Fast Two-Dimensional Smoothing with Discrete Cosine Transform

Pavel Lyubin(✉) and Eugeny Shchetinin(✉)

Department of Applied Mathematics,
Moscow State Technology University "STANKIN",
3a, Vadkovsky lane, Moscow, Russia 119136
lyubin.p@gmail.com, riviera-molto@mail.ru

Abstract. Smoothing is the process of removing "noise" and "insignificant" fragments while preserving the most important properties of the data structure. We propose a fast spline method for two-dimensional smoothing. Data smoothing usually attained by parametric and nonparametric regression. The nonparametric regression requires a prior knowledge of the regression equation form. However, most of the investigated data can't be parameterized simply. From this point of view, our algorithm belongs to nonparametric regression. Our simulation study shows that smoothing with discrete cosine transform is orders of magnitude faster to compute than other two-dimensional spline smoothers.

Keywords: Nonparametric regression · Two-dimensional estimation · Penalized splines · Smoothing splines · Cross-validation · Discrete cosine transform

1 Problem Statement

Raw data of real processes are noisy and need "smoothing" before analyse. Smoothing is attempt to filter "noise" or "insignificant" fragments while preserving the most important properties of data structure. Consider the following model

$$y = \hat{y} + \varepsilon \tag{1}$$

where ε - Gaussian white noise. There are supposed that function $\hat{y}$ should be smooth, i.e. has continuous derivatives up to some order. Data smoothing is usually carried out by a parametric or nonparametric regression. In the case of parametric regression, it requires some a priori knowledge of regression equation form, which must well described original proccess. However, most of the observed data is impossible to parameterize and function $f(x)$ can't be determined analytically. From this point of view, nonparametric and semiparametric regression is the best approach to solving the problem (1). One of the classical methods for smoothing data is the use of various modifications least squares with

V.M. Vishnevskiy et al. (Eds.): DCCN 2016, CCIS 678, pp. 646–656, 2016.
DOI: 10.1007/978-3-319-51917-3_55

penalty. It was first introduced in 1920 [1] and it has been extensively studied ever since 1990 [2]. This technique consists in minimize some functional that balances between "approximation" and "smoothness" of estimation and it has follow form

$$F(\hat{y}) = RSS + \lambda \cdot P(\hat{y}) = ||\hat{y} - y||^2 + \lambda \cdot P(\hat{y}), \tag{2}$$

where $||\cdot||$ - Euclidean norm. The parameter λ is a real positive number controlling the smoothness of solutions: smoothness of $\hat{y}$ growing when parameter increases. The regression is called smoothing spline [1,3,4], when the penalty function written like square integral of p-order derivatives of $\hat{y}$. Apart from this, simple and effective approach to solving problem (1) is squared form of penalty function [5]:

$$P(\hat{y}) = ||D\hat{y}||^2 \tag{3}$$

where D - tridiagonal matrix as

$$\begin{bmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{bmatrix}$$

This paper is a continuation of the authors research reviewed in papers [6,7]. Additionally, some main ideas gleaned from the articles [8,9] (Fig. 2).

2 One-Dimensional Smoothing

Suppose $\{x_i\}_{1 \le i \le n}$ is equally spaced points and response function follows

$$y_i = f(x_i) + \varepsilon_i \tag{4}$$

where $\varepsilon_i \sim N(0, \sigma^2)$. Let $\hat{y}$ is an estimate of $f(x_i)$. After minimization (2) we have

$$\hat{y} = H(\lambda) \cdot y, \tag{5}$$

where $H(\lambda) = (I + \lambda \cdot D^T D)^{-1}$ is a projection matrix and λ is smoothing parameter. Smoothing parameter selecting by minimization of following equation

$$GCV(\lambda) = \frac{RSS(\lambda)/n}{(1 - Tr(H(\lambda))/n)^2}. \tag{6}$$

This approach is called as method of cross-validation. Matrix D has some special properties if observations is equidistant. That's possible to simplify the calculation GCV, because matirx D can explain $U\Gamma U^T$, where matrix U is unitary and it is a discrete cosine transformation [9]. Then RSS can be rewritten as follows:

$$\begin{aligned} RSS &= ||\hat{y} - y||^2 = ||H(\lambda) \cdot y - y||^2 \\ &= ||((I + \lambda \cdot D^T D)^{-1} - I) \cdot y||^2 \\ &= ||(U \cdot (I + \lambda \cdot \Gamma^2)^{-1} - I) \cdot U^T \cdot y||^2 \\ &= \sum_i (\frac{1}{1 + \lambda \gamma_i^2} - 1)^2 \cdot DCT_i^2(y). \end{aligned}$$

In this case, (6) can be rewritten in more convenient for computing form

$$GCV(\lambda) = \frac{n \cdot \sum_i (\frac{1}{1+\lambda\gamma_i^2} - 1)^2 \cdot DCT_i^2(y)}{(n - \sum_i (\frac{1}{1+\lambda\gamma_i^2})^2}. \tag{7}$$

3 Two-Dimensional Smoothing

Suppose $\{(x_{1,i}, x_{2,j})\}_{1 \le i \le n_1, 1 \le j \le n_2}$ is uniform grid and response function follows

$$y_{i,j} = f(x_{1,i}, x_{2,j}) + \varepsilon_{i,j} \tag{8}$$

where $\varepsilon_{i,j} \sim N(0, \sigma^2)$. In this case, values of response function can represent like matrix Y, where element of i-th row and j-th column is value $y_{i,j}$. Then the smoothed values will be denoted by $\hat{Y}$. Introduce the operation vec, which represents matrix in column vector form. Then $vec(\hat{Y})$ can be written:

$$vec(\hat{Y}) = (H_{x_2} \otimes H_{x_1}) \cdot vec(Y) = H_{x_2,x_1} \cdot vec(Y), \tag{9}$$

where H_{x_1}, H_{x_2} - projection matrix for corresponding dimension. Obviously, the projection matrix has follow form

$$H_{x_i} = (I_{n_i} + \lambda_i D_{n_i}^T D_{n_i})^{-1}, \quad i = 1, 2. \tag{10}$$

Applying the approach and properties of the tensor product [10], expression (9) can be simplified as follows:

$$\begin{aligned} \hat{y} &= (H_{x_2} \otimes H_{x_1}) \cdot y \\ &= (I_{n_2} + \lambda_2 D_{n_2}^T D_{n_2})^{-1} \otimes (I_{n_1} + \lambda_1 D_{n_1}^T D_{n_1})^{-1} \cdot y \\ &= U_{x_2} \cdot (\frac{1}{1 + \lambda_2 \gamma_{x_2}^2}) \cdot U_{x_2}^T \otimes U_{x_1} \cdot (\frac{1}{1 + \lambda_1 \gamma_{x_1}^2}) \cdot U_{x_1}^T \cdot y \\ &= U_{x_2} \otimes U_{x_1} \cdot (\frac{1}{1 + \lambda_1 \gamma_{x_1}^2}) \otimes (\frac{1}{1 + \lambda_2 \gamma_{x_2}^2}) \cdot U_{x_2}^T \otimes U_{x_1}^T \cdot y \\ &= U_{x_2,x_1} \cdot \Gamma_{x_2,x_1} \cdot U_{x_2,x_1}^T \cdot y \end{aligned}$$

To automatically search for the best values λ_1 and λ_2, we use a cross-validation adapted for two-dimensional case:

$$GCV(\lambda_1, \lambda_2) = \frac{RSS/n}{(1 - Tr(H_{x_2,x_1})/n^2)}. \tag{11}$$

Properties of tensor product of matrices [10] denotes $Tr(H_{x_2,x_1}) = \sum \frac{1}{1+\lambda_1\gamma_{x_1}^2} \cdot \sum \frac{1}{1+\lambda_2\gamma_{x_2}^2}$. Obviously, main consuming place of the estiomation is a calculation RSS, because it requires evaluation of $\hat{y}$ for all combinations λ_1 and λ_2 . This calculation can be simplified:

$$\begin{aligned}
RSS &= ||\hat{y} - y||^2 = ||H_{x_2,x_1} \cdot y - y||^2 = ||(H_{x_2,x_1} - I_n) \cdot y||^2 \\
&= ||U_{x_2,x_1} \cdot (\Gamma_{x_2,x_1} - I_n) \cdot U_{x_2,x_1}^T \cdot y||^2 \\
&= (U_{x_2,x_1} \cdot (\Gamma_{x_2,x_1} - I_n) \cdot U_{x_2,x_1}^T \cdot y)^T \cdot (U_{x_2,x_1} \cdot (\Gamma_{x_2,x_1} - I_n) \cdot U_{x_2,x_1}^T \cdot y) \\
&= (DCT_2 \cdot y)^T \cdot (\Gamma_{x_2,x_1} - I_n)^2 \cdot DCT_2 \cdot y \\
&= \sum (\gamma_{x_2,x_1} - 1)^2 \cdot (DCT_2 \cdot y)^2,
\end{aligned}$$

where DCT_2 - is a two-dimensional discrete cosine transform. From the simplified equation shows the transformation must evaluate one times and result change with values γ_{x_2,x_1} depending values λ_1 and λ_2. This approach implemented in R. To demonstrate the advantages of considered approach performed numerical experiments: with model and real data (Fig. 4).

4 Experiments

Model data: To illustrate the effectiveness of the algorithm, sample data have been modeled from function $sin(2\pi(x_1 - 0.5)^3) \cdot cos(4\pi x_2)$ with noise - random values from normal distribution of $N(0, 0.2^2)$ (Fig. 1). Smoothing was carried by presented approach and MGCV package [11], which implements smoothing with penalized splines, including multidimensional case with tensor product of basic functions. Below is a table contains result of smoothing with different methods (Table 1).

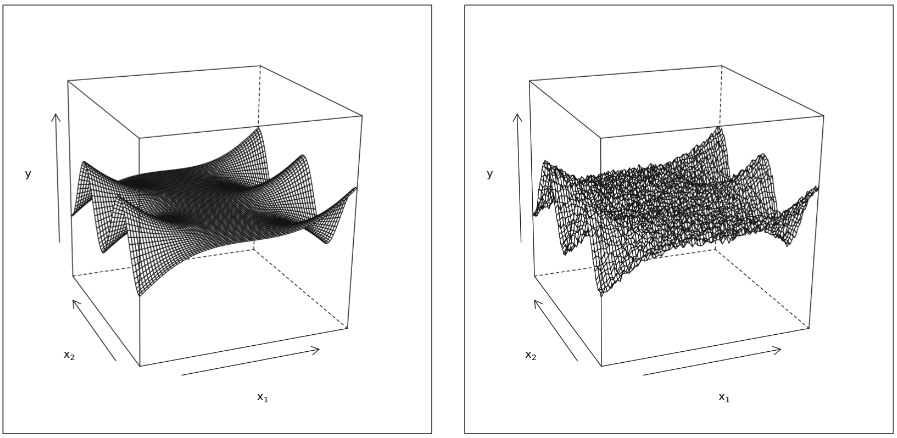

Fig. 1. Function $sin(2\pi(x - 0.5)^3) \cdot cos(4\pi y)$: raw (left) and with noise (right).

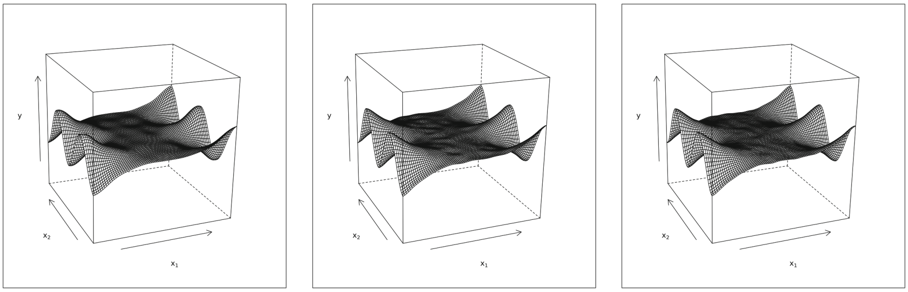

Fig. 2. Results of smoothing $sin(2\pi(x - 0.5)^3) \cdot cos(4\pi y)$: GAM with 10^2 knots (top-left), GAM with 20^2 knots (top-right) and DCT (bottom).

Table 1. The results of smoothing model data with different methods.

	P-splines with DCT	GAM with 10^2 knots	GAM with 20^2 knots
RSS	9.488243	11.72485	9.87163
MSE	0.001483	0.001832	0.00154
Corr. with true values	0.9993394	0.996919	0.9991624
Est. time (s)	1.941	10.237	29.875

Real data: To demonstrate the practical application of the approach, real data of mortality in Russia have been smoothed and compared with results of another approaches. The data are taken from the open source [12] and contains observations for ages of 0 and 110 between years of 1959 and 2010. For experiment was taken part of data, which belongs to the older ages (50–101, Fig. 3). That part was chosen, because observations contain many errors and outliers. Thus, analyzed data are evenly spaced values of mortality rates on grid with size 52×52. Smoothing conducted outlined approach, package MGCV and parametric model of the Lee-Carter, who has become a classic for appraisals dimensional mortality surface. The next table contains result of estimations (Table 2).

Table 2. The results of smoothing a two-dimensional surface of Russian mortality rates.

	P-splines with DCT	Lee-Carter model	GAM with 12^2 knots
RSS	0.21637	18.5092	0.41395
MSE	0.0000905	0.0077379	0.0001731
Est. time (s)	0.49	1.194	4.185

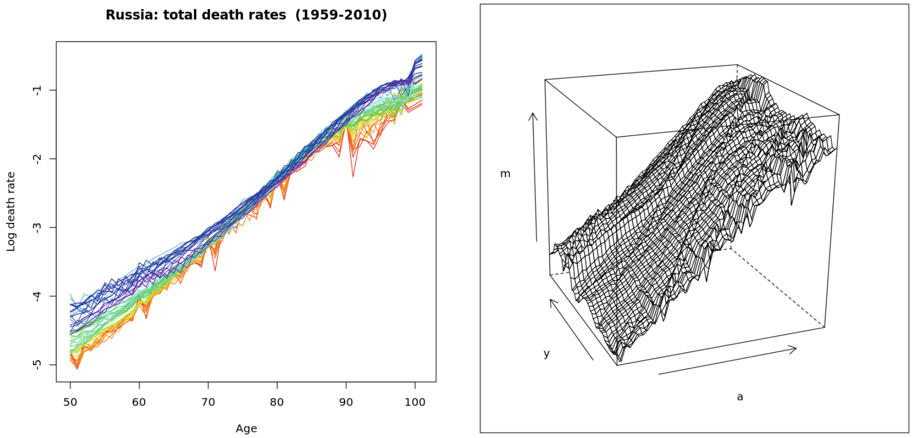

Fig. 3. The raw mortality rates in Russia for ages of 50 and 101 between years of 1959 and 2010.

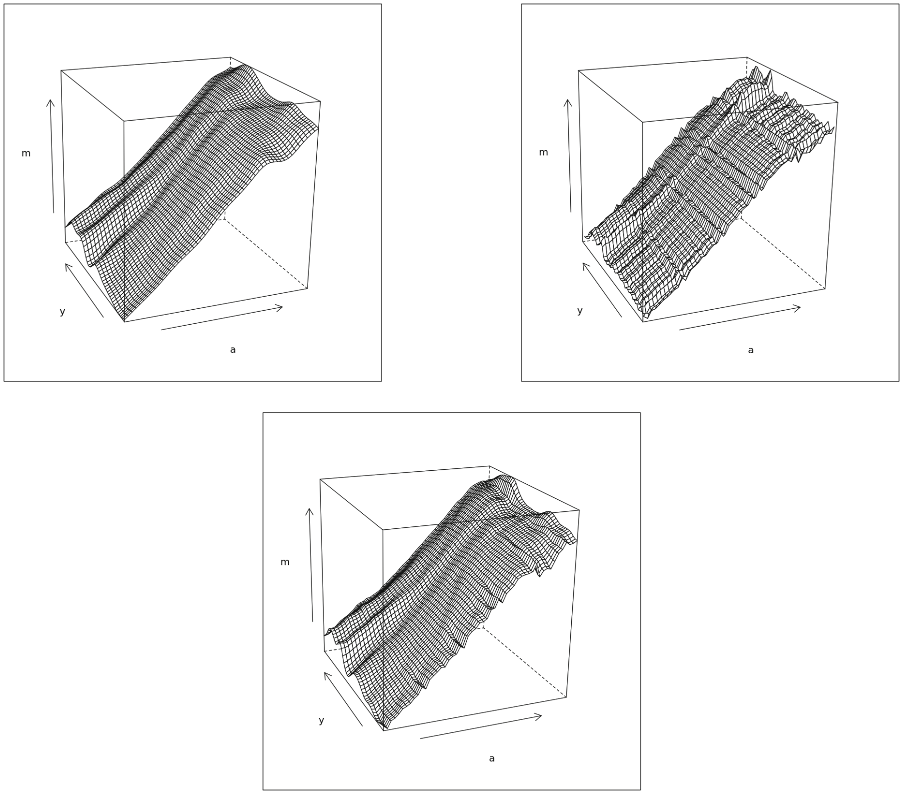

Fig. 4. Results of smoothing mortality data: GAM with 12^2 knots (top-left), Lee-Carter (top-right) and DCT (bottom).

5 Conclusion

Obviously, described approach is very effective, because it's fast and no need much memory. Note, if the sample size increases then calculation speed increases slightly with same estimation quality. Results:

1. equations obtained for two-dimensional case with two smoothing parameters;
2. the approach implemented in R for one- and two-dimensional cases;
3. the approach compared with similar approache and model.

In next studies is expected to consider the following possibilities:

- extension of the approach to the multidimensional case with many smoothing parameters;
- use other common criteria for smoothing parameters selection, for example, BIC or AIC;
- use a faster method for minimization GCV instead of grid search.

Appendix: R code

Program commands for model data

```
#Clear workspace
rm(list=ls(all=T))

library(mgcv)
library(lattice)

f1 <- function(x,y) { sin(2 * pi * (x - .5)^3) * cos (4 * pi * y) }
n <- 80
xn <- seq(0,1, length.out = n)
yn <- seq(0,1, length.out = n)
xy <- expand.grid(x = xn, y = yn)]
Ytrue <- f1(xy[,1], xy[,2])
zn <- Ytrue + rnorm(dim(xy)[1], mean = 0, sd = .2^2)

## TPRS
st <- proc.time()
b0 <- gam(zn~s(xy[,1],xy[,2], bs='ts', k=20^2))
en <- proc.time()
ti <- en['elapsed'] - st['elapsed']
cat("Gam time passed:", ti, "\n")

wireframe(
  matrix(fitted(b0), nrow=n, ncol=n),
  zlim = c ( -2, 2),
  xlab = expression(x[1]),
```

```
  ylab = expression(x[2]),
  zlab = expression(y),
  screen = list(z = 20, x = -70, y = 3)
  )

#DCT
lr <- seq( 64, 66, by = .1 )
lc <- seq( 36, 38, by = .1 )
fit <- psdct2d(matrix(zn, nrow=n, ncol=n))
plot(fit, theta = -15, phi = 30, zlim = c ( -2, 2))
summary(fit)

wireframe(
  matrix(fitted(fit), nrow=n, ncol=n),
  zlim = c ( -2, 2),
  xlab = expression(x[1]),
  ylab = expression(x[2]),
  zlab = expression(y),
  screen = list(z = 20, x = -70, y = 3)
  #screen = list(z = -60, x = -60)
  )

cat("RSS DCT:", sum( (residuals(fit))^2 ), "\n")
cat("RSS GAM:", sum( (residuals(b0))^2 ), "\n")
cat("Corr DCT:", cor(fitted(fit), Ytrue), "\n")
cat("Corr GAM:", cor(fitted(b0), Ytrue))
```

Program commands for model data

```
#Clear workspace
rm(list=ls(all=T))

library(demography)

#Raw
ru.mort <- read.demogdata("data/Mx_1x1.txt",
"data/Exposures_1x1.txt", "mortality", "Russia")
plot(ru.mort, series="total")

ru.ext <- extract.ages(ru.mort, 50:101 , FALSE)
plot(ru.ext, series="total")

wireframe(
  matrix(log(ru.ext$rate$total), nrow=52, ncol=52),
  xlab = expression(a),
  ylab = expression(y),
```

```
  zlab = expression(m),
  screen = list(z = 20, x = -70, y = 3)
  )

#GAM
library(mgcv)
gamst <- proc.time()
z <- as.vector(log(ru.ext$rate$total))
x <- 1:nrow(ru.ext$rate$total)
y <- 1:ncol(ru.ext$rate$total)
xy <- expand.grid(x, y)
ru.gam <- gam(z~s(xy[,1],xy[,2], bs='ts', k=12^2))
gamen <- proc.time()
gamel <- gamen['elapsed'] - gamst['elapsed']
cat("Gam time passed:", gamel, "\n")
persp(matrix(fitted(ru.gam), nrow=length(x), ncol=length(y)))
persp(matrix(residuals(ru.gam), nrow=length(x), ncol=length(y)))
levelplot(matrix(residuals(ru.gam), nrow=length(x), ncol=length(y)))

wireframe(
  matrix(fitted(ru.gam), nrow=52, ncol=52),
  xlab = expression(a),
  ylab = expression(y),
  zlab = expression(m),
  screen = list(z = 20, x = -70, y = 3)
  )

#Lee-Carter
lcst <- proc.time()
ru.lc <- lca(ru.ext, adjust="e0")
plot(ru.lc)
persp(ru.lc$fitted$y)
persp(ru.lc$residuals$y)
levelplot(ru.lc$residuals$y)
lcen <- proc.time()
lcel <- lcen['elapsed'] - lcst['elapsed']
cat("LC time passed:", lcel, "\n")

wireframe(
  ru.lc$fitted$y,
  xlab = expression(a),
  ylab = expression(y),
  zlab = expression(m),
  screen = list(z = 20, x = -70, y = 3)
  )
```

```
#DCT
ru.dct <- psdct2d(log(ru.ext$rate$total))
persp(matrix(residuals(ru.dct2), nrow=length(x), ncol=length(y)))
levelplot(matrix(residuals(ru.dct2), nrow=length(x), ncol=length(y)))

wireframe(
  matrix(fitted(ru.dct), nrow=52, ncol=52),
  xlab = expression(a),
  ylab = expression(y),
  zlab = expression(m),
  screen = list(z = 20, x = -70, y = 3)
  )
```

References

1. Whittaker, E.T.: On a new method of graduation. Proc. Edinb. Math. Soc. **41**, 62–75 (1923)
2. Wahba, G.: Spline Models for Observational Data. Society for Industrial Mathematics, Philadelphia (1990)
3. Schoenberg, I.J.: Spline functions and the problem of graduation. Proc. Natl. Acad. Sci. USA **52**, 947–950 (1964)
4. Takezawa, K.: Introduction to Nonparametric Regression. Wiley, New York (2005)
5. Weinert, H.L.: Efficient computation for whittaker-henderson smoothing. Comput. Stat. Data Anal. **52**, 959–974 (2007)
6. Yu, S.E., Lyubin, P.G.: Robust smoothing with splines. Sci. Rev. **1**, 86–94 (2015)
7. Lyubin, P.G., Shetinin, E.Y.: Stochastic models of mortality estimation. Sci. Rev. **18**, 147–155 (2015)
8. Xiao, L., Li, Y., Ruppert, D.: Fast bivariate p-splines: the sandwich smoother. J. Roy. Stat. Soc. **75**, 577–599 (2013)
9. Garcia, D.: Robust smoothing of gridded data in one and higher dimensions with missing values. Comput. Stat. Data Anal. **54**, 1167–1178 (2010)
10. Seber, G.: A Matrix Handbook for Statisticians. Wiley-Interscience, Hoboken (2007)
11. Wood, S.: mgcv: mixed gam computation vehicle with gcv/aic/reml smoothness estimation. R package version 1.8.10
12. D̃epartment of Demography at the University of California. The human mortality database. Last visited on 25.02.2016
13. Wood, S.N.: Thin plate regression splines. R. Stat. Soc. B **65**, 95–114 (2003)
14. Ruppert, D.: Selecting the number of knots for penalized splines. Comput. Graph. Stat. **1**, 735–757 (2006)
15. Li, Y., Ruppert, D.: On the asymptotics of penalized splines. Biometrika **95**, 415–436 (2008)
16. Dierckx, P.: A fast algorithm for smoothing data on a rectangular grid while using spline functions. SIAM J. Numer. Anal. **19**(6), 1286–1304 (1982)
17. Wood, S.N., Smith, L., Hyndman, R.J.: Spline interpolation for demographic variables: the monotonicity problem. J. Popul. Res. **21**(1), 95–98 (2004)

18. Eilers, P.H.C., Marx, B.D.: Generalized linear additive smooth structures. J. Comput. Graph. Stat. **11**(4), 758–783 (2002)
19. Eilers, P.H.C., Marx, B.D.: Multidimensional penalized signal regression. Technometrics **47**(1), 13–22 (2005)
20. Hyndman, R.J., Booth, H., Tickle, L., Maindonald, J.: Demography: forecasting mortality, fertility, migration and population data, R package version 1.18 (2014)
21. Yueh, W.C.: Eigenvalues of several tridiagonal matrices. Appl. Math. E-Not. **5**, 66–74 (2005)
22. Hoaglin, D.C., Welsch, R.E.: The hat matrix in regression and ANOVA. Am. Stat. **32**, 17–22 (1978)
23. Eilers, P.H.C., Marx, B.D.: Splines, knots, and penalties. Wiley Interdisc. Rev. Comput. Stat. **2**, 637–653 (2010)