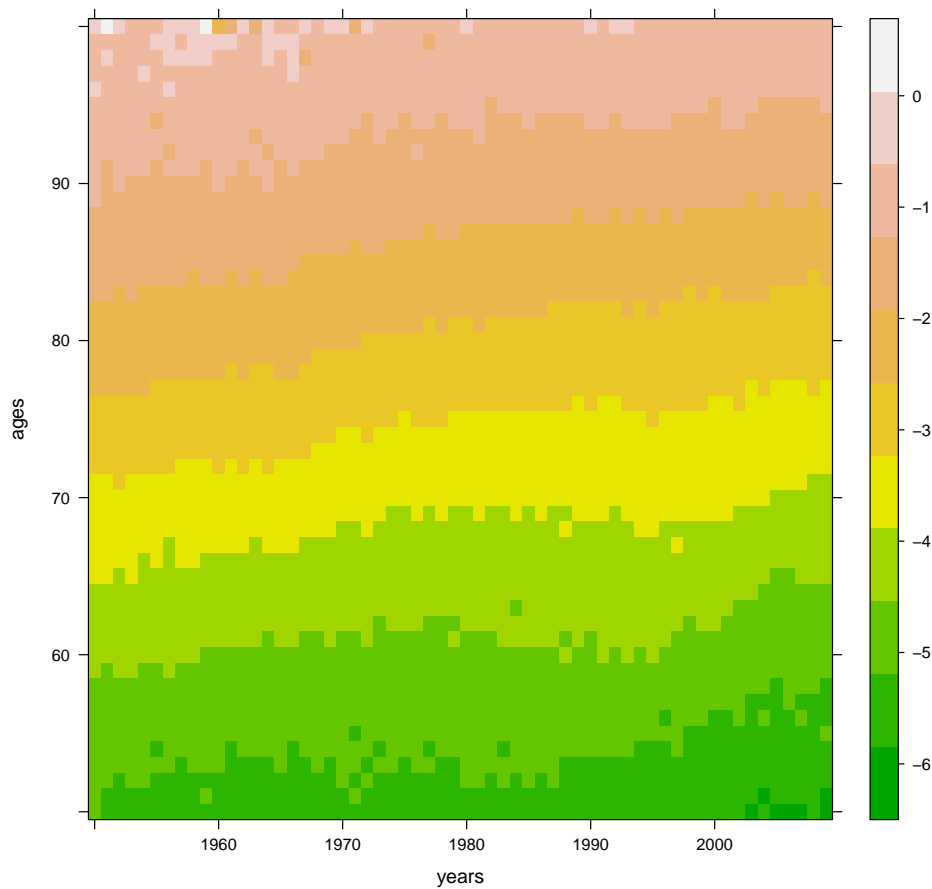
Figure 1: Actual death rates, logarithmic scale. Denmark, females, ages 50–100, 1950 to 2009.

```
R> plot(mydata, cut = 10, col.regions = terrain.colors(11))
```

The function `plot` will recognize the dimension(s) of the data and reproduce either a simple plot or a shaded contour map for one- and two-dimensional data, respectively. By default, while death counts are plotted on the original scale, death rates are presented on a log-scale. Figure 1 presents the outcome for the Danish data previously extracted.

4. Smoothing with MortalitySmooth

The main functions for smoothing in **MortalitySmooth** are `Mort1Dsmooth` and `Mort2Dsmooth`; they are designed to smooth one- and two-dimensional Poisson counts, respectively. They are both implemented in an object-oriented design using the standard S3 paradigm.

Internally, these functions use rich B-splines bases for ensuring flexibility (one knot every five data-points). By default, polynomials with $q = 3$ are chosen for constructing B-splines and the penalty matrix has $d = 2$. However, the number and the degree of the B-splines, as well as the order of differences, can be adjusted by the user.

4.1. Smoothing counts in one dimension

The function `Mort1Dsmooth` returns an object of class `Mort1Dsmooth`, which is a P-splines smoothing of the input data of a degree and order (eventually) fixed by the user.

The main arguments are an abscissa of the data (`x`), a set of count response variable values series (`y`), and the offset term (`offset`). In a mortality context, the first argument would be either the ages or the years under study. Death counts would be the response, `y`. As shown in Section 2.1, the offset is the logarithm of the exposure population.

To provide an example, we smooth mortality data from Danish males at age 60 from 1945 to 2000:

```
R> x <- 1945:2000
R> y <- selectHMDdata("Denmark", "Deaths", "Males", ages = 60, years = x)
R> e <- selectHMDdata("Denmark", "Exposures", "Males", ages = 60, years = x)
R> fit1D <- Mort1Dsmooth(x = x, y = y, offset = log(e))
R> fit1D

 Call:
 Mort1Dsmooth(x = x, y = y, offset = log(e))

 Number of Observations            : 56
 Effective dimension               : 6.358
 (Selected) smoothing parameter    : 316.23
 Bayesian Information Criterion (BIC): 98.671
```

The object `fit1D` contains several attributes, and little information can be printed directly; e.g., the optimal λ is equal to 316.23, and we model the 56 observations using a smoothing function with an effective dimension of 6.358. Internally, the object contains additional information which could be directly extracted, such as the following deviance:

```
R> fit1D$deviance

[1] 73.0779
```

Given such figures, the user could manually compute the overdispersion parameter ψ^2 from Equation 15: $73.0779/(56 - 6.358) = 1.472098$. However, this value, along with others, could be read directly from the resulting `summary` of the model:

```
R> summary(fit1D)

 Call:
 Mort1Dsmooth(x = x, y = y, offset = log(e))

 Number of Observations            : 56
 Effective dimension               : 6.358
 (Selected) smoothing parameter    : 316.23
 Bayesian Information Criterion (BIC) : 98.671
```

```
Akaike's Information Criterion (AIC)    : 85.794
(Estimated) dispersion parameter (psi^2): 1.47

Residuals:
     Min        1Q   Median        3Q       Max
-2.07555  -0.97930  -0.08075   0.75848   2.75227

Settings and control:
  number of B-splines     : 14
  degree of the B-splines: 3
  order of differences    : 2
  convergence tolerance   : 1.6507e-09
```

As noted above, `Mort1Dsmooth` uses BIC as the default information criterion for selecting the smoothing parameter. The user can adopt alternative approaches using the argument `method`:

```
R> fit1Daic <- Mort1Dsmooth(x = x, y = y, offset = log(e), method = 2)
R> fit1Dsub <- Mort1Dsmooth(x = x, y = y, offset = log(e), method = 3,
+    lambda = 10000)
R> fit1Dfix <- Mort1Dsmooth(x = x, y = y, offset = log(e), method = 4,
+    df = 8)
```

The model `fit1Daic` optimizes λ by AIC (`method = 2`). For the object `fit1Dsub`, $\lambda = 10000$ is subjectively provided by the user (`method = 3`). Additionally, `Mort1Dsmooth` can select an optimal smoothing parameter for a given value of ED: this is the case for `fit1Dfix`, in which we opt for a smooth function with ED equal to eight.

The **MortalitySmooth** package provides several support functions for a `Mort1Dsmooth` object. For instance, we can directly plot `fit1D`, obtaining the actual death rates on a log-scale along with the fitted rates[1]:

```
R> plot(fit1D)
```

We can add to this plot log-rates from `fit1Daic`, `fit1Dsub`, and `fit1Dfix`, which can be extracted directly (see Figure 2):

```
R> lines(x, fit1Daic$logmortality, col = 3, lwd = 2)
R> lines(x, fit1Dsub$logmortality, col = 4, lwd = 2)
R> lines(x, fit1Dfix$logmortality, col = 5, lwd = 2)
R> legend("bottom", c("Actual", "BIC", "AIC", "Subjective", "Fix df=8"),
+    col = 1:5, pch = c(1, -1, -1, -1, -1), lwd = c(1, 2, 2, 2, 2),
+    lty=c(0, 1, 1, 1, 1), bty = "n")
```

Figure 2 shows how well P-splines are able to capture the development of mortality over time. Moreover, the degree of smoothness selected by the BIC certainly makes it possible to be

---

[1] Mortality development is commonly described by log-rates. An alternative option is given in `plot.Mort1Dsmooth`.
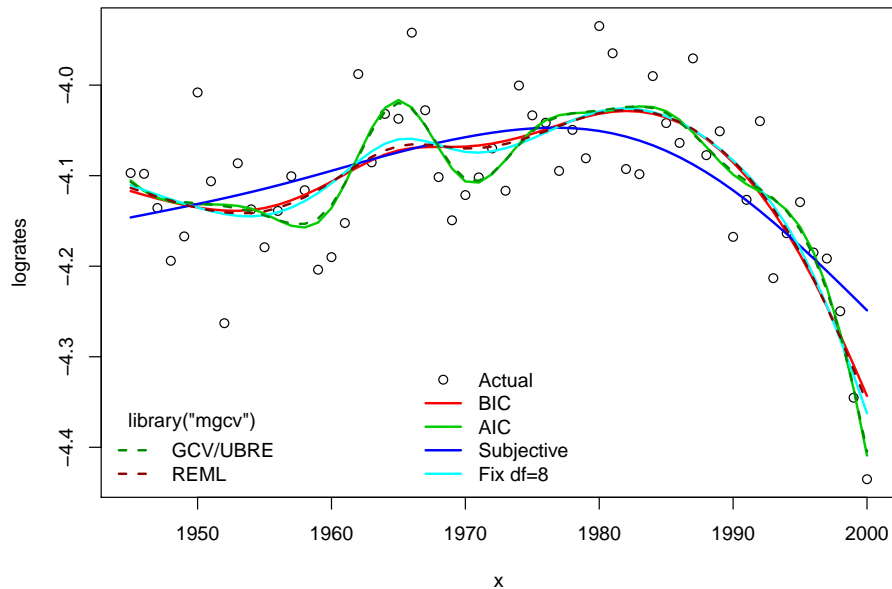
Figure 2: Actual and fitted death rates from a P-spline approach, logarithmic scale. Different criteria for selecting λ. Denmark, males, age 60, 1945 to 2000. The outcomes from **mgcv** are also plotted.

reasonable faithful to the data (see how well the slight increase in mortality in the 1960s and 1970s is portrayed), without leading to an overestimation of small fluctuations, which are likely due to randomness.

In Figure 2, we also plot the log-mortality from analogous models fitted with **mgcv** (Wood 2006). Given the same number of B-splines, we use the default GCV/UBRE criterion and the REML algorithm to select the degree of smoothness. The code for fitting such models is given by

```
R> library("mgcv")
R> kk <- fit1D$ndx + fit1D$deg
R> mgam <- gam(y ~ offset(log(e)) + s(x, bs = "ps", k = kk),
+    family = "poisson")
R> mgam2 <- gam(y ~ offset(log(e)) + s(x, bs = "ps", k = kk),
+    family = "poisson", method = "REML")
```

Whereas the default criterion in **mgcv** leads to estimates similar to P-splines with AIC, the log-mortality fitted by the REML optimization is rather similar to the P-splines with BIC. To plot these outcomes in R:

```
R> lines(x, log(fitted(mgam)/e), lwd = 2, lty = 2, col = "green4")
R> lines(x, log(fitted(mgam2)/e), lty = 2, lwd = 2, col = "red4")
R> legend("bottomleft", c("GCV/UBRE", "REML"), col = c("green4", "red4"),
+    title = "library(\"mgcv\")", lwd = c(2, 2), lty = c(2, 2), bty = "n")
```
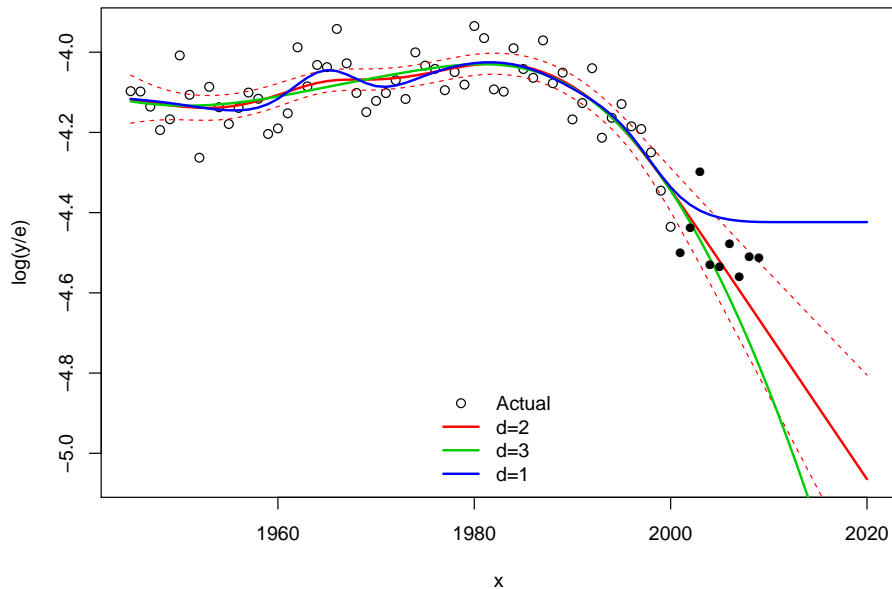
Figure 3: Actual, fitted, and forecasted death rates from a P-spline approach, with a logarithmic scale. λ selected by BIC based on the actual data. Different values of d. The 95% confidence intervals are included for $d = 2$. Denmark, males, age 60, 1945 to 2000. The actual death rates from 2001 to 2009 are depicted with dark black dots.

Extrapolation

The function `predict.Mort1Dsmooth` can be used to compute standard errors for response and log-rates from `Mort1Dsmooth` objects. Furthermore, this function allows us to forecast mortality (see Currie *et al.* 2004). The following example takes the model previously fitted to Danish males at age 60 and uses it to produce forecasts to 2020, as well as to compute standard errors for the whole period.

```
R> x.new <- x[1]:2020
R> fit1Dfor <- predict(fit1D, newdata = x.new, se.fit = TRUE)
```

Alternatively, the user could employ the argument `w` and an augmented dataset to manually forecasting mortality. It should be emphasized that the order of difference for the penalty term is crucial when P-splines are employed for extrapolating past trends. Specifically, the B-spline coefficients form a polynomial sequence of degree $d - 1$ in the future trend; i.e., using the default order of difference $d = 2$, we get a linear extrapolation of the last two B-spline coefficients. This induces an approximate log-linear extrapolation of the death rates. We note that extrapolation depends on the spacing of the knots, with smaller spacing leading to a more volatile extrapolation.

To acknowledge differences when forecasting using a diverse order of difference, we estimate and extrapolate the same data using $d = 1$ and $d = 3$; i.e., constant and quadratic polynomials for the future log-rates, respectively.

```
R> fit1D3d <- Mort1Dsmooth(x = x, y = y, offset = log(e), pord = 3)
R> fit1D1d <- Mort1Dsmooth(x = x, y = y, offset = log(e), pord = 1)
R> fit1Dfor3d <- predict(fit1D3d, newdata = x.new)
R> fit1Dfor1d <- predict(fit1D1d, newdata = x.new)
```

The outcomes are presented in Figure 3. See the following paragraph on confidence intervals.

Confidence intervals

In a P-splines approach, standard errors can be obtained using the hat-matrix, and a `predict` function can be used for extracting them from a `Mort1Dsmooth` object. In the following, we extract the standard errors from the object `fit1Dfor`, and we plot actual death rates along with the fitted and forecasted ones and a 95% confidence interval. The outcomes are presented in Figure 3, along with the forecasted death rates from P-splines with $d = 1$ and $d = 3$. We use the 1945–2000 data and predict the 2001–2020 rates. As validation, we also plot the actual rates from 2001 to 2009.

```
R> plot(x, log(y/e), xlim = range(x.new),
+    ylim = range(fit1Dfor$fit, log(y/e)))
R> lines(x.new, fit1Dfor$fit, lwd = 2, col = 2)
R> lines(x.new, fit1Dfor$fit + 2 * fit1Dfor$se.fit, lty = 2, col = 2)
R> lines(x.new, fit1Dfor$fit - 2 * fit1Dfor$se.fit, lty = 2, col = 2)
R> lines(x.new, fit1Dfor3d, lwd = 2, col = 3)
R> lines(x.new, fit1Dfor1d, lwd = 2, col = 4)
R> legend("bottom", c("Actual", "d=2", "d=3", "d=1"), lty = c(0, 1, 1, 1),
+    col = 1:4, pch = c(1, -1, -1, -1), lwd = c(1, 2, 2, 2), bty = "n")
```

Overdispersion

The level of overdispersion in the data for an optimal smoothing parameter is given by the value of ψ^2, Equation 15. From a `Mort1Dsmooth` object we already know the value: 1.47.

A value of ψ^2 larger than one is a sign of the presence of overdispersion. We can account for such a phenomenon simply by setting the argument `overdispersion` equal to `TRUE`:

```
R> fit1Dover <- Mort1Dsmooth(x = x, y = y, offset = log(e),
+    overdispersion = TRUE)
```

As was pointed out in Section 2.5, smoother results are obtained when we account for overdispersion. In other words, a $\psi^2 > 1$ within a quasi-likelihood approach for fitting P-splines leads to a larger smoothing parameter. In the case of the Danish data, we more than triple the parameter λ, and, consequently, decrease the effective dimension:

```
R> c(fit1D$lambda, fit1D$df)
```

```
 [1] 316.227766    6.357878
```

```
R> c(fit1Dover$lambda, fit1Dover$df)
```

```
 [1] 1000.000000    4.819838
```

4.2. Smoothing counts in two dimensions

In order to smooth mortality over both ages and years, another function needs to be employed: `Mort2Dsmooth`. As with the uni-dimensional approach, this function returns an object of class, `Mort2Dsmooth`, which can be used by support functions. In this generalization, the user needs to provide a series of ages and years, as well as two matrices with deaths and exposures. The arguments are named `x` for the abscissa (commonly, the ages), `y` for the ordinate (here, the years), and `Z` for the matrix of death counts over age and years. In this case as well, the log of the matrix of the exposure population is needed as an offset in the model.

In the following example, we extract deaths and exposures for Swedish females, and smooth these data over age and time. The optimal combination of the two smoothing parameters (see Section 2.4) is by default obtained by minimizing the BIC.

```
R> x <- 10:100
R> y <- 1930:2010
R> Y <- selectHMDdata("Sweden", "Deaths", "Females", ages = x, years = y)
R> E <- selectHMDdata("Sweden", "Exposures", "Females", ages = x, years = y)
R> fit2D <- Mort2Dsmooth(x = x, y = y, Z = Y, offset = log(E))
R> fit2D

 Call:
 Mort2Dsmooth(x = x, y = y, Z = Y, offset = log(E))

 Number of Observations               : 7371
                     of which x-axis : 91
                             y-axis : 81
 Effective dimension                  : 131.0869
 (Selected) smoothing parameters
                         over x-axis: 316.23
                         over y-axis: 31.623
 Bayesian Information Criterion (BIC): 11475
```

Additional information on the fitted object can be found in the `summary`. Also, in the two-dimensional setting, the function predict could be used for computing the confidence interval, as well as for predicting values. A shaded contour map of both the actual and the smooth death rates is automatically produced by plotting the `Mort2Dsmooth` object:

```
R> plot(fit2D, palette = "terrain.colors")
```

Figure 4 presents the outcome using the color palette `terrain.colors`. Other color options are available in `?plot.Mort2Dsmooth` . Again, P-splines demonstrate their ability to retain the signals coming from important mortality changes, and, at the same time, to leave out random noise. This feature is particularly useful in a two-dimensional setting: the option to borrow information from neighboring years allows the model to capture the actual trends in the presence of large random fluctuations. Smoothing small populations and cause-specific mortality is easier when this feature is used.
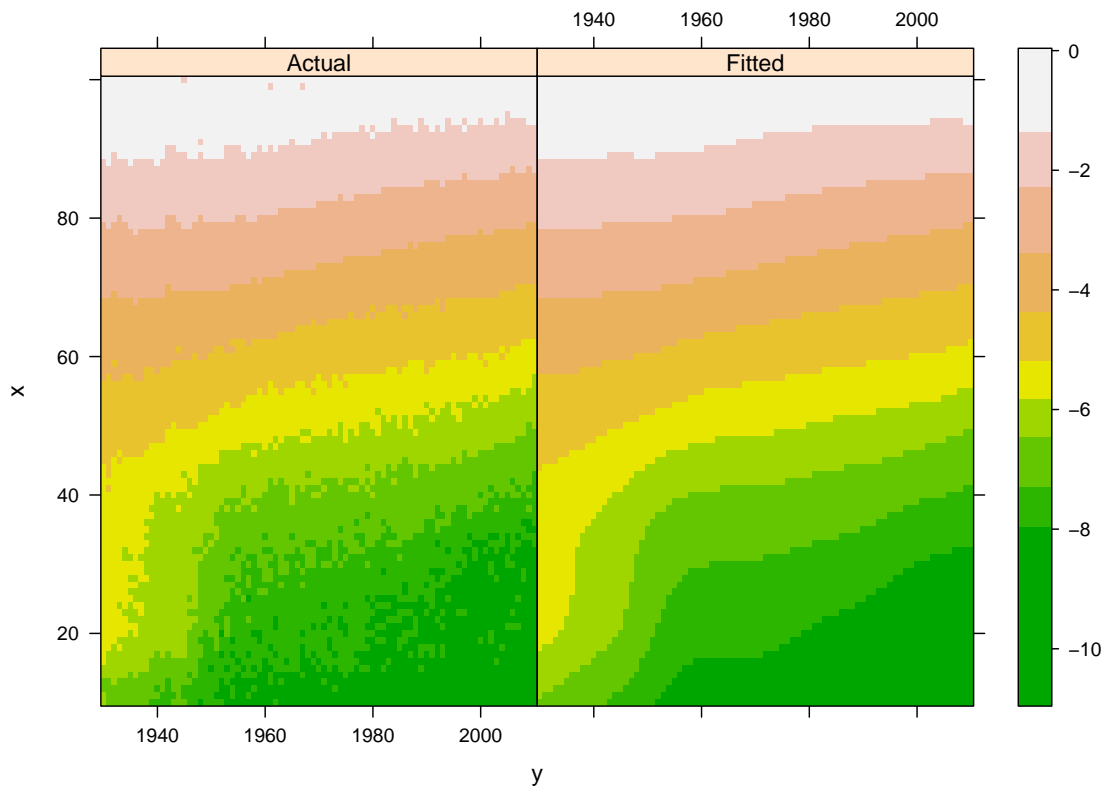
Residuals

Figure 4: Actual and fitted death rates from a P-spline approach, logarithmic scale. λ_a and λ_y selected by BIC. Sweden, females, ages 10–100, 1930 to 2010.

The function `residuals` can be applied to a `Mort2Dsmooth` object as well as in the uni-dimensional setting. Deviance, Pearson, Anscombe and working residuals can be extracted:

```
R> res2D <- residuals(fit2D, type = "pearson")
```

Residuals can be plotted over the predictors of the model. This is done in Figure 5, in which residuals from a two-dimensional P-splines fit for the Swedish example are plotted over age and time. This two-dimensional representation makes it possible to discern where the model cannot capture the actual data. When applied to mortality data, this type of plot may reveal peculiar period shocks and cohort effects. For instance, in Figure 5 most residuals are around the value of zero, without showing any particular systematic features. The only exceptions are the high values during World War II, during which the model overestimates the actual death rates (the negative residuals are represented by dark-pink). In R, this figure is produced by:

```
R> res.grid <- expand.grid(list(x = x, y = y))
R> res.grid$res <- c(res2D)
R> levelplot(res ~ y * x, data = res.grid,
+    at = c(min(res2D), -2, -1, 1, 2, max(res2D)))
```
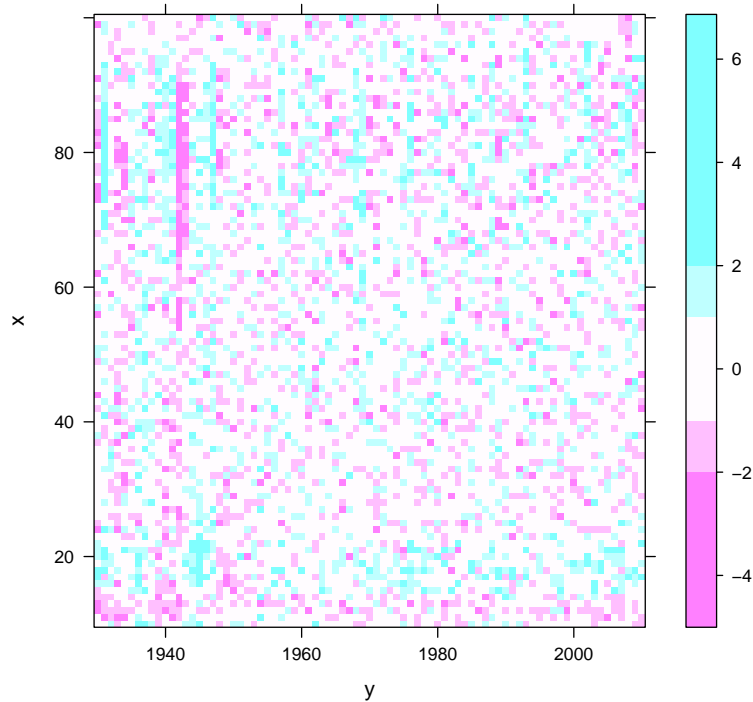
Figure 5: Pearson residuals over ages and years for death rates modeled with 2D smoothing with P-splines. λ_a and λ_y selected by BIC. Sweden, females, ages 10–100, 1930 to 2010.

Interpolation

The weights within the estimation procedure are by default all equal to one. A modification of this argument is allowed, and can be used for extrapolation as well as interpolation. Here, we present an example in which we randomly assign zero-weights to about 80% of the data; i.e., we interpolate the complete surface using only one-fifth of the available information.

```
R> m <- length(x)
R> n <- length(y)
R> set.seed(1)
R> whi0 <- sample(x = 1:(m * n), size = 5900)
R> W <- matrix(1, m, n)
R> W[whi0] <- 0
R> fit2Dint <- Mort2Dsmooth(x = x, y = y, Z = Y, offset = log(E), W = W)

 Warning message:
 Interpolation and/or extrapolation is taking place
```

The warning message informs the user that part of the weights is set equal to zero; i.e., interpolation and/or extrapolation are taking place with the estimation procedure. The function `predict.Mort2Dsmooth` can also be used for interpolating mortality over ages and/or years when a fitted object is already available. Likewise, in the extrapolation example, the
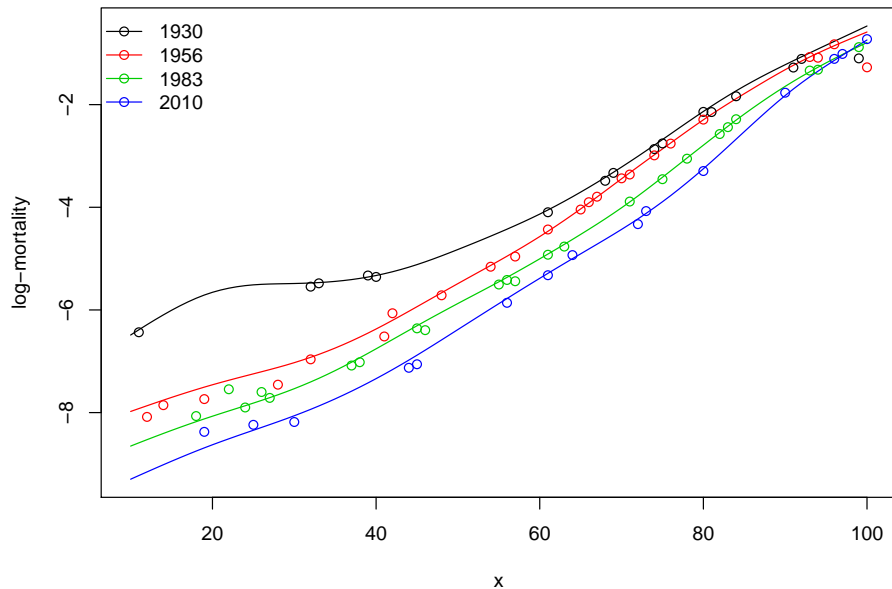
Figure 6: Actual and fitted death rates from a 2D smoothing with P-splines over ages for selected years, logarithmic scale. λ_a and λ_y selected by BIC computed on less than 20% of actual data (randomly chosen). Sweden, females, ages 10–100, 1930 to 2010.

power of penalties becomes clearly visible when interpolating: the B-spline coefficients lead to polynomial sequences of degree $2d - 1$, which means cubic interpolation for the default $d = 2$.

Instead of a shaded contour map, we present the outcomes by plotting a series of (actual and fitted) death rates over age for selected years. We replace the actual rates with zero-weights with NA, since the estimation procedure did not take into account this piece of information:

```
R> lmx.hat <- fit2Dint$logmortality[, c(1, 27, 54, 81)]
R> lmx.act0 <- log(Y/E)
R> lmx.act0[whi0] <- NA
R> lmx.act <- lmx.act0[,c(1,27,54,81)]
R> matplot(x, lmx.hat, type = "l", lty = 1, ylab = "log-mortality")
R> matpoints(x, lmx.act, pch = 1)
R> legend("topleft", legend = y[c(1, 27, 54, 81)], col = 1:4,
+    pch = 1, lty = 1)
```

Figure 6 clearly shows that P-splines are capable of reconstructing mortality developments fairly well using less than 20% of the information from the original data.

5. Conclusions and outlook

In this paper, we present an R package for smoothing mortality using a P-spline approach: **MortalitySmooth**. While the routines can be applied to any Poisson-distributed data, the

emphasis is on mortality data, and the package is structured as a user-friendly tool for demographers, actuaries, epidemiologists, and geneticists who deal with mortality research.

Smoothing death counts is crucial in many demographic analyses, and the outcomes of smoothing methods can be exploited for studies that involve changes in mortality rates, age-structure decompositions, and the construction of continuous life-tables.

We have outlined several reasons for choosing P-splines in this setting. In general, greater flexibility is provided when there are a relatively large number of B-splines in the basis, while a reduction of the effective dimension is neatly produced by an additional penalty on the coefficients of the B-splines. In our mortality analysis, we modeled directly the death counts which are distributed as Poisson in a single age-year interval. Because P-splines can be seen as a generalization of linear regression, this method is useful for smoothing mortality trends. We have also demonstrated that this methodology allows for a comprehensive analysis of the mortality developments over both ages and time with the construction of a two-dimensional basis.

The **MortalitySmooth** package offers two main functions: `Mort1Dsmooth` and `Mort2Dsmooth` for fitting data over one- and two-dimensional settings. A user needs to provide only a few arguments for a default outcome: the death counts, the eventual exposure population, and the domain(s) over which the data are collected. Additional arguments can be used for setting features of the B-spline bases and the penalty term. Nevertheless, it is known that, for a P-spline approach, such choices make little difference. The only exception is the order of difference in cases of interpolation and extrapolation (see Eilers and Marx 2010).

Different criteria for selecting the degree of smoothness in the model are available. Additional support functions help the user to extract and plot outputs, as well as to compute confidence intervals and residuals from fitted objects. Extrapolation and interpolation are also allowed using `predict` functions. In addition, although not shown here, weights can be used to analyze mortality over an age-cohort grid. The user would first need to re-arrange both the deaths and the exposures into suitable matrices, and to construct weights of zero for uncompleted cohorts. Then the package would smooth this structure, assuming smoothness over ages and cohorts. Richards, Kirkby, and Currie (2006) have shown that this structure may lead to a better fit than models based on age and period. Future work will be done that includes the age-cohort approach directly as an argument in the main functions.

In a relational model framework, different populations can be modeled in terms of their distances from a reference population. Biatat and Currie (2010) present a new approach for jointly modeling several populations, allowing for the comparison and classification of mortality in different countries. This model is embedded in the GLAM structure, and we are currently developing an extension of the presented package to incorporate this methodology.

Human mortality generally increases quite smoothly after about age 10, but it shows a steep decline between birth and this age due to infant and child mortality. This pattern is also observed in many other species, and global smoothers over the whole age range do not produce satisfactory outcomes. Camarda, Eilers, and Gampe (2010) present an alternative solution starting from a P-spline approach. Moreover, Kirkby and Currie (2010) have proposed a smoothing model that successfully deals with another specific feature of mortality: period shocks, such as wars, flu epidemics, hot summers, or cold winters. We plan to incorporate into the package these approaches for infant mortality and period shocks.

On the computational side, the last two model extensions can benefit from the implementa-

tion of sparse matrix methods within the GLAM algorithm. We plan to accommodate such methodology within **MortalitySmooth**. We also intend to include additional support functions, and, to provide a more general perspective, other distributions from the exponential family; e.g., binomial and gamma. This revised package will thus be seen as a tool that can be used for estimating generalized linear array models in a comprehensive framework.

Acknowledgments

The author is grateful to both Iain Currie, Paul Eilers and Nadine Ouellette for their useful suggestions, which improved the paper and the interface of the package itself. The author would also like to thank the anonymous reviewers for their valuable comments and advice for revision.

References

Akaike H (1973). "Information Theory and an Extension of the Maximum Likelihood Principle." In B Petrov, F Cźaki (eds.), *Second International Symposium on Information Theory*, pp. 267–281. Akademiai Kiadó, Budapest, Hungary.

Biatat VAD, Currie I (2010). "Joint Models for Classification and Comparison of Mortality in Different Countries." In A Bowman (ed.), *Proceedings of the 25th International Workshop of Statistical Modelling.*

Breslow N (1984). "Extra-Poisson Variation in Log-Linear Models." *Applied Statistics*, **33**, 38–44.

Camarda CG (2012). *MortalitySmooth: Smoothing and Forecasting Poisson Counts with P-splines*. R package version 2.3, URL http://CRAN.R-project.org/package=MortalitySmooth.

Camarda CG, Eilers PHC, Gampe J (2010). "Sums of Smooth Exponentials." In A Bowman (ed.), *Proceedings of the 25th International Workshop of Statistical Modelling*, pp. 113–118.

Cameron AC, Trivedi PK (1986). "Econometric Models Based on Count Data: Comparisons and Applications of Some Estimators and Tests." *Journal of Applied Econometrics*, **1**, 29–53.

Cameron CA, Trivedi PK (1998). *Regression Analysis of Count Data*. Cambridge University Press.

Cleveland WS, Devlin SJ (1988). "Locally Weighted Regression: an Approach to Regression Analysis by Local Fitting." *Journal of the Statistical American Association*, **83**, 597–610.

Currie ID, Durban M, Eilers PHC (2004). "Smoothing and Forecasting Mortality Rates." *Statistical Modelling*, **4**, 279–298.

Currie ID, Durban M, Eilers PHC (2006). "Generalized Linear Array Models with Applications to Multidimensional Smoothing." *Journal of the Royal Statistical Society B*, **68**, 259–280.

De Boor C (1978). *A Practical Guide to Splines.* Springer-Verlag, New York.

De Boor C (2001). *A Practical Guide to Splines.* Revised edition. Springer-Verlag, New York.

Dierckx P (1993). *Curve and Surface Fitting with Splines.* Clarendon Press, Oxford.

Djeundje VAB, Currie ID (2011). "Smoothing Dispersed Counts with Applications to Mortality Data." *Annals of Actuarial Science*, **5**, 33–52.

Eilers PHC, Currie ID, Durban M (2006). "Fast and Compact Smoothing on Large Multidimensional Grids." *Computational Statistics & Data Analysis*, **50**, 61–76.

Eilers PHC, Marx BD (1996). "Flexible Smoothing with *B*-splines and Penalties." *Statistical Science*, **11**, 89–102.

Eilers PHC, Marx BD (2002a). "Generalized Linear Additive Smooth Structures." *Journal of Computational and Graphical Statistics*, **11**, 758–783.

Eilers PHC, Marx BD (2002b). "Multivariate Calibration with Temperature Interaction using Two-Dimensional Penalized Signal Regression." *Chemometrics and Intelligent Laboratory Systems*, **66**, 159–174.

Eilers PHC, Marx BD (2010). "Splines, Knots, and Penalties." *Wiley Interdisciplinary Reviews: Computational Statistics*, **2**, 637–653.

Forfar D, McCutcheon J, Wilkie A (1988). "On Graduation by Mathematical Formula." *Journal of the Institute of Actuaries*, **115**, 1–149.

Friedman JH, Silverman BW (1989). "Flexible Parsimonious Smoothing and Additive Modeling." *Technometrics*, **31**, 3–39.

Girosi F, King G (2008). *Demographic Forecasting.* Princeton University Press.

Gu C (2012). *gss: General Smoothing Splines.* R package version 2.0-9, URL http://CRAN.R-project.org/package=gss.

Gu C, Wahba G (1993). "Semiparametric Analysis of Variance with Tensor Product Thin Plate Splines." *Journal of the Royal Statistical Society B*, **55**, 353–368.

Hastie TJ, Tibshirani RJ (1990). *Generalized Additive Models.* Chapman & Hall.

Heim S (2009). *svcm: 2d and 3d Space-Varying Coefficient Models in R.* R package version 0.1.2, URL http://CRAN.R-project.org/package=svcm.

Human Mortality Database (2011). *University of California, Berkeley (USA), and Max Planck Institute for Demographic Research (Germany).* Available at http://www.mortality.org/ or http://www.humanmortality.de/. (Data downloaded in August 2011).

Keiding N (1990). "Statistical Inference in the Lexis Diagram." *Philosophical Transactions: Physical Sciences and Engineering*, **332**, 487–509.

Kirkby J, Currie ID (2010). "Smooth Models of Mortality with Period Shocks." *Statistical Modelling*, **10**, 177–196.

Kooperberg C, Stone CJ (1991). "A Study of Logspline Density Estimation." *Computational Statistics & Data Analysis*, **12**, 327–347.

Kooperberg C, Stone CJ (1992). "Logspline Density Estimation for Censored Data." *Journal of Computational and Graphical Statistics*, **1**, 301–328.

Loader C (2010). ***locfit***: *Local Regression, Likelihood and Density Estimation*. R package version 1.5-6, URL http://CRAN.R-project.org/package=locfit.

McCullagh P, Nelder JA (1989). *Generalized Linear Model*. 2nd edition. Chapman & Hall, London.

R Development Core Team (2012). *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/.

Richards SJ, Kirkby J, Currie ID (2006). "The Importance of Year of Birth in Two-Dimentional Mortality Data." *British Actuarial Journal*, **12**, 5–61.

Schwarz G (1978). "Estimating the Dimension of a Model." *The Annals of Statistics*, **6**, 461–464.

Wood S (2003). "Thin Plate Regression Splines." *Journal of the Royal Statistical Society B*, **65**, 95–114.

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

Affiliation:

Carlo G. Camarda
Max Planck Institute for Demographic Research
Konrad-Zuse Strasse 1
18057 Rostock, Germany
E-mail: camarda@demogr.mpg.de
URL: http://www.demogr.mpg.de/en/staff/camarda