Smooth models of mortality with period shocks

Running headline: Mortality with period shocks

J G Kirkby, Department of Actuarial Mathematics and Statistics, and the Maxwell Institute for Mathematical Sciences, Heriot-Watt University, Edinburgh, EH14 4AS, UK.
I D Currie, Department of Actuarial Mathematics and Statistics, and the Maxwell Institute for Mathematical Sciences, Heriot-Watt University, Edinburgh, EH14 4AS, UK.

Responsible author: I D Currie, Email: I.D.Currie@hw.ac.uk

Abstract: We suppose that we have mortality data arranged in two-way tables of deaths and exposures classified by age at death and year of death. It is natural to suppose that there is a smooth underlying force of mortality, the mortality surface, that varies with age and year (or period). However, observed mortality is subject to more than stochastic deviation from this smooth surface; for example, flu epidemics, hot summers or cold winters can disproportionately effect the mortality of certain age groups in particular years. We call such an effect a period shock. We describe the mortality surface with an additive model with two components: the underlying smooth surface is modelled with 2-dimensional $P$-splines; the period shocks are modelled with a 1-dimensional $P$-spline in the age direction for each year. This is a large regression model but array methods (Currie et al., 2006) enable the computations to be performed. We illustrate our methods with Swedish mortality data taken from the Human Mortality Database.

Keywords: Generalized linear array model; mortality; $P$-splines; period shock; smoothing.
1 A smooth model of mortality with period shocks

We suppose that we have mortality data arranged in two-way tables of deaths and exposures classified by age at death and year of death. It is natural to suppose that there is a smooth underlying force of mortality, the mortality surface, that varies with age and year (or period). However, observed mortality is subject to more than stochastic deviation from this smooth surface; for example, flu epidemics, hot summers or cold winters can disproportionately effect the mortality of certain age groups in particular years. We call such an effect a period shock.

An example of a period shock is the Spanish flu epidemic of 1918 which affected the mortality of those under the age of 60. We illustrate the extent of this effect with data on Swedish males from the Human Mortality Database; the data runs from 1900 to 2003 and from age 10 to 90. The upper right panel of Figure 1 shows the differences between the logarithms of observed mortality for 1918 and 1919. The remaining panels show the corresponding differences for other years. The 1918/1919 experience is extreme but each panel indicates a systematic age dependent departure from a smoothly changing underlying mortality.

We propose an additive model with two components for the mortality surface: the first component describes a smooth two-dimensional surface and the second describes the period shocks. We describe each component in turn.

We suppose that the deaths and exposures are arranged in \( n_a \times n_y \) matrices \( Y \) and \( E \), that \( y = \text{vec}(Y) \) and \( e = \text{vec}E \) are their vector equivalents, and that the rows and columns of \( Y \) and \( E \) are classified respectively by ages \( x_a \) and years \( x_y \) each arranged in ascending order. The first component of our model uses two-dimensional \( P \)-splines (Eilers and Marx, 1996, Currie et al., 2004) to model the smooth underlying mortality surface. Let \( B_a = B(x_a), n_a \times c_a \), and \( B_y = B(x_y), n_y \times c_y \), be one-dimensional regression matrices of \( B \)-splines evaluated at age and year respectively; \( B_a \) and \( B_y \) are known as marginal regression matrices. The Kronecker product \( B_y \otimes B_a \) creates a two-dimensional regression basis. We suppose that the number of deaths \( y_{ij} \) at age \( i \) and year \( j \) follows a Poisson distribution with mean \( \mu_{ij} = e_{ij} \theta_{ij} \) where \( \theta_{ij} \) is the force of mortality. We define a generalized linear model (GLM) for \( y \) with regression matrix \( B_y \otimes B_a \), offset \( \log e \), log link and Poisson error. If we suppose that we have a rich basis of \( B \)-splines for age and year then a smooth surface is obtained by marginal penalization. We define the penalty matrix

\[
P = \lambda_a I_{c_y} \otimes D'_a D_a + \lambda_y D'_y D_y \otimes I_{c_a}
\]  

where \( D_a, \lambda_a, D_y \) and \( \lambda_y \) are the difference matrices and smoothing parameters for age and
Figure 1: Differences of the logarithms of observed and fitted mortalities for Swedish males for selected successive years.

We have defined a two-dimensional $P$-spline model; see Currie et al. (2004, 2006) and elsewhere for details of fitting these models.

We require the second component to have two properties. First, it must be versatile and capable of modelling different patterns in different years and second, the underlying patterns in different years must be smooth. Figure 1 illustrates both these features: the patterns in different years are distinct and follow separate underlying smooth curves. We define a second regression matrix by $I_{n_y} \otimes \bar{B}_a$ where $\bar{B}_a = \bar{B}_a(x_a)$, $n_a \times c$, is a marginal regression matrix of $B$-splines.

We suppose that $c$ is small so that the modelling by age for each year is quite crude. With $c$ regression coefficients for each of $n_y$ years $I_{n_y} \otimes \bar{B}_a$ is a large matrix, even with small $c$. The underlying smooth mortality surface is modelled by the first component of our model so we force smoothness on the second component by applying a ridge penalty to the age coefficients in each year. We define the penalty matrix

$$\bar{P} = \lambda_c I_{n_y} \otimes I_c = \lambda_c I_{n_y c}$$  \hspace{1cm} (1.2)
where $\lambda_s$ is the smoothing parameter for the shocks. Our two component model of mortality is thus a penalized generalized linear model with two additive components, linear predictor
\[
\log \mu = \log,e + B_y \otimes B_a a + I_{n_y} \otimes \tilde{B}_a \tilde{a}
\] (1.3) eq:lp2
and block diagonal penalty blockdiag$[P : \tilde{P}]$. This is a computationally demanding problem since three smoothing parameters must be chosen within the framework of a large GLM with $n_a n_y$ observations and $c_a c_y + cn_y$ regression variables. We describe how we deal with the computational problem in the next section.

2 Generalized linear array models

Generalized linear array models or GLAMs, introduced by Currie et al. (2006), provide a structure and a computational procedure for fitting GLMs whose model matrix can be written as a Kronecker product and whose data can be written as an array. The GLAM form of the 2-dimensional smooth model with regression matrix $B_y \otimes B_a$ is
\[
\log M = \log E + B_a AB_y'
\] (2.1) eq:GLAM.basic
where $M = E(Y)$ and $A, c_a \times c_y$, is the matrix of coefficients. In a large problem the GLAM approach gives very substantial savings in both storage and computational time over the usual GLM algorithm. The method can be extended to GLMs with additive components each of which has the GLAM form. The GLAM form of (1.3) is
\[
\log M = \log E + B_a AB_y' + \tilde{B}_a \tilde{A}
\] (2.2) eq:GLAM
where $\tilde{A}, c \times n_y$, is a further matrix of coefficients. We use the GLAM procedure to fit model (1.3). Efficient computation of the linear predictor in (1.3) is provided by (2.2). The fitting of a GLM also requires the computation of a weighted inner product; for the additive model (1.3) we require
\[
\begin{bmatrix}
(B_y \otimes B_a)' W_\delta (B_y \otimes B_a) & (B_y \otimes B_a)' W_\delta (I_{n_y} \otimes \tilde{B}_a) \\
(I_{n_y} \otimes \tilde{B}_a)' W_\delta (B_y \otimes B_a) & (I_{n_y} \otimes \tilde{B}_a)' W_\delta (I_{n_y} \otimes \tilde{B}_a)
\end{bmatrix}
\] (2.3) eq:XWX
where $W_\delta$ is the diagonal matrix of weights. Details of the efficient computation of this matrix are given in Currie et al. (2006), in particular the example in section 7.1. The GLM is now fitted with the usual scoring algorithm with the linear predictor, weighted inner product and working variable all computed using array computations.
A simpler method of modelling shocks is the mean shock, by which we mean a constant shock for all ages within a year. The GLAM form of the linear predictor for this model is

$$\log M = \log \mathbf{E} + B_\alpha \mathbf{A} \mathbf{B}_y' + \mathbf{1}_{n_\alpha} \mathbf{h}'$$

(2.4) \text{eq:lp3}

where $\mathbf{1}_{n_\alpha}$ is a vector of 1’s of length $n_\alpha$ and $\mathbf{h}' = (h_1, \ldots, h_{n_y})$. Mean shocks are used in some well-known models of mortality, such as the Lee-Carter and the Age-Period-Cohort models; neither of these models is capable of modelling the kind of effects we see in Figure 1 but we consider model (2.4) for comparison.

### 2.1 An application to Swedish data

We use the Baysian Information Criterion (BIC) for model selection and fit our three models to the Swedish mortality data used in section 1. We have $n_\alpha = 81$ ages and $n_y = 104$ years and used cubic $B$-splines with $c_\alpha = 19$, $c_y = 24$ and $c = 9$. The full regression matrix in (1.3) is $8424 \times 1392$, a large regression matrix. The fitted values have been added to Figure 1; it appears that our model has successfully modelled the systematic increases and decreases seen for our selected years. The extent of the period shocks is shown in Figure 2; the age structure of these shocks is evident.

Table 1 gives various summary statistics for all three models: the basic 2-dimensional smooth model (2.1), the mean shock model (2.4) and the age dependent shock model (2.2). It is clear that the period shock model is superior to the mean shock model which is, in turn, superior to the 2-dimensional model.

### 3 Concluding remarks

The 1918 observations are extreme. An alternative modelling strategy is to regard the 1918 data as missing. This has the effect of increasing the value of $\lambda_s$ (since the remaining data are much less variable by year). The estimates of the remaining period shocks are smoother compared to the shocks from the full data set.
Figure 2: Age dependent shocks to mortality surface by year

It is possible to express all our models in mixed model form. One consequence of this approach is that the period shocks can be regarded as random effects which is appealing given their nature. The GLAM methodology is available for mixed models (Currie et al., 2006).

The models in this paper and model (1.3) in particular are computationally intensive. Further computational savings over and above those provided by GLAM can be made by taking advantage of the form of the regression matrices of the period components in (1.3) and (2.4).

In conclusion, the period shock model was successful in modelling age-dependent departures within years for the Swedish data considered in this paper. Experience with other data sets has shown that the model is more widely applicable. We see two main uses of the model: first, it can detect age effects within single years and this will often be of interest in its own right; second, the two component model enables the underlying smooth surface to be more successfully identified.
References


Appendix

Estimation in a penalized GLM requires the repeated evaluation of

\[(B'\tilde{W}_\delta B + P)\hat{a} = B'\tilde{W}_\delta \hat{z}, \quad (3.1)\]

the penalized version of the usual scoring algorithm for a GLM. In our case, the regression matrix is from (1.3)

\[B = [B_y \otimes B_a : I_{n_y} \otimes \tilde{B}_a] \quad (3.2)\]

and the penalty matrix is

\[
\begin{bmatrix}
P & 0 \\ 0 & \tilde{P}
\end{bmatrix}
\quad (3.3)
\]

where \(P\) and \(\tilde{P}\) are defined in (1.1) and (1.2) respectively. The weight matrix is the diagonal matrix \(\tilde{W}_\delta = \text{diag}(\tilde{\mu})\). The most demanding component of (3.1) is the computation of the inverse of the weighted inner product matrix

\[
\begin{bmatrix}
(B_y \otimes B_a)'\tilde{W}_\delta (B_y \otimes B_a) + P & (B_y \otimes B_a)'\tilde{W}_\delta (I_{n_y} \otimes \tilde{B}_a) \\
(I_{n_y} \otimes \tilde{B}_a)'\tilde{W}_\delta (B_y \otimes B_a) & (I_{n_y} \otimes \tilde{B}_a)'\tilde{W}_\delta (I_{n_y} \otimes \tilde{B}_a) + \tilde{P}
\end{bmatrix}.
\quad (3.4)
\]

Currie et al. (2006) gave a method for fast, low-footprint evaluation of such matrices and we use their approach here. The ideas behind the method are (a) reduce the number of multiplications required by rearrangement of the calculations, and (b) rearrange the output from the fast computation into the required answer. The R and Splus languages enable the rearrangements to be performed easily and we describe the rearrangement process by giving some outline R-code. We hope that this code will accessible even to non-users of R; see also Currie et al. (2006). We consider (3.4) as a 2 \(\times\) 2 partition matrix and describe steps (a) and (b) for each block in turn. Step (a) requires a definition.

**Definition:** The row tensor of the matrices \(X_1, n \times c_1\) and \(X_2, n \times c_2\) is defined as

\[G(X_1, X_2) = (X_1 \otimes 1_{c_2}) \ast (1_{c_1} \otimes X_2), \quad n \times c_1c_2, \quad (3.5)\]

where \(1_{c_1}\) and \(1_{c_2}\) are vectors of 1s of lengths \(c_1\) and \(c_2\) respectively and \(\ast\) denotes element-by-element multiplication.

We define \(W\) to be the \(n_a \times n_y\) matrix of weights, i.e. \(\text{vec}(W) = \text{diag}(W_\delta)\). Then we have

(a) The upper left block is computed from

\[(B_y \otimes B_a)'\tilde{W}_\delta (B_y \otimes B_a), c_a c_y \times c_a c_y \equiv G(B_a, B_a)'WG(B_y, B_y), c_a^2 \times c_y^2. \quad (3.6)\]

\[\text{eq:block11}\]
Here ‘≡’ means that both sides have the same elements, although their dimensions, as indicated, are different. This achieves efficient computation of the entries in the upper left block. The elements of the right-hand side must now be rearranged to give the left-hand side and we now give the R-code to achieve this rearrangement. We suppose that the row tensors \( G(B_a, B_a) \) and \( G(B_y, B_y) \) are stored as \( GBaa \) and \( GByy \) respectively, and that the weight matrix is \( W \). The column dimensions of \( B_a \) and \( B_y \) are stored in \( c.a \) and \( c.y \) respectively. The R-function \( aperm \) permutes the dimensions of an array. We have

\[
\text{Block11} = \text{t}(\text{GBaa}) \% \% W \% \% \text{GByy} \\
\text{dim(Block11)} = (c.a, c.a, c.y, c.y) \\
\text{Block11} = \text{matrix}(\text{aperm(Block11, c(1, 3, 2, 4))), nrow = c.a * c.y)
\]

(b) The upper right block is

\[
(B_y \otimes B_a)'W(\delta(I_{ny} \otimes \bar{B}_a), c_a c_y \times cn_y \equiv G(B_a, \bar{B}_a)'WG(B_y, I_{ny}), c_a c \times c_y n_y. \tag{3.7} \]

This identity is a generalization of the well-known method for calculating the product \( \delta \cdot M \) where \( \Delta \) is a diagonal matrix and \( \delta \) is the vector which consists of the diagonal elements in \( \Delta \); \( M \) is any conformable matrix. Here we observe the convention that the elements in \( \delta \) are recycled through the columns of \( M \), as in R or Splus, for example. Thus, in (3.8) we have \( b_y \) has length \( n_y c_y \) which equals the number of rows in \( \text{stack}(W', c_y) \).

For the rearrangement R-code we suppose that the row tensor \( G(B_a, \bar{B}_a) \) is stored in \( GBaa2 \), \( \text{vec}(B_y) \) in \( b.y \), the column dimension of \( \bar{B}_a \) in \( c.c \) and row dimension of \( B_y \) in \( n.y \). Then

\[
\text{stackW} = \text{NULL} \\
\text{for}(j \text{ in } 1: c.y) \text{stackW} = \text{rbind}(\text{stackW}, \text{t}(W)) \\
\text{Block12} = \text{t}(\text{GBaa2}) \% \% \text{t}(b.y \* \text{stackW}) \\
\text{dim(Block12)} = (c.c, c.a, n.y, c.y) \\
\text{Block12} = \text{matrix}(\text{aperm(Block12, c(2, 4, 1, 3))), nrow = c.y * c.a)
\]
(c) Let \( W = [w_1, \ldots, w_{ny}] \) be the column representation of the weight matrix. First, we compute the lower right block directly as

\[
(I_{ny} \otimes \tilde{B}_a)' W_{\delta}(I_{ny} \otimes \tilde{B}_a) = \text{blockdiag}[\tilde{B}_a' W_{\delta_1} \tilde{B}_a : \ldots : \tilde{B}_a' W_{\delta_{ny}} \tilde{B}_a]
\]  

(3.9)  

where \( W_{\delta_j} = \text{diag}(w_j), j = 1, \ldots, ny \). Now each of these \( ny \) blocks can be computed from

\[
\tilde{B}_a' W_{\delta_j} \tilde{B}_a, c \times c \equiv G(\tilde{B}_a, \tilde{B}_a)' w_j, c^2 \times 1, j = 1, \ldots, ny
\]  

(3.10)  

so the elements of the \( i \)th block of the right-hand side of (3.9) are given by the \( i \)th column of \( G(\tilde{B}_a, \tilde{B}_a)' W, c^2 \times ny \). The \( i \)th block itself is obtained by redimensioning the \( i \)th column of \( G(\tilde{B}_a, \tilde{B}_a)' W \) into a \( c \times c \) matrix; no special code is required for this.

It remains to find the inverse of the matrix in (3.4). The inverse of a partition matrix can be computed by

\[
\begin{bmatrix} A & B \\ B' & C \end{bmatrix}^{-1} = \begin{bmatrix} X & -XBC^{-1} \\ -C^{-1}B'X & C^{-1} + C^{-1}B'XBC^{-1} \end{bmatrix}
\]  

(3.11)  

where \( X = (A - BC^{-1}B')^{-1} \) which can be efficiently evaluated since \( C \) is block-diagonal. The calculation of \( C^{-1} \) is best done at the same time as the calculation of the lower right block of (3.4) in (c) above.