Dynamic Retrospective Functional Regression
Technical Supplement

Daniel Gervini
University of Wisconsin–Milwaukee

December 5, 2013

Contents

1 Ordinary retrospective LS regression **1**

2 Dynamic retrospective functional regression **2**
2.1 Objective function and constraints 2
2.2 Derivatives 4
2.3 Warping function derivatives 7
2.3.1 Hermite splines 7
2.3.2 Smooth monotone warping functions 9

3 Hermite splines **9**
3.1 Interpolation on a single interval 9
3.1.1 Interpolation on the interval $[0, 1]$ 9
3.1.2 Interpolation on a general interval $[x_k, x_{k+1}]$ 10
3.2 Interpolating a data set 10
3.3 Monotone interpolation 13

1 Ordinary retrospective LS regression

Let $\boldsymbol{\phi}(s) \in \mathbb{R}^p$ be a basis of the x-space and $\boldsymbol{\psi}(t) \in \mathbb{R}^q$ a basis of the y-space. The regression slope has the form

$$\beta(s,t) = \boldsymbol{\phi}(s)^T \mathbf{B} \boldsymbol{\psi}(t)$$

with $\mathbf{B} \in \mathbb{R}^{p\times q}$. (We use the pc's of the x's and the y's as bases but the derivation that follows is valid for any other system.) We seek to minimize

$$SS(\mathbf{B}) = \sum_{i=1}^{n} \int_a^b \left\{ y_i^*(t) - \int_a^t \beta(s,t) x_i^*(s) ds \right\}^2 dt$$

where $x_i^* = x_i - \bar{x}$ and $y_i^* = y_i - \bar{y}$. We have

$$\int_a^t \beta(s,t) x_i^*(s) ds = \left\{ \int_a^t \boldsymbol{\phi}(s)^T x_i^*(s) ds \right\} \mathbf{B} \boldsymbol{\psi}(t),$$

so let

$$\boldsymbol{\gamma}_i(t) = \int_a^t \boldsymbol{\phi}(s) x_i^*(s) ds.$$

Then

$$\begin{aligned} SS(\mathbf{B}) &= \sum_{i=1}^{n} \int_a^b \left\{ y_i^*(t) - \boldsymbol{\gamma}_i(t)^T \mathbf{B} \boldsymbol{\psi}(t) \right\}^2 dt \\ &= \sum_{i=1}^{n} \int_a^b \left[y_i^*(t) - \left\{ \boldsymbol{\psi}(t)^T \otimes \boldsymbol{\gamma}_i(t)^T \right\} \mathrm{vec}(\mathbf{B}) \right]^2 dt \end{aligned}$$

and we can expand

$$\begin{aligned} SS(\mathbf{B}) \propto\ & \mathrm{vec}(\mathbf{B})^T \left[\sum_{i=1}^{n} \int_a^b \left\{ \boldsymbol{\psi}(t) \otimes \boldsymbol{\gamma}_i(t) \right\} \left\{ \boldsymbol{\psi}(t)^T \otimes \boldsymbol{\gamma}_i(t)^T \right\} dt \right] \mathrm{vec}(\mathbf{B}) \\ & -2 \left[\sum_{i=1}^{n} \int_a^b y_i^*(t) \left\{ \boldsymbol{\psi}(t)^T \otimes \boldsymbol{\gamma}_i(t)^T \right\} dt \right] \mathrm{vec}(\mathbf{B}). \end{aligned}$$

The LS estimator is then

$$\begin{aligned} \widehat{\mathrm{vec}(\mathbf{B})} &= \left[\sum_{i=1}^{n} \int_a^b \left\{ \boldsymbol{\psi}(t)\boldsymbol{\psi}(t)^T \otimes \boldsymbol{\gamma}_i(t)\boldsymbol{\gamma}_i(t)^T \right\} dt \right]^{-1} \left[\sum_{i=1}^{n} \int_a^b y_i^*(t) \left\{ \boldsymbol{\psi}(t) \otimes \boldsymbol{\gamma}_i(t) \right\} dt \right] \\ &= \left[\int_a^b \left\{ \boldsymbol{\psi}(t)\boldsymbol{\psi}(t)^T \otimes \mathbf{M}(t) \right\} dt \right]^{-1} \left[\int_a^b \left\{ \boldsymbol{\psi}(t) \otimes \mathbf{u}(t) \right\} dt \right] \end{aligned}$$

with $\mathbf{M}(t) = \sum_{i=1}^{n} \boldsymbol{\gamma}_i(t)\boldsymbol{\gamma}_i(t)^T$ and $\mathbf{u}(t) = \sum_{i=1}^{n} y_i^*(t)\boldsymbol{\gamma}_i(t)$.

2 Dynamic retrospective functional regression

2.1 Objective function and constraints

We minimize

$$F(\boldsymbol{\eta}) = \frac{1}{2} \sum_{i=1}^{n} \int r_i^2(t) dt + \lambda_w \frac{1}{2} \sum_{i=1}^{n} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_0\|^2$$

where

$$r_i(t) = y_i \circ w_i(t) - \mu_y(t) - \int_a^t \beta(s,t) \{x_i \circ w_i(s) - \mu_x(s)\} ds$$

with $\mu_x = \overline{x \circ w}$, $\mu_y = \overline{y \circ w}$, and

$$\beta(s,t) = \boldsymbol{\phi}(s)^T \mathbf{B} \boldsymbol{\psi}(t).$$

Here $\mathbf{B} \in \mathbb{R}^{p\times q}$, $\boldsymbol{\phi}(s) \in \mathbb{R}^p$ is a basis of the x-space and $\boldsymbol{\psi}(t) \in \mathbb{R}^q$ is a basis of the y-space. Concretely, we take $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ as eigenfunctions of the covariance operators of the warped x's and y's, respectively:

$$\begin{aligned} \int \rho_x(s,t)\phi_k(s)ds &= \lambda_k\phi_k(t), \quad k = 1,\ldots,p, \\ \int \rho_y(s,t)\psi_k(s)ds &= \xi_k\psi_k(t), \quad k = 1,\ldots,q, \end{aligned}$$

where

$$\begin{aligned} \rho_x(s,t) &= \frac{1}{n}\sum_{i=1}^{n}\{x_i \circ w_i(s) - \mu_x(s)\}\{x_i \circ w_i(t) - \mu_x(t)\}, \\ \rho_y(s,t) &= \frac{1}{n}\sum_{i=1}^{n}\{y_i \circ w_i(s) - \mu_y(s)\}\{y_i \circ w_i(t) - \mu_y(t)\}. \end{aligned}$$

Given a B-spline basis $\boldsymbol{\gamma}(s) \in \mathbb{R}^\nu$ we can express

$$\begin{aligned} \boldsymbol{\phi}(s) &= \mathbf{C}^T\boldsymbol{\gamma}(s), \\ \boldsymbol{\psi}(s) &= \mathbf{D}^T\boldsymbol{\gamma}(s), \end{aligned}$$

with $\mathbf{C} \in \mathbb{R}^{\nu\times p}$ and $\mathbf{D} \in \mathbb{R}^{\nu\times q}$. Then

$$\begin{aligned} \int \rho_x(s,t)\mathbf{C}^T\boldsymbol{\gamma}(s)ds &= \boldsymbol{\Lambda}\mathbf{C}^T\boldsymbol{\gamma}(t), \\ \int \rho_y(s,t)\mathbf{D}^T\boldsymbol{\gamma}(s)ds &= \boldsymbol{\Xi}\mathbf{D}^T\boldsymbol{\gamma}(t), \end{aligned}$$

with $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1,\ldots,\lambda_p)$ and $\boldsymbol{\Xi} = \mathrm{diag}(\xi_1,\ldots,\xi_q)$, so post-multiplying by $\boldsymbol{\gamma}(t)^T$ and integrating t out we get

$$\begin{aligned} \boldsymbol{\Omega}_x\mathbf{C} &= \mathbf{J}_0\mathbf{C}\boldsymbol{\Lambda}, \\ \boldsymbol{\Omega}_y\mathbf{D} &= \mathbf{J}_0\mathbf{D}\boldsymbol{\Xi}, \end{aligned}$$

with $\mathbf{J}_0 = \int \boldsymbol{\gamma}(t)\boldsymbol{\gamma}(t)^T dt$, $\boldsymbol{\Omega}_x = \iint \rho_x(s,t)\boldsymbol{\gamma}(s)\boldsymbol{\gamma}(t)^T\, ds\, dt$ and $\boldsymbol{\Omega}_y = \iint \rho_y(s,t)\boldsymbol{\gamma}(s)\boldsymbol{\gamma}(t)^T\, ds\, dt$. In addition, we have the "smoothed orthogonality" conditions

$$\begin{aligned} \mathbf{C}^T\mathbf{J}_x\mathbf{C} &= \mathbf{I}_p, \\ \mathbf{D}^T\mathbf{J}_y\mathbf{D} &= \mathbf{I}_q, \end{aligned}$$

where $\mathbf{J}_x = \mathbf{J}_0 + \lambda_x \mathbf{J}_2$, $\mathbf{J}_y = \mathbf{J}_0 + \lambda_y \mathbf{J}_2$, and $\mathbf{J}_2 = \int \boldsymbol{\gamma}''(t)\boldsymbol{\gamma}''(t)^T dt$. We also have the (linear) identifiability constraint

$$\bar{\boldsymbol{\theta}} = \boldsymbol{\theta}_0$$

and the non-negativity constraints

$$\begin{aligned} \lambda_k &\geq 0, \quad k = 1, \ldots, p, \\ \xi_k &\geq 0, \quad k = 1, \ldots, q. \end{aligned}$$

Typically we want the λ's and $\boldsymbol{\xi}$'s in decreasing order but to simplify the algorithm we do not impose this as explicit constraints, we just reorder them (if necessary) after estimation.

"The" parameter of the objective function F is then

$$\boldsymbol{\eta} = \begin{bmatrix} \mathrm{vec}(\mathbf{B}) \\ \mathrm{vec}(\mathbf{C}) \\ \mathrm{vec}(\mathbf{D}) \\ \boldsymbol{\lambda} \\ \boldsymbol{\xi} \\ \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{\theta}_n \end{bmatrix}.$$

The vector-valued nonlinear-constraint function will be

$$\mathbf{G}(\boldsymbol{\eta}) = \begin{bmatrix} \mathrm{vec}(\mathbf{C}^T \mathbf{J}_x \mathbf{C} - \mathbf{I}_p) \\ \mathrm{vec}(\mathbf{D}^T \mathbf{J}_y \mathbf{D} - \mathbf{I}_q) \\ \mathrm{vec}(\boldsymbol{\Omega}_x \mathbf{C} - \mathbf{J}_0 \mathbf{C} \boldsymbol{\Lambda}) \\ \mathrm{vec}(\boldsymbol{\Omega}_y \mathbf{D} - \mathbf{J}_0 \mathbf{D} \boldsymbol{\Xi}) \end{bmatrix}.$$

2.2 Derivatives

The gradient of the objective function is

$$\nabla F = \sum_{i=1}^{n} \int r_i(t) \{\nabla r_i(t)\} dt + \lambda_w \nabla P$$

with

$$\nabla P = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{\theta}_1 - \boldsymbol{\theta}_0 \\ \vdots \\ \boldsymbol{\theta}_n - \boldsymbol{\theta}_0 \end{bmatrix}.$$

To compute $\nabla r_i(t)$ we break $\boldsymbol{\eta}$ down into its constituent sub-parameters. Let $\tilde{x}_i^*(s) = x_i \circ w_i(s) - \mu_x(s)$. Since

$$\begin{aligned} \int_a^t \beta(s,t)\tilde{x}_i^*(s)ds &= \int_a^t \boldsymbol{\phi}(s)^T \mathbf{B} \boldsymbol{\psi}(t) \tilde{x}_i^*(s)ds \\ &= \left\{ \boldsymbol{\psi}(t)^T \otimes \int_a^t \boldsymbol{\phi}(s)^T \tilde{x}_i^*(s)ds \right\} \mathrm{vec}(\mathbf{B}) \end{aligned}$$

we have

$$\nabla_{\mathrm{vec}(\mathbf{B})} r_i(t) = -\boldsymbol{\psi}(t) \otimes \int_a^t \boldsymbol{\phi}(s)\tilde{x}_i^*(s)ds.$$

Similarly,

$$\begin{aligned} \int_a^t \boldsymbol{\phi}(s)^T \mathbf{B} \boldsymbol{\psi}(t) \tilde{x}_i^*(s)ds &= \int_a^t \boldsymbol{\gamma}(s)^T \mathbf{C} \mathbf{B} \boldsymbol{\psi}(t) \tilde{x}_i^*(s)ds \\ &= \left\{ (\mathbf{B}\boldsymbol{\psi}(t))^T \otimes \int_a^t \boldsymbol{\gamma}^T(s) \tilde{x}_i^*(s)ds \right\} \mathrm{vec}(\mathbf{C}) \end{aligned}$$

so

$$\nabla_{\mathrm{vec}(\mathbf{C})} r_i(t) = -\mathbf{B}\boldsymbol{\psi}(t) \otimes \int_a^t \boldsymbol{\gamma}(s)\tilde{x}_i^*(s)ds,$$

and

$$\begin{aligned} \int_a^t \boldsymbol{\phi}(s)^T \mathbf{B} \boldsymbol{\psi}(t) \tilde{x}_i^*(s)ds &= \boldsymbol{\gamma}(t)^T \mathbf{D} \mathbf{B}^T \int_a^t \boldsymbol{\phi}(s) \tilde{x}_i^*(s)ds \\ &= \left\{ \left(\mathbf{B}^T \int_a^t \boldsymbol{\phi}(s)\tilde{x}_i^*(s)ds \right)^T \otimes \boldsymbol{\gamma}(t)^T \right\} \mathrm{vec}(\mathbf{D}) \end{aligned}$$

so

$$\nabla_{\mathrm{vec}(\mathbf{D})} r_i(t) = -\mathbf{B}^T \int_a^t \boldsymbol{\phi}(s)\tilde{x}_i^*(s)ds \otimes \boldsymbol{\gamma}(t).$$

For the gradients with respect to $\boldsymbol{\theta}_i$, we have

$$\nabla_{\boldsymbol{\theta}_j} r_i(t) = \mathbf{0}, \;\; i \neq j,$$

and

$$\nabla_{\boldsymbol{\theta}_i} r_i(t) = y_i'(w_i(t))\{\nabla_{\boldsymbol{\theta}_i} w_i(t)\} - \int_a^t \beta(s,t) x_i'(w_i(s))\{\nabla_{\boldsymbol{\theta}_i} w_i(s)\} ds.$$

The specific form of $\nabla_{\boldsymbol{\theta}_i} w_i(t)$ will depend on the specific family used for the warping functions; see next section.

To find the derivatives of the constraint function $\mathbf{G}$ we use differentials, breaking down both $\mathbf{G}$ and $\boldsymbol{\eta}$ into its constituents. Concretely, the Jacobian of $\mathbf{G}(\boldsymbol{\eta})$ is going to have the block structure

$$\mathrm{J}\mathbf{G} = \begin{bmatrix} \mathbf{0} & \mathrm{J}_{\mathrm{vec}(\mathbf{C})}\mathbf{G}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \cdots & \mathbf{0} \\ \vdots & \mathbf{0} & \mathrm{J}_{\mathrm{vec}(\mathbf{D})}\mathbf{G}_2 & \mathbf{0} & \mathbf{0} & \vdots & \vdots \\ \vdots & \mathrm{J}_{\mathrm{vec}(\mathbf{C})}\mathbf{G}_3 & \mathbf{0} & \mathrm{J}_{\boldsymbol{\lambda}}\mathbf{G}_3 & \mathbf{0} & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathrm{J}_{\mathrm{vec}(\mathbf{D})}\mathbf{G}_4 & \mathbf{0} & \mathrm{J}_{\boldsymbol{\xi}}\mathbf{G}_4 & \mathbf{0} \cdots & \mathbf{0} \end{bmatrix}$$

For $\mathbf{G}_1 = \mathrm{vec}(\mathbf{C}^T\mathbf{J}_x\mathbf{C} - \mathbf{I}_p)$ we have

$$\begin{aligned} \mathrm{d}\mathbf{G}_1 &= \mathrm{vec}(\mathrm{d}\mathbf{C}^T\mathbf{J}_x\mathbf{C} + \mathbf{C}^T\mathbf{J}_x\mathrm{d}\mathbf{C}) \\ &= (\mathbf{C}^T\mathbf{J}_x \otimes \mathbf{I}_p)\mathrm{vec}(\mathrm{d}\mathbf{C}^T) + (\mathbf{I}_p \otimes \mathbf{C}^T\mathbf{J}_x)\mathrm{vec}(\mathrm{d}\mathbf{C}) \\ &= \{(\mathbf{C}^T\mathbf{J}_x \otimes \mathbf{I}_p)\mathbf{K}_{\nu,p} + (\mathbf{I}_p \otimes \mathbf{C}^T\mathbf{J}_x)\}\mathrm{vec}(\mathrm{d}\mathbf{C}) \end{aligned}$$

where $\mathbf{K}_{\nu,p}$ is the commutation matrix, so

$$\mathrm{J}_{\mathrm{vec}(\mathbf{C})}\mathbf{G}_1 = (\mathbf{C}^T\mathbf{J}_x \otimes \mathbf{I}_p)\mathbf{K}_{\nu,p} + (\mathbf{I}_p \otimes \mathbf{C}^T\mathbf{J}_x).$$

In an analogous way we obtain

$$\mathrm{J}_{\mathrm{vec}(\mathbf{D})}\mathbf{G}_2 = (\mathbf{D}^T\mathbf{J}_y \otimes \mathbf{I}_q)\mathbf{K}_{\nu,q} + (\mathbf{I}_q \otimes \mathbf{D}^T\mathbf{J}_y).$$

For $\mathbf{G}_3 = \mathrm{vec}(\boldsymbol{\Omega}_x\mathbf{C} - \mathbf{J}_0\mathbf{C}\boldsymbol{\Lambda})$, first note that if $\mathbf{e}_{pj}$ represents the jth canonical vector in $\mathbb{R}^p$ then

$$\boldsymbol{\Lambda} = \sum_{j=1}^p \lambda_j \mathbf{e}_{pj}\mathbf{e}_{pj}^T = \sum_{j=1}^p (\mathbf{e}_{pj}^T\boldsymbol{\lambda})\mathbf{e}_{pj}\mathbf{e}_{pj}^T = \sum_{j=1}^p (\mathbf{e}_{pj}\mathbf{e}_{pj}^T)\boldsymbol{\lambda}\mathbf{e}_{pj}^T$$

so

$$\mathrm{vec}(\boldsymbol{\Lambda}) = \mathbf{M}_p\boldsymbol{\lambda}$$

with $\mathbf{M}_p = \sum_{j=1}^{p}(\mathbf{e}_{pj} \otimes \mathbf{e}_{pj}\mathbf{e}_{pj}^T)$; similarly,

$$\mathrm{vec}(\boldsymbol{\Xi}) = \mathbf{M}_q \boldsymbol{\xi}.$$

Then

$$\begin{aligned} \mathrm{d}_C \mathbf{G}_3 &= \mathrm{vec}(\boldsymbol{\Omega}_x \mathrm{d}\mathbf{C} - \mathbf{J}_0 \mathrm{d}\mathbf{C}\boldsymbol{\Lambda}) \\ &= (\mathbf{I}_p \otimes \boldsymbol{\Omega}_x)\mathrm{vec}(\mathrm{d}\mathbf{C}) - (\boldsymbol{\Lambda} \otimes \mathbf{J}_0)\mathrm{vec}(\mathrm{d}\mathbf{C}), \end{aligned}$$

so

$$\mathrm{J}_{\mathrm{vec}(\mathbf{C})} \mathbf{G}_3 = (\mathbf{I}_p \otimes \boldsymbol{\Omega}_x) - (\boldsymbol{\Lambda} \otimes \mathbf{J}_0);$$

and

$$\mathbf{G}_3 = \mathrm{vec}(\boldsymbol{\Omega}_x \mathbf{C}) - (\mathbf{I}_p \otimes \mathbf{J}_0 \mathbf{C})\mathbf{M}_p \boldsymbol{\lambda},$$

so

$$\mathrm{J}_{\boldsymbol{\lambda}} \mathbf{G}_3 = -(\mathbf{I}_p \otimes \mathbf{J}_0 \mathbf{C})\mathbf{M}_p.$$

In an analogous way we get

$$\mathrm{J}_{\mathrm{vec}(\mathbf{D})} \mathbf{G}_4 = (\mathbf{I}_q \otimes \boldsymbol{\Omega}_y) - (\boldsymbol{\Xi} \otimes \mathbf{J}_0)$$

and

$$\mathrm{J}_{\boldsymbol{\xi}} \mathbf{G}_4 = -(\mathbf{I}_q \otimes \mathbf{J}_0 \mathbf{D})\mathbf{M}_q.$$

2.3 Warping function derivatives

2.3.1 Hermite splines

When $w_i(t)$ belongs to the family of monotone Hermite splines, it can be expressed as

$$\begin{aligned} w_i(t) &= \sum_{j=0}^{r+1} \tau_{ij} \alpha_j(t; \boldsymbol{\tau}_0) + \sum_{j=0}^{r+1} d_{ij} \beta_j(t; \boldsymbol{\tau}_0) \\ &= [a, \boldsymbol{\tau}_i^T, b]\boldsymbol{\alpha}(t) + \mathbf{d}_i^T \boldsymbol{\beta}(t) \end{aligned}$$

with $\boldsymbol{\alpha}(t)$ and $\boldsymbol{\beta}(t)$ polynomial basis functions depending on the knots $\boldsymbol{\tau}_0$ whose exact form does not matter here; $\boldsymbol{\tau}_i = (\tau_{i1}, \ldots, \tau_{ir})$ the "landmarks", such that $w_i(\tau_{0j}) = \tau_{ij}$ for $j = 1, \ldots, r$, $w_i(a) = a$ and $w_i(b) = b$; and $\mathbf{d}_i \in \mathbb{R}^{r+2}$ the tangents, such that $w_i'(\tau_{0j}) = d_{ij}$ for $j = 1, \ldots, r$, $w_i'(a+) = d_{i0}$ and $w_i'(b-) = d_{i,r+1}$. Given $\boldsymbol{\tau}_i$, a vector of derivatives $\mathbf{d}_i$ that guarantees the monotonicity of $w_i(t)$ everywhere is found using the (deterministic) Fritsch–Carlson algorithm; thus $\mathbf{d}_i = \mathbf{d}(\boldsymbol{\tau}_i)$ for some function $\mathbf{d} : \mathbb{R}^r \to \mathbb{R}^{r+2}$. Then

$$\mathrm{J}_{\boldsymbol{\tau}_i} w_i(t) = [\alpha_1(t), \ldots, \alpha_r(t)] + \boldsymbol{\beta}(t)^T \mathrm{J}\mathbf{d}(\boldsymbol{\tau}_i)$$

The $\boldsymbol{\tau}_i$'s are Jupp-transformed in order to remove the monotonicity constraints on its coordinates, so $\boldsymbol{\theta}_i = \mathbf{H}(\boldsymbol{\tau}_i)$ for an invertible $\mathbf{H} : \mathbb{R}^r \to \mathbb{R}^r$. Then

$$\mathrm{J}_{\boldsymbol{\theta}_i} w_i(t) = \mathrm{J}_{\boldsymbol{\tau}_i} w_i(t) \mathrm{J}\mathbf{H}^{-1}(\boldsymbol{\theta}_i) = \mathrm{J}_{\boldsymbol{\tau}_i} w_i(t) \{\mathrm{J}\mathbf{H}(\boldsymbol{\tau}_i)\}^{-1}.$$

More explicitly, for a given $\boldsymbol{\tau}$ the Jupp transform $\boldsymbol{\theta}$ is defined as

$$\begin{aligned}
\theta_1 &= \log\left(\frac{\tau_2 - \tau_1}{\tau_1 - a}\right) \\
\theta_j &= \log\left(\frac{\tau_{j+1} - \tau_j}{\tau_j - \tau_{j-1}}\right), \quad j = 2, \ldots, r-1, \\
\theta_r &= \log\left(\frac{b - \tau_r}{\tau_r - \tau_{r-1}}\right),
\end{aligned}$$

so

$$\begin{aligned}
\frac{\partial \theta_j}{\partial \tau_{j-1}} &= \frac{1}{\tau_j - \tau_{j-1}}, \\
\frac{\partial \theta_j}{\partial \tau_j} &= -\frac{1}{\tau_{j+1} - \tau_j} - \frac{1}{\tau_j - \tau_{j-1}}, \\
\frac{\partial \theta_j}{\partial \tau_{j+1}} &= \frac{1}{\tau_{j+1} - \tau_j},
\end{aligned}$$

and the other derivatives are zero. Then

$$\mathrm{J}\mathbf{H}(\boldsymbol{\tau}) = \begin{bmatrix}
-\delta_2 - \delta_1 & \delta_2 & & & & \\
\delta_2 & -\delta_3 - \delta_2 & \delta_3 & & & \\
& \delta_3 & -\delta_4 - \delta_3 & \delta_4 & & \\
& & \ddots & \ddots & \ddots & \\
& & & \delta_{r-1} & -\delta_r - \delta_{r-1} & \delta_r \\
& & & & \delta_r & -\delta_{r+1} - \delta_r
\end{bmatrix}$$

with $\delta_j = \tau_j - \tau_{j-1}$.

2.3.2 Smooth monotone warping functions

When $w_i(t)$ belongs to the family of Ramsay and Li's smooth monotone functions, we have

$$\log w_i'(t) = \boldsymbol{\theta}_i^T \boldsymbol{\phi}(t) + c$$

with $\boldsymbol{\phi}(t) \in \mathbb{R}^r$ a B-spline basis (strictly speaking Ramsay and Li model the derivative of $\log w_i'(t)$ using linear B-splines, but this is equivalent, up to a reparametrization, to using a quadratic B-spline basis for $\log w_i'(t)$ itself). Let

$$\xi(t, \boldsymbol{\theta}_i) = \int_a^t \exp\{\boldsymbol{\theta}_i^T \boldsymbol{\phi}(s)\} ds.$$

Then

$$w_i(t) = (b-a)\xi(t, \boldsymbol{\theta}_i)/\xi(b, \boldsymbol{\theta}_i) + a$$

satisfies the boundary conditions $w_i(a) = a$ and $w_i(b) = b$. The Jacobian is

$$\mathrm{J}_{\boldsymbol{\theta}_i} w_i(t) = \{(b-a)/\xi(b, \boldsymbol{\theta}_i)\}\mathrm{J}_{\boldsymbol{\theta}_i}\xi(t, \boldsymbol{\theta}_i) - \{(b-a)\xi(t, \boldsymbol{\theta}_i)/\xi^2(b, \boldsymbol{\theta}_i)\}\mathrm{J}_{\boldsymbol{\theta}_i}\xi(b, \boldsymbol{\theta}_i)$$

with

$$\mathrm{J}_{\boldsymbol{\theta}_i}\xi(t, \boldsymbol{\theta}_i) = \int_a^t \exp\{\boldsymbol{\theta}_i^T \boldsymbol{\phi}(s)\} \; \boldsymbol{\phi}(s)^T \, ds.$$

3 Hermite splines

3.1 Interpolation on a single interval

3.1.1 Interpolation on the interval $[0, 1]$

Given values of the function to be interpolated, f_0 at $t = 0$ and f_1 at $t = 1$, and values of the derivatives at those points, d_0 at $t = 0$ and d_1 at $t = 1$, then

$$f(t) = h_{00}(t)f_0 + h_{10}(t)d_0 + h_{01}(t)f_1 + h_{11}(t)d_1$$

with

$$\begin{aligned}
h_{00}(t) &= 2t^3 - 3t^2 + 1 = (1+2t)(1-t)^2 \\
h_{10}(t) &= t^3 - 2t^2 + t = t(1-t)^2 \\
h_{01}(t) &= -2t^3 + 3t^2 = t^2(3-2t) = h_{00}(1-t) \\
h_{11}(t) &= t^3 - t^2 = t^2(t-1) = -h_{10}(1-t)
\end{aligned}$$

satisfies

$$\begin{aligned} f(0) &= f_0, \;\; f(1) = f_1, \\ f'_+(0) &= d_0, \;\; f'_-(1) = d_1. \end{aligned}$$

Since $f(t)$ is a polynomial of degree 3 (i.e. has 4 free coefficients) and satisfies these four conditions, it's the *only* cubic polynomial that satisfies these conditions.

3.1.2 Interpolation on a general interval $[x_k, x_{k+1}]$

If we are now given values of the function f_k at $t = x_k$ and f_{k+1} at $t = x_{k+1}$, and values of the derivatives at those points, d_k at $t = x_k$ and d_{k+1} at $t = x_{k+1}$, then

$$f(t) = h_{00}\left(\frac{t - x_k}{s_k}\right) f_k + h_{10}\left(\frac{t - x_k}{s_k}\right) s_k d_k + h_{01}\left(\frac{t - x_k}{s_k}\right) f_{k+1} + h_{11}\left(\frac{t - x_k}{s_k}\right) s_k d_{k+1}$$

with $s_k = x_{k+1} - x_k$ and $h_{ij}(t)$ as before, satisfies

$$\begin{aligned} f(x_k) &= f_k, \;\; f(x_{k+1}) = f_{k+1}, \\ f'_+(x_k) &= d_k, \;\; f'_-(x_{k+1}) = d_{k+1}. \end{aligned}$$

Again, this cubic polynomial is unique, subject to these four conditions.

3.2 Interpolating a data set

Suppose we have p ponts in some interval $[a, b]$, $a < x_1 < x_2 < \cdots < x_p < b$, and corresponding values of the function, f_k, and the derivative, d_k, at each point x_k. Let's define $x_0 = a$, $x_{p+1} = b$ and f_0, f_{p+1}, d_0 and d_{p+1} accordingly. Then the piecewise cubic interpolant is going to be

$$
\begin{aligned}
f(t) &= \sum_{k=0}^{p}\left\{ h_{00}\left(\frac{t-x_k}{s_k}\right) f_k + h_{10}\left(\frac{t-x_k}{s_k}\right) s_k d_k \right. \\
&\quad \left. + h_{01}\left(\frac{t-x_k}{s_k}\right) f_{k+1} + h_{11}\left(\frac{t-x_k}{s_k}\right) s_k d_{k+1} \right\} \\
&= \sum_{k=0}^{p} h_{00}\left(\frac{t-x_k}{s_k}\right) f_k + \sum_{k=0}^{p} h_{10}\left(\frac{t-x_k}{s_k}\right) s_k d_k \\
&\quad + \sum_{k=1}^{p+1} h_{01}\left(\frac{t-x_{k-1}}{s_{k-1}}\right) f_k + \sum_{k=1}^{p+1} h_{11}\left(\frac{t-x_{k-1}}{s_{k-1}}\right) s_{k-1} d_k \\
&= h_{00}\left(\frac{t-x_0}{s_0}\right) f_0 + h_{10}\left(\frac{t-x_0}{s_0}\right) s_0 d_0 \\
&\quad + \sum_{k=1}^{p}\left\{ h_{00}\left(\frac{t-x_k}{s_k}\right) + h_{01}\left(\frac{t-x_{k-1}}{s_{k-1}}\right)\right\} f_k \\
&\quad + \sum_{k=1}^{p}\left\{ h_{10}\left(\frac{t-x_k}{s_k}\right) s_k + h_{11}\left(\frac{t-x_{k-1}}{s_{k-1}}\right) s_{k-1}\right\} d_k \\
&\quad + h_{01}\left(\frac{t-x_p}{s_p}\right) f_{p+1} + h_{11}\left(\frac{t-x_p}{s_p}\right) s_p d_{p+1}
\end{aligned}
$$

with $s_k = x_{k+1} - x_k$ as before, and we are defining $h_{ij}(t) = 0$ for $t \notin [0, 1]$.

Then we can introduce the basis functions

$$
\phi_k(t) = \begin{cases} h_{00}\left(\frac{t-x_0}{s_0}\right) & \text{if } k = 0 \\ h_{00}\left(\frac{t-x_k}{s_k}\right) + h_{01}\left(\frac{t-x_{k-1}}{s_{k-1}}\right) & \text{if } k = 1, \dots, p \\ h_{01}\left(\frac{t-x_p}{s_p}\right) & \text{if } k = p+1 \end{cases}
$$

and

$$
\psi_k(t) = \begin{cases} h_{10}\left(\frac{t-x_0}{s_0}\right) s_0 & \text{if } k = 0 \\ h_{10}\left(\frac{t-x_k}{s_k}\right) s_k + h_{11}\left(\frac{t-x_{k-1}}{s_{k-1}}\right) s_{k-1} & \text{if } k = 1, \dots, p \\ h_{11}\left(\frac{t-x_p}{s_p}\right) s_p & \text{if } k = p+1, \end{cases}
$$

and we have

$$
\boxed{f(t) = \sum_{k=0}^{p+1} \phi_k(t) f_k + \sum_{k=0}^{p+1} \psi_k(t) d_k}
$$

Since $h_{01}(t) = h_{00}(1-t)$ and $h_{11}(t) = -h_{10}(1-t)$,

$$\phi_k(t) = \begin{cases} h_{00}\left(\frac{t-x_0}{s_0}\right) & \text{if } k=0 \\ h_{00}\left(\frac{t-x_k}{s_k}\right) + h_{00}\left(\frac{x_k-t}{s_{k-1}}\right) & \text{if } k=1,\ldots,p \\ h_{00}\left(\frac{x_{p+1}-t}{s_p}\right) & \text{if } k=p+1 \end{cases}$$

and

$$\psi_k(t) = \begin{cases} h_{10}\left(\frac{t-x_0}{s_0}\right) s_0 & \text{if } k=0 \\ h_{10}\left(\frac{t-x_k}{s_k}\right) s_k - h_{10}\left(\frac{x_k-t}{s_{k-1}}\right) s_{k-1} & \text{if } k=1,\ldots,p \\ -h_{10}\left(\frac{x_{p+1}-t}{s_p}\right) s_p & \text{if } k=p+1. \end{cases}$$

For $k = 1, \ldots, p$ we have

$$\phi_k(t) = \begin{cases} 0 & \text{if } t < x_{k-1} \text{ or } t > x_{k+1} \\ h_{00}\left(\frac{x_k-t}{s_{k-1}}\right) & \text{if } x_{k-1} \le t \le x_k \\ h_{00}\left(\frac{t-x_k}{s_k}\right) & \text{if } x_k \le t \le x_{k+1} \end{cases}$$

and

$$\psi_k(t) = \begin{cases} 0 & \text{if } t < x_{k-1} \text{ or } t > x_{k+1} \\ -h_{10}\left(\frac{x_k-t}{s_{k-1}}\right) s_{k-1} & \text{if } x_{k-1} \le t \le x_k \\ h_{10}\left(\frac{t-x_k}{s_k}\right) s_k & \text{if } x_k \le t \le x_{k+1}. \end{cases}$$

Since $h_{00}(0) = 1$, $h_{00}(1) = 0$ and $h'_{00}(0) = h'_{00}(1) = 0$, the ϕ_ks are continuous and differentiable everywhere as functions of t, with $\phi_k(x_{k-1}) = \phi_k(x_{k+1}) = 0$, $\phi_k(x_k) = 1$ and $\phi'_k(x_{k-1}) = \phi'_k(x_k) = \phi'_k(x_{k+1}) = 0$. Similarly, since $h_{10}(0) = h_{10}(1) = 0$, $h'_{10}(0) = 1$ and $h'_{10}(1) = 0$, the ψ_ks are also continuous and differentiable everywhere as functions of t, with $\psi_k(x_{k-1}) = \psi_k(x_k) = \psi_k(x_{k+1}) = 0$, $\psi'_k(x_{k-1}) = \psi'_k(x_{k+1}) = 0$ and $\psi'_k(x_k) = 1$.

For the "border" basis functions we have

$$\phi_0(t) = \begin{cases} 0 & \text{if } t < a \text{ or } t > x_1 \\ h_{00}\left(\frac{t-a}{x_1-a}\right) & \text{if } a \le t \le x_1 \end{cases}$$

which is discontinuous only at $t = a$, with $\phi_0(a) = 1$, $\phi_0(x_1) = 0$, and $(\phi_0)'_+(a) = \phi'_0(x_1) = 0$;

$$\phi_{p+1}(t) = \begin{cases} 0 & \text{if } t < x_p \text{ or } t > b \\ h_{00}\left(\frac{b-t}{b-x_p}\right) & \text{if } x_p \le t \le b \end{cases}$$

which is discontinuous only at $t = b$, with $\phi_{p+1}(x_p) = 0$, $\phi_{p+1}(b) = 1$, and $\phi'_{p+1}(x_p) = \left(\phi_{p+1}\right)'_{-}(b) = 0$;

$$\psi_0(t) = \begin{cases} 0 & \text{if } t < a \text{ or } t > x_1 \\ h_{10}\left(\frac{t-a}{x_1-a}\right) s_0 & \text{if } a \le t \le x_1 \end{cases}$$

which is continuous everywhere, with $\psi_0(a) = \psi_0(x_1) = 0$, $(\psi_0)'_{+}(a) = 1$, and $\psi'_0(x_1) = 0$; and

$$\psi_{p+1}(t) = \begin{cases} 0 & \text{if } t < x_p \text{ or } t > b \\ -h_{10}\left(\frac{b-t}{b-x_p}\right) s_p & \text{if } x_p \le t \le b \end{cases}$$

which is continuous everywhere, with $\psi_{p+1}(x_p) = \psi_{p+1}(b) = 0$, $\psi'_{p+1}(x_p) = 0$, and $\left(\psi_{p+1}\right)'_{-}(b) = 1$.

3.3 Monotone interpolation

Given $a = x_0 < x_1 < \cdots < x_p < x_{p+1} = b$ and $f_0 < \cdots < f_{p+1}$, with d_ks unspecified, it's always possible to find d_ks such that the resulting $f(t)$ is strictly increasing (Fritsch and Carlson, 1980). Let

$$\Delta_k = \frac{f_{k+1} - f_k}{x_{k+1} - x_k}, \quad \alpha_k = \frac{d_k}{\Delta_k}, \quad \beta_k = \frac{d_{k+1}}{\Delta_k}.$$

Then $f(t)$ is monotone in $[x_k, x_{k+1}]$ if and only if:

1. $\alpha_k + \beta_k - 2 \le 0$ and $\mathrm{sign}(d_k) = \mathrm{sign}(d_{k+1}) = \mathrm{sign}(\Delta_k)$; or
2. $\alpha_k + \beta_k - 2 > 0$, $\mathrm{sign}(d_k) = \mathrm{sign}(d_{k+1}) = \mathrm{sign}(\Delta_k)$, and:
   (a) $2\alpha_k + \beta_k - 3 \le 0$, or
   (b) $\alpha_k + 2\beta_k - 3 \le 0$, or
   (c) $\alpha_k - (2\alpha_k + \beta_k - 3)^2/\{3(\alpha_k + \beta_k - 2)\} \ge 0$.

The condition

$$\sqrt{\alpha_k^2 + \beta_k^2} \le 3$$

implies either 1 or 2(a)–2(c) above, so it's sufficient to guarantee monotonicity. This motivates the following algorithm for constructing the d_ks:

1. Initialize the derivatives $\{d_k\}$ so that $\text{sign}(d_k) = \text{sign}(d_{k+1}) = \text{sign}(\Delta_k)$. For instance,
$$d_0 = \Delta_0, \;\; d_k = \frac{\Delta_{k-1} + \Delta_k}{2} \text{ for } k = 1, \ldots, p, \;\; d_{p+1} = \Delta_p.$$
2. For $k = 0, \ldots, p$:
   (a) If $\sqrt{\alpha_k^2 + \beta_k^2} \leq 3$ the interpolant will be monotone in $[x_k, x_{k+1}]$; go to next k.
   (b) If $\sqrt{\alpha_k^2 + \beta_k^2} > 3$, let $\tau_k = 3/\sqrt{\alpha_k^2 + \beta_k^2}$, $\alpha_k^* = \tau_k \alpha_k$, and $\beta_k^* = \tau_k \beta_k$; set
$$d_k = \alpha_k^* \Delta_k, \;\; d_{k+1} = \beta_k^* \Delta_k.$$
   The interpolant will be monotone in $[x_k, x_{k+1}]$; go to next k.

The algorithm may change the value of each d_k at most twice from its initial value: first when the interval $[x_{k-1}, x_k]$ is considered and again when the interval $[x_k, x_{k+1}]$ is considered. But since $0 \leq \alpha_k^* \leq \alpha_k$ and $0 \leq \beta_k^* \leq \beta_k$, the modification of d_k for $[x_k, x_{k+1}]$ will maintain the monotonicity condition on $[x_{k-1}, x_k]$; see comments on p. 241 of Fritsch and Carlson (1980).

References

Fritsch, F.N. and Carlson, R.E. (1980). Monotone piecewise cubic interpolation. *SIAM J. Numer. Anal.* **17** 238–246.