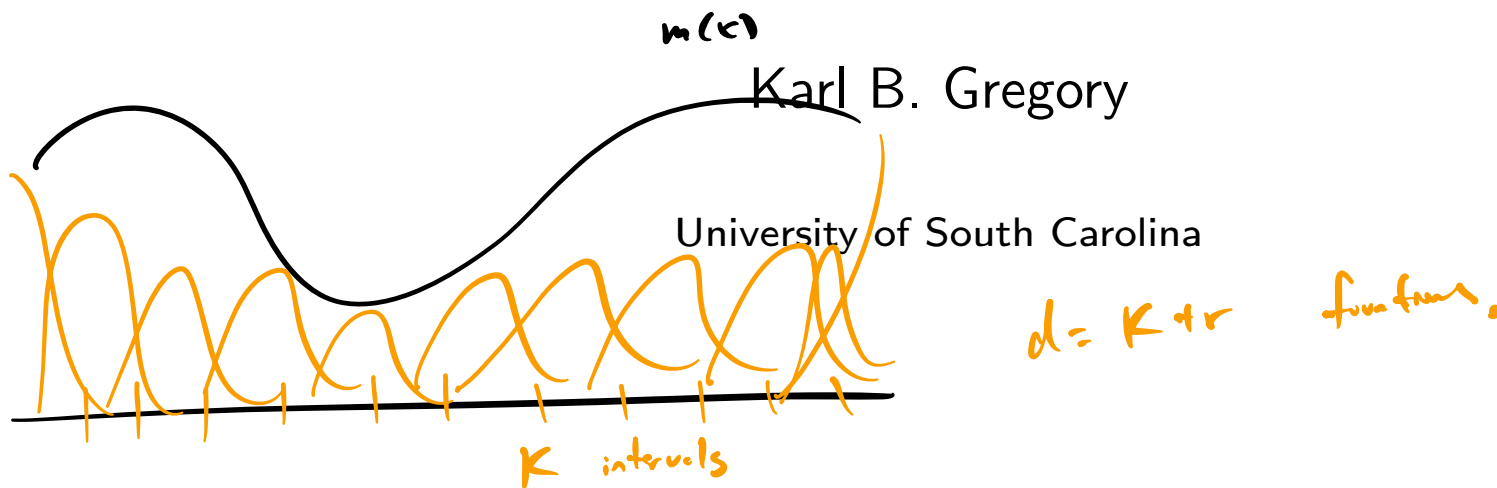


# STAT 824 sp 2023 Lec 06 slides

## Nonparametric regression: Penalized splines and trend filtering



These slides are an instructional aid; their sole purpose is to display, during the lecture, definitions, plots, results, etc. which take too much time to write by hand on the blackboard. They are not intended to explain or expound on any material.

# Table of Contents

- 1 Penalized splines
- 2 Connection between penalized splines and kernel smoothing
- 3 Trend filtering

Let  $(X_1, Y_1), \dots, (X_n, Y_n)$  be indep. realizations of  $(X, Y) \in [0, 1] \times \mathbb{R}$ , where

$$Y = m(X) + \varepsilon, \quad \text{for some } m : [0, 1] \rightarrow \mathbb{R},$$

where  $\varepsilon$  is independent of  $X$  with  $\mathbb{E}\varepsilon = 0$  and  $\mathbb{E}\varepsilon^2 = \sigma^2$ .

Define the class of functions (which is a Sobolev space)

$$f(x) = mx + b$$

$$f'(x) = m \quad f''(x) = 0$$

$$\mathcal{W}_2 = \left\{ g : [0, 1] \rightarrow \mathbb{R} : g' \text{ is continuous, } \int_0^1 [g''(x)]^2 dx < \infty \right\}.$$

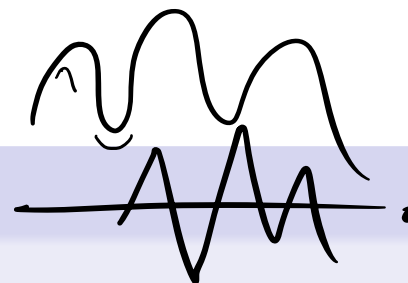
Note that  $\mathcal{H}(2, L)$  on  $[0, 1]$  is contained in  $\mathcal{W}_2$ .

## Smoothing spline estimator

The estimator

$$\hat{m}_n^{\text{sspl}} = \operatorname{argmin}_{g \in \mathcal{W}_2} \sum_{i=1}^n [Y_i - g(X_i)]^2 + \lambda \int_0^1 [g''(x)]^2 dx,$$

for  $\lambda > 0$  is called the *smoothing spline estimator* of  $m$ .



How do we search among all functions belonging to  $\mathcal{W}_2$ ?

Beautiful result:  $\hat{m}_n^{\text{sspl}}$  is a natural cubic spline with knots  $u_i = X_i, i = 1, \dots, n$ .

*Natural cubic splines* are cubic splines constrained to be linear beyond end knots.

Can bound MSE by  $C \cdot n^{-4/5}$  when  $\lambda = c \cdot n^{1/5}$ . See Grace Wahba's book, [4].

Sets of basis functions for natural cubic splines lack nice properties of B-splines. Since we love B-splines, we often consider (instead of smoothing splines) this:

### Penalized spline estimator

Let  $\mathcal{M}_{n,3}$  be the space of cubic splines on the knots

$$u_{-3} = u_{-2} = u_{-1} = u_0 < u_1 < \dots < u_{K_n} = u_{K_n+1} = u_{K_n+2} = u_{K_n+3}.$$

Then  $\hat{m}_n^{\text{pspl}}(x) = \sum_{j=1}^d \hat{a}_j b_j(x)$ ,  $\hat{a} = \underset{a \in \mathbb{R}^d}{\text{argmin}} \left\{ \|Y - B a\|_2^2 + \lambda a^T \Omega a \right\}$

$$\hat{m}_n^{\text{pspl}} = \underset{g \in \mathcal{M}_{n,3}}{\text{argmin}} \sum_{i=1}^n [Y_i - \underline{g(X_i)}]^2 + \lambda \int_0^1 [g''(x)]^2 dx,$$

for  $\lambda > 0$  is the *penalized spline estimator* of  $m$ . Nice reference is [1].

Idea is to choose  $K_n$  very large and then tune wiggleness by choosing  $\lambda$ .

When  $K_n$  is very large,  $\hat{m}_n^{\text{pspl}}$  is practically identical to  $\hat{m}_n^{\text{sspl}}$ .

**Exercise:** Give a representation of  $\hat{m}_n^{\text{pspl}}(x_0)$  in matrices given a basis for  $\mathcal{M}_{n,3}$ .

$$m(x) \approx \sum_{j=1}^d \alpha_j b_j(x)$$

$$f(x) = \sum_{j=1}^d \alpha_j b_j(x)$$

$$\sum_{i=1}^n [y_i - f(x_i)]^2 = \sum_{i=1}^n [y_i - \sum_{j=1}^d \alpha_j b_j(x_i)]^2 = \|y - B\alpha\|_2^2.$$

$$\lambda \int_0^2 [f''(x)]^2 dx = \dots = \alpha^T \Omega \alpha$$

$$f''(x) = \sum_{j=1}^d \alpha_j b_j''(x)$$

$$[f''(x)]^2 = \left( \sum_{j=1}^d \alpha_j b_j''(x) \right)^2 = \sum_{j=1}^d \sum_{k=1}^d \alpha_j \alpha_k b_j''(x) b_k''(x)$$

$$\int_0^2 [f''(x)]^2 dx = \sum_{j=1}^d \sum_{k=1}^d \alpha_j \alpha_k \underbrace{\int_0^2 b_j''(x) b_k''(x) dx}_{\Omega_{jk}}$$

$$= \alpha^T \Omega \alpha \quad \Omega = (\Omega_{jk})_{1 \leq j, k \leq d}$$

Now I want to find

$$\hat{\alpha} = \underset{\alpha \in \mathbb{R}^d}{\text{argmin}} \left\{ (y - B\alpha)^T (y - B\alpha) + \lambda \alpha^T \Omega \alpha \right\}$$

$$\frac{\partial}{\partial \alpha} \left[ (y - B\alpha)^T (y - B\alpha) + \lambda \alpha^T \Omega \alpha \right]$$

(Matrix Cookbook)

$$= \frac{\partial}{\partial \alpha} \left[ y^T y - 2y^T B\alpha + \alpha^T B^T B \alpha + \lambda \alpha^T \Omega \alpha \right]$$

$$= -2B^T y + 2B^T B \alpha + 2\lambda \Omega \alpha \stackrel{\text{set}}{=} 0$$

$$(B^T B + \lambda \Omega) \alpha = B^T y$$

$$\hat{\alpha} = (B^T B + \lambda \Omega)^{-1} B^T y.$$

$$\frac{\partial}{\partial x} b^T x = b$$

$$\frac{\partial}{\partial x} x^T \Omega x = 2\Omega x \quad (\Omega \text{ sym.})$$

Unpenalized splines:

$$\hat{m}_n = \underset{f \in \mathcal{M}_{n,r}}{\operatorname{arg\,min}} \sum_{i=1}^n (y_i - f(x_i))^2$$

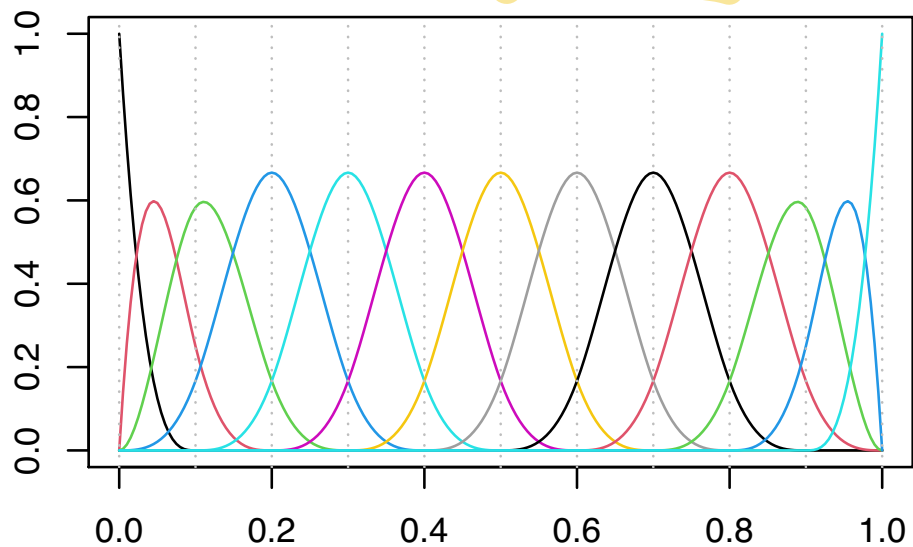
↑ splines of order  $r$

We can obtain the row vector  $\mathbf{b}''(x) = (b_1''(x), \dots, b_d''(x))$  with

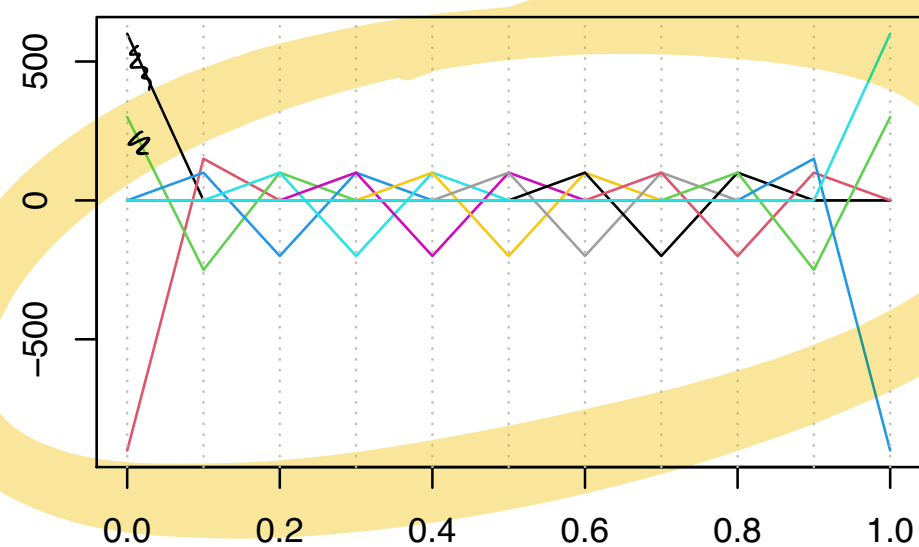
`splineDesign(knots=knots, x=x, ord=4, derivs=rep(2, K+1))`

where `knots` is the complete set of knots  $u_{-3}, \dots, u_{K+3}$ .

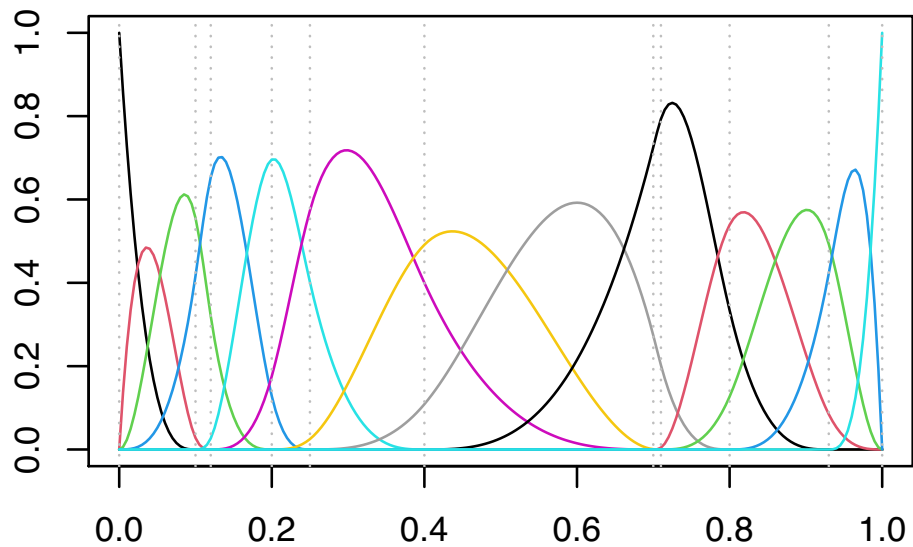
B-splines of order  $r = 3$



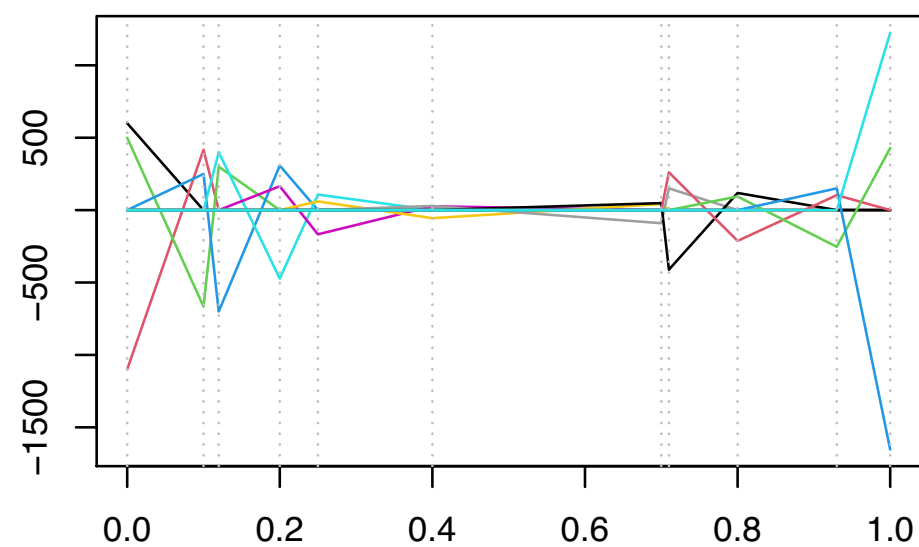
2nd derivatives of cubic B-splines



B-splines of order  $r = 3$



2nd derivatives of cubic B-splines





## Computation of $\Omega$

For cubic B-splines basis functions  $b_1, \dots, b_{d_n}$ ,  $d_n = K_n + 3$ , based on knots  $u_{-3} = u_{-2} = u_{-1} = u_0 < u_1 < \dots < u_{K_n} = u_{K_n+1} = u_{K_n+2} = u_{K_n+3}$ , we have

$$\int_0^1 \underbrace{b_\ell''(x)} \underbrace{b_j''(x)} dx = \sum_{k=0}^{K-1} (u_{k+1} - u_k) \left[ \frac{1}{2} (b_j''(u_k) b_\ell''(u_{k+1}) + b_j''(u_{k+1}) b_\ell''(u_k)) \right. \\ \left. + \frac{1}{3} (b_\ell''(u_{k+1}) - b_\ell''(u_k))(b_j''(u_{k+1}) - b_j''(u_k)) \right]$$

for each  $1 \leq j, \ell \leq d_n$ .

We can derive the above using the fact that each  $b_\ell''$  is piecewise linear.

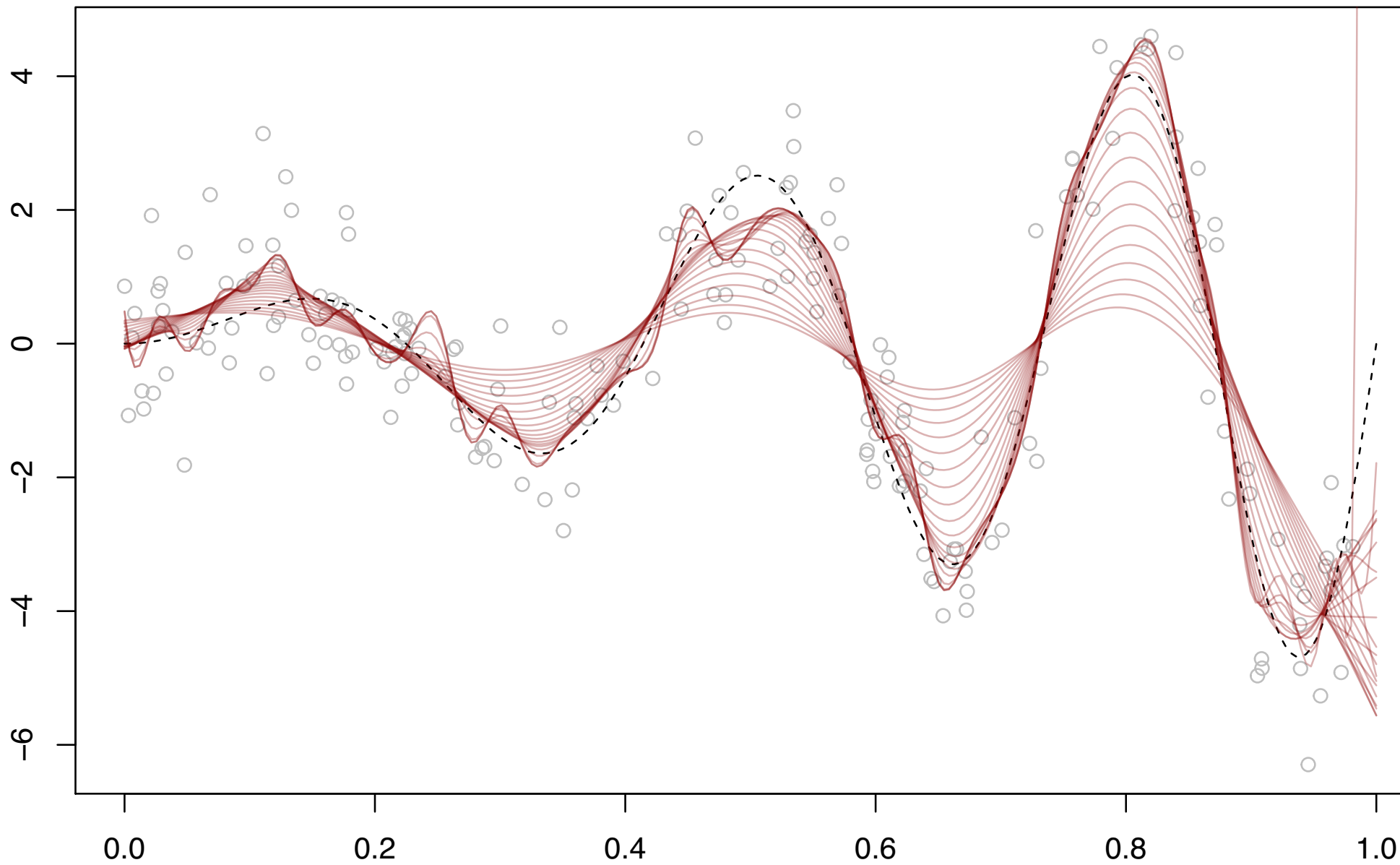
**Exercise:** Demonstrate fitting the penalized splines estimator.

Once I have  $\hat{\mathbf{a}}_{\tilde{n}}$ , I evaluate  $\hat{w}_n$  at  $x_0$  with

$$\hat{w}_n(x_0) = \mathbf{b}_{\tilde{n}x_0}^T \hat{\mathbf{a}}_{\tilde{n}} = \sum_{j=1}^{\tilde{n}} \hat{a}_j b_j(x_0), \quad \mathbf{b}_{\tilde{n}x_0} = \begin{pmatrix} b_1(x_0) \\ \vdots \\ b_d(x_0) \end{pmatrix}$$

$$\mathbf{B}_{\tilde{n}} = \begin{pmatrix} \mathbf{b}_{\tilde{n}x_1}^T \\ \vdots \\ \mathbf{b}_{\tilde{n}x_p}^T \end{pmatrix} = \begin{pmatrix} b_1(x_1) & \dots & b_d(x_1) \\ \vdots & \ddots & \vdots \\ b_1(x_p) & \dots & b_d(x_p) \end{pmatrix}, \quad x_1, \dots, x_p \text{ a grid of values}$$

$$\begin{pmatrix} \hat{w}_n(x_1) \\ \vdots \\ \hat{w}_n(x_p) \end{pmatrix} = \mathbf{B}_{\tilde{n}} \hat{\mathbf{a}}_{\tilde{n}}$$



Note that we may write

$$(\hat{m}_n^{\text{pspl}}(X_1), \dots, \hat{m}_n^{\text{pspl}}(X_n))^T = \mathbf{S}\mathbf{Y},$$

where  $\mathbf{S} = \mathbf{B}^T(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{\Omega})^{-1}\mathbf{B}^T$ , with

$$\mathbf{B} = (b_\ell(X_i))_{1 \leq i \leq n, 1 \leq \ell \leq d_n} \quad \text{and} \quad \mathbf{\Omega} = \left( \int_0^1 b_\ell''(x)b_j''(x)dx \right)_{1 \leq \ell, j \leq d}.$$

The  $n \times n$  matrix  $\mathbf{S}$  is called a *smoother matrix*.

## Linear estimator and smoother matrix

A *linear estimator* is any estimator  $\hat{m}_n$  of the form

$$\hat{m}_n(x) = \sum_{i=1}^n W_{ni}(x)Y_i \quad \text{for some weights } W_{n1}(x), \dots, W_{nn}(x).$$

The *smoother matrix* associated with  $\hat{m}_n$  is the matrix  $\mathbf{S} = (W_{ni}(X_{i'}))_{1 \leq i, i' \leq n}$ .

**Exercise:** Plot some rows of a penalized spline smoother matrix  $\mathbf{S}$ . Discuss.

To compute the estimated function at the "design points" (observed values of  $x$ ).

$$\begin{bmatrix} \hat{m}_n(x_1) \\ \vdots \\ \hat{m}_n(x_n) \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^d \hat{\alpha}_j b_j(x_1) \\ \vdots \\ \sum_{j=1}^d \hat{\alpha}_j b_j(x_n) \end{bmatrix}$$

$$= \begin{bmatrix} \hat{\alpha}_1 b_1(x_1) + \dots + \hat{\alpha}_d b_d(x_1) \\ \vdots \\ \hat{\alpha}_1 b_1(x_n) + \dots + \hat{\alpha}_d b_d(x_n) \end{bmatrix}$$

$$= \begin{bmatrix} b_1(x_1) & \dots & b_d(x_1) \\ \vdots & & \vdots \\ b_1(x_n) & \dots & b_d(x_n) \end{bmatrix} \begin{pmatrix} \hat{\alpha}_1 \\ \vdots \\ \hat{\alpha}_d \end{pmatrix}$$

$$= \mathbf{B} \hat{\alpha}$$

$$= \mathbf{B} (\mathbf{B}^T \mathbf{B} + \lambda \mathbf{\Omega})^{-1} \mathbf{B}^T \mathbf{y}$$

$$= \mathbf{S} \mathbf{y}$$

↑  
"smoother matrix"

$$\hat{m}_n(x_i) = (\text{row } i \text{ of } \mathbf{S}) \mathbf{y} = \sum_{k=1}^n S_{ik} y_k = \sum_{k=1}^n W_{nk}(x_i) y_k$$

Linear estimator:

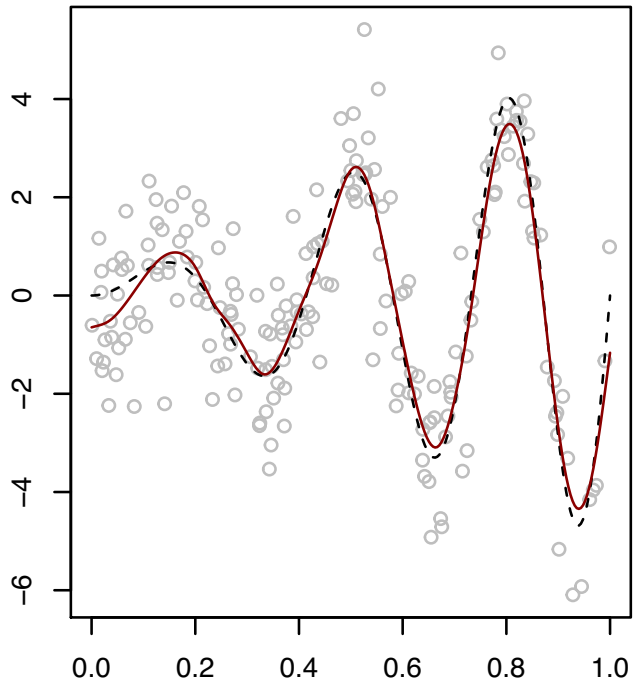
$$\hat{m}(x_0) = \sum_{i=1}^n w_{ni}(x_0) y_i, \quad \text{so}$$

$$\begin{bmatrix} \hat{m}_n(x_1) \\ \vdots \\ \hat{m}_n(x_n) \end{bmatrix} = \begin{bmatrix} \sum w_{ni}(x_1) y_i \\ \vdots \\ \sum w_{ni}(x_n) y_i \end{bmatrix}$$

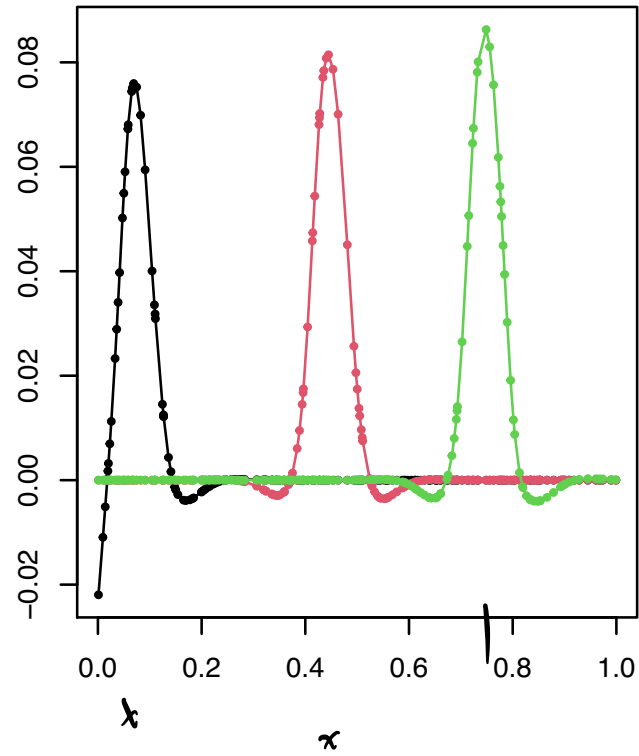
$$= \begin{bmatrix} w_{n1}(x_1) & \dots & w_{nn}(x_1) \\ \vdots & & \vdots \\ w_{n1}(x_n) & \dots & w_{nn}(x_n) \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$$= \mathbf{S} \mathbf{y}$$

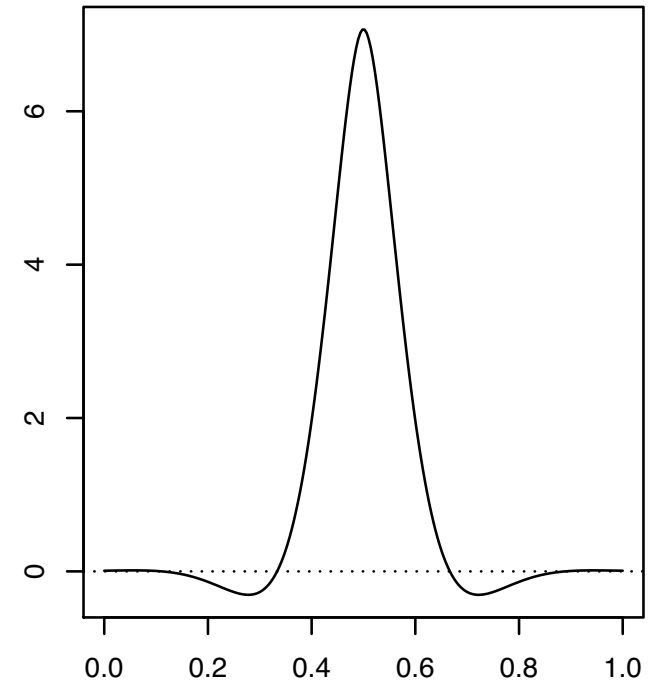
Penalized spline fit



Weights for fitted values  $i = 20, 100, 160$



Silverman Kernel



## Silverman's kernel, [2]

For  $X_1, \dots, X_n$  having the density  $f$  on  $[0, 1]$ , the smoothing spline estimator is asymptotically equivalent to N-W under  $K$  and local bandwidth  $h(x)$  given by

$$K(u) = \frac{1}{2} e^{-|u|/\sqrt{2}} \cdot \sin(|u|/\sqrt{2} + \pi/4) \quad \text{and} \quad h(x) = (n^{-1} \lambda / f(x))^{1/4}.$$

The previous slide shows a plot of the Silverman kernel function.

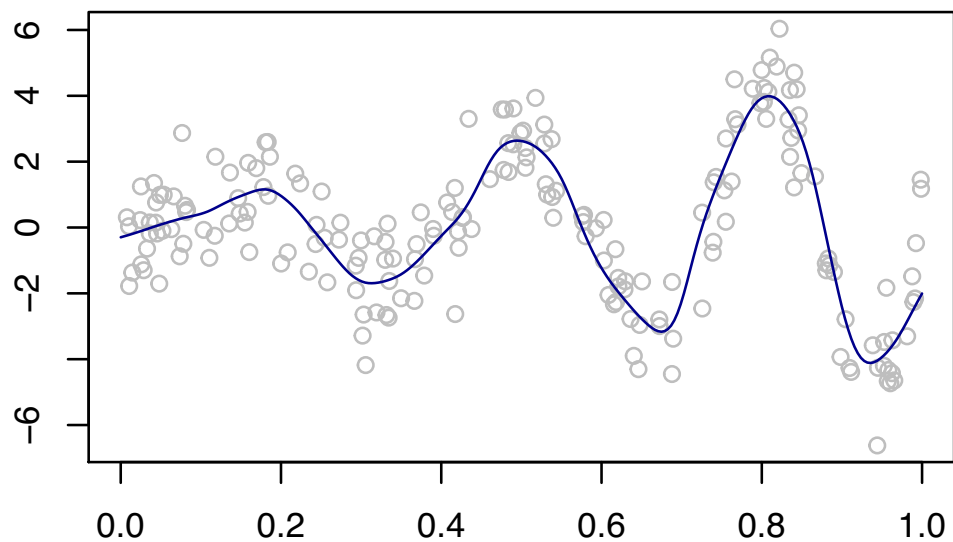
**Exercise:** Make some plots comparing the rows of the smoother matrices

- ① of a penalized spline estimator.
- ② of the N-W estimator under Silverman's kernel with bandwidth  $h = (\lambda/n)^{1/4}$ .

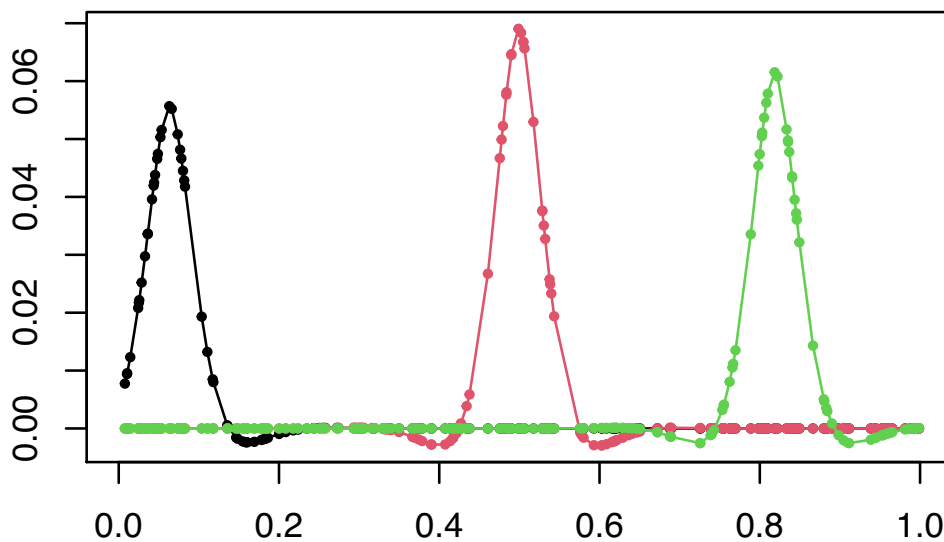
For the exercise, generate  $X_1, \dots, X_n \stackrel{\text{ind}}{\sim} \text{Uniform}(0, 1)$ .



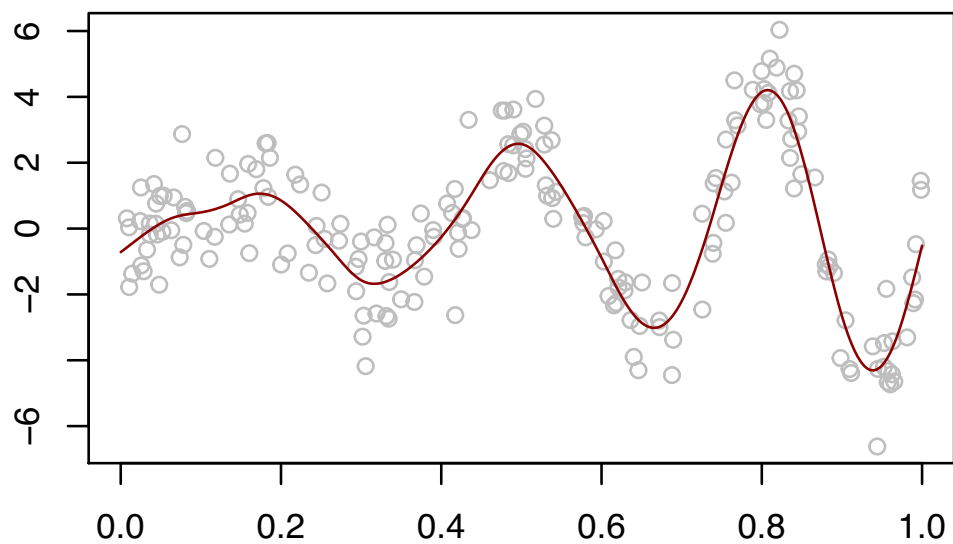
N-W under Silverman kernel



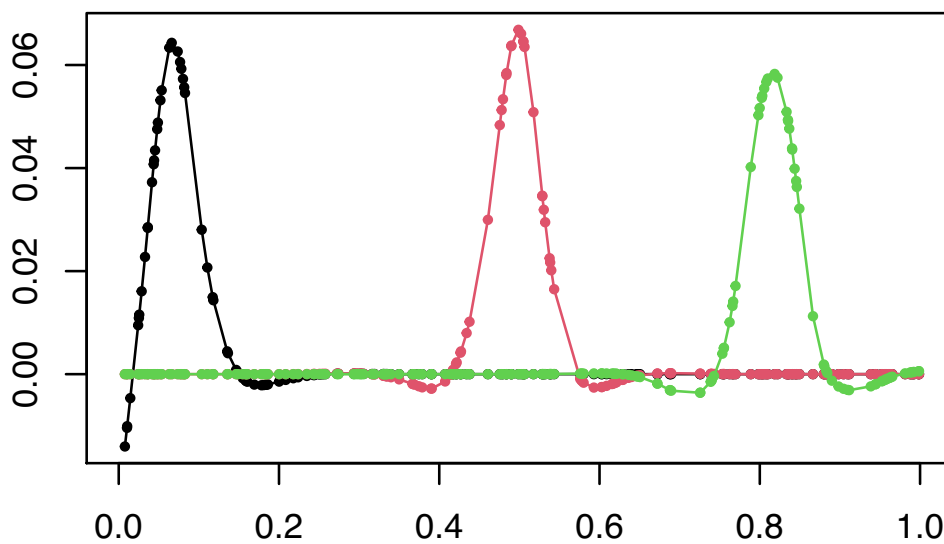
Some rows of the smoother matrix



Penalized spline fit



Some rows of the smoother matrix



Consider the eigendecomposition of the smoother matrix

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}_n, \quad \mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n),$$

where  $\lambda_1 \geq \lambda_2 \geq \dots \lambda_n \geq 0$  are the eigenvalues.

We see that the vector of fitted values can be written as

$$(\hat{m}_n^{\text{pspl}}(X_1), \dots, \hat{m}_n^{\text{pspl}}(X_n))^T = \mathbf{S}\mathbf{Y} = \mathbf{U}\mathbf{\Lambda}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{Y}.$$

**Discuss:** Can we learn anything from this?

**Exercise:** Inspect eigenvectors/eigenvalues of the smoother matrices

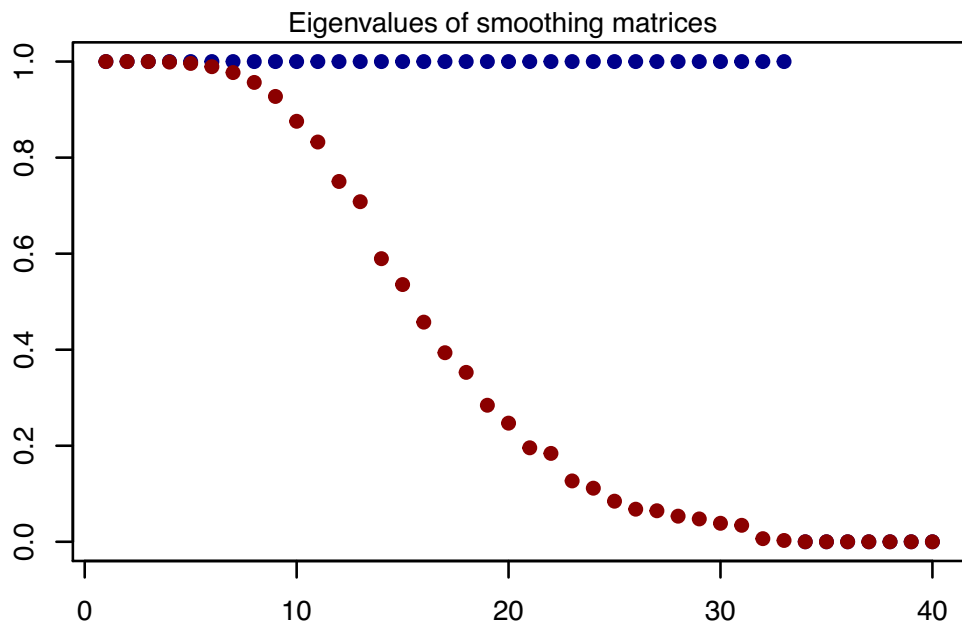
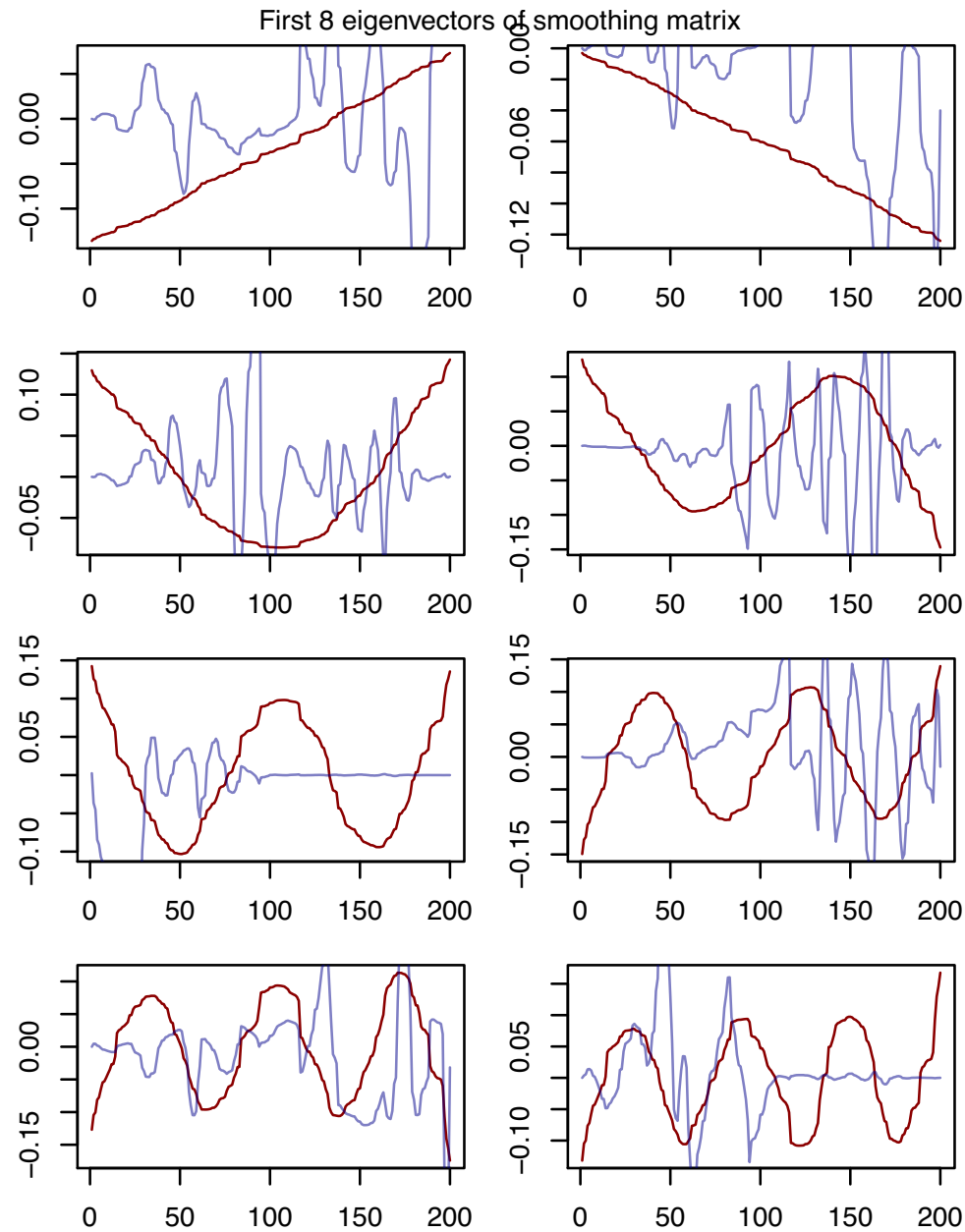
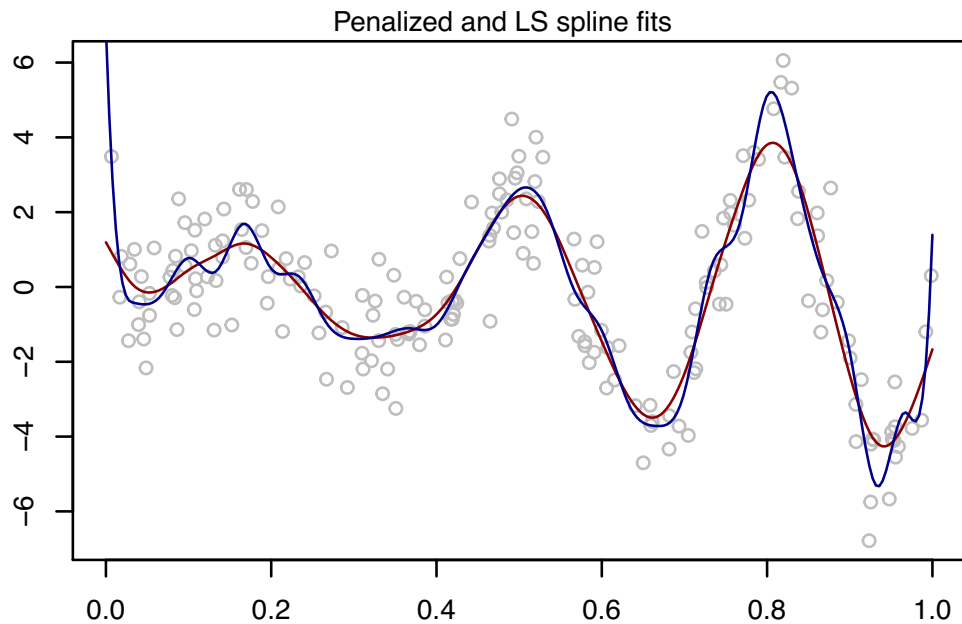
- 1  $\mathbf{S} = \mathbf{B}^T(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T$
- 2  $\mathbf{S} = \mathbf{B}^T(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{\Omega})^{-1}\mathbf{B}^T$

Make plots and discuss.

$$\begin{aligned}
 \hat{\beta} &= \begin{bmatrix} \hat{\beta}_1(x_1) \\ \vdots \\ \hat{\beta}_n(x_n) \end{bmatrix} = \delta \tilde{y} = U \Lambda U^T \tilde{y} \\
 &= U \Lambda \underbrace{(U^T U)^{-1}}_{I_n} \tilde{y} \\
 &= U \Lambda (U^T U)^{-1} U^T \tilde{y} \\
 &= U \hat{\beta} \\
 &= U \begin{pmatrix} \lambda_1 \hat{\beta}_1 \\ \vdots \\ \lambda_n \hat{\beta}_n \end{pmatrix} \\
 &= U \begin{pmatrix} \lambda_1 & \hat{\beta}_1 \\ \vdots & \vdots \\ \lambda_n & \hat{\beta}_n \end{pmatrix}
 \end{aligned}$$

If I regress  $\tilde{y}$  on  $U$ , we would get

$$\hat{\beta} = \underset{\beta}{\text{argmin}} \| \tilde{y} - U \beta \|_2^2, \quad \hat{\beta} = (U^T U)^{-1} U^T \tilde{y}$$



Show that  $B(B^T B)^{-1} B^T$  has  $d$  nonzero eigenvalues equal to 1:

Unpenalized:

$$\hat{\alpha} = (B^T B)^{-1} B^T Y$$
$$\hat{\alpha}_{\text{design matrix}} = B \hat{\alpha} = \underbrace{B (B^T B)^{-1} B^T}_{S} Y$$

$B (B^T B)^{-1} B^T$  MUST have eigenvalues of 0 and 1.  
Also if  $B$  is  $n \times d$ , full rank, then it has exactly  $d$  nonzero eigenvalues.

Have  $Y$ , want to find the linear combination of columns of  $B$  which is closest to  $Y$ , I did

$$P_B Y = \underbrace{B (B^T B)^{-1} B^T}_{S} Y$$

$$B (B^T B)^{-1} B^T B (B^T B)^{-1} B^T = B (B^T B)^{-1} B^T$$

Idempotent Matrices can only have eigenvalues of 0 and 1.

trace of a matrix is = sum of eigen values:

$$\text{tr}(S) = \text{tr}(U \Lambda U^T) = \text{tr}(U^T U \Lambda) = \text{tr}(\Lambda) = \sum_j \lambda_j$$

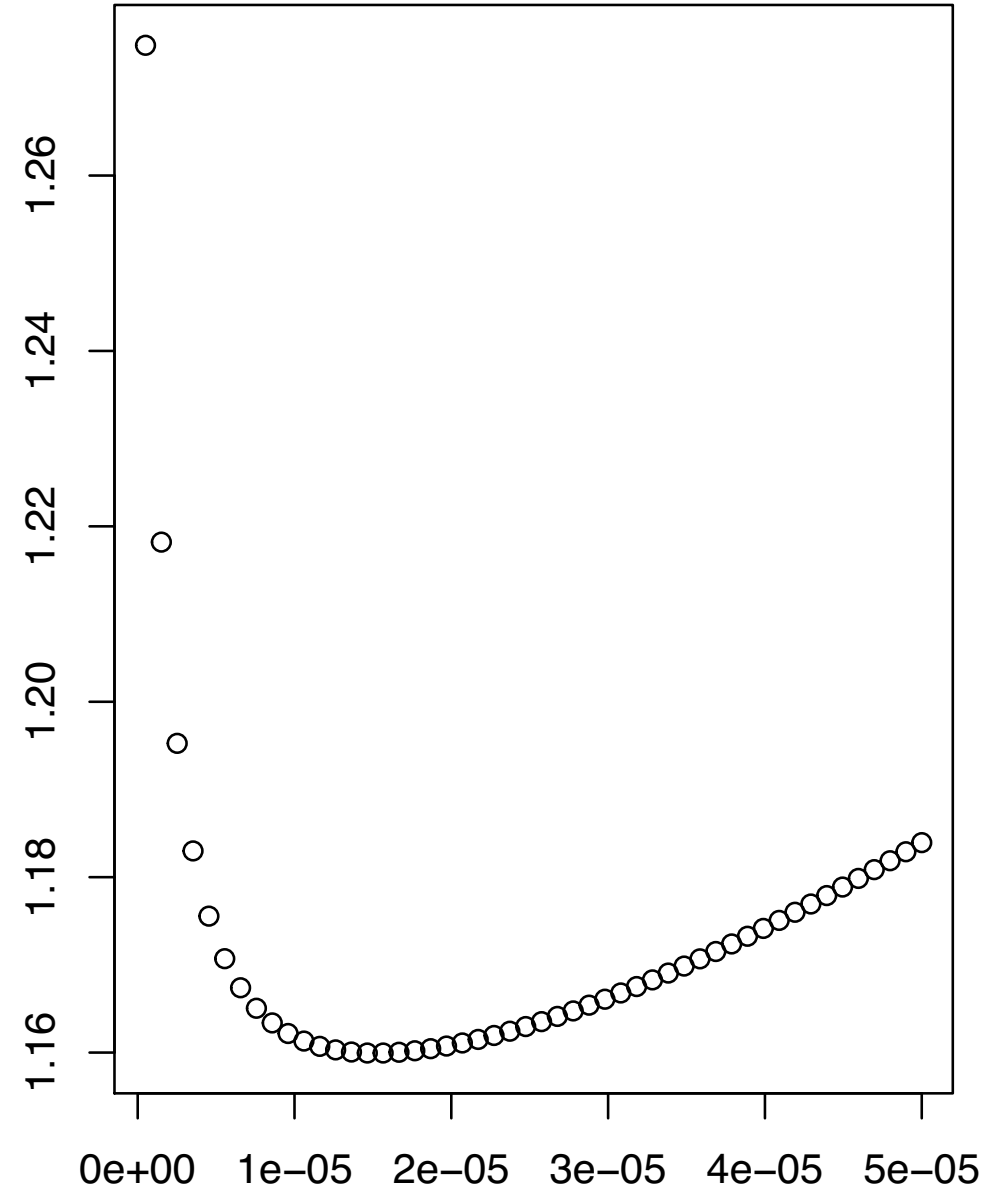
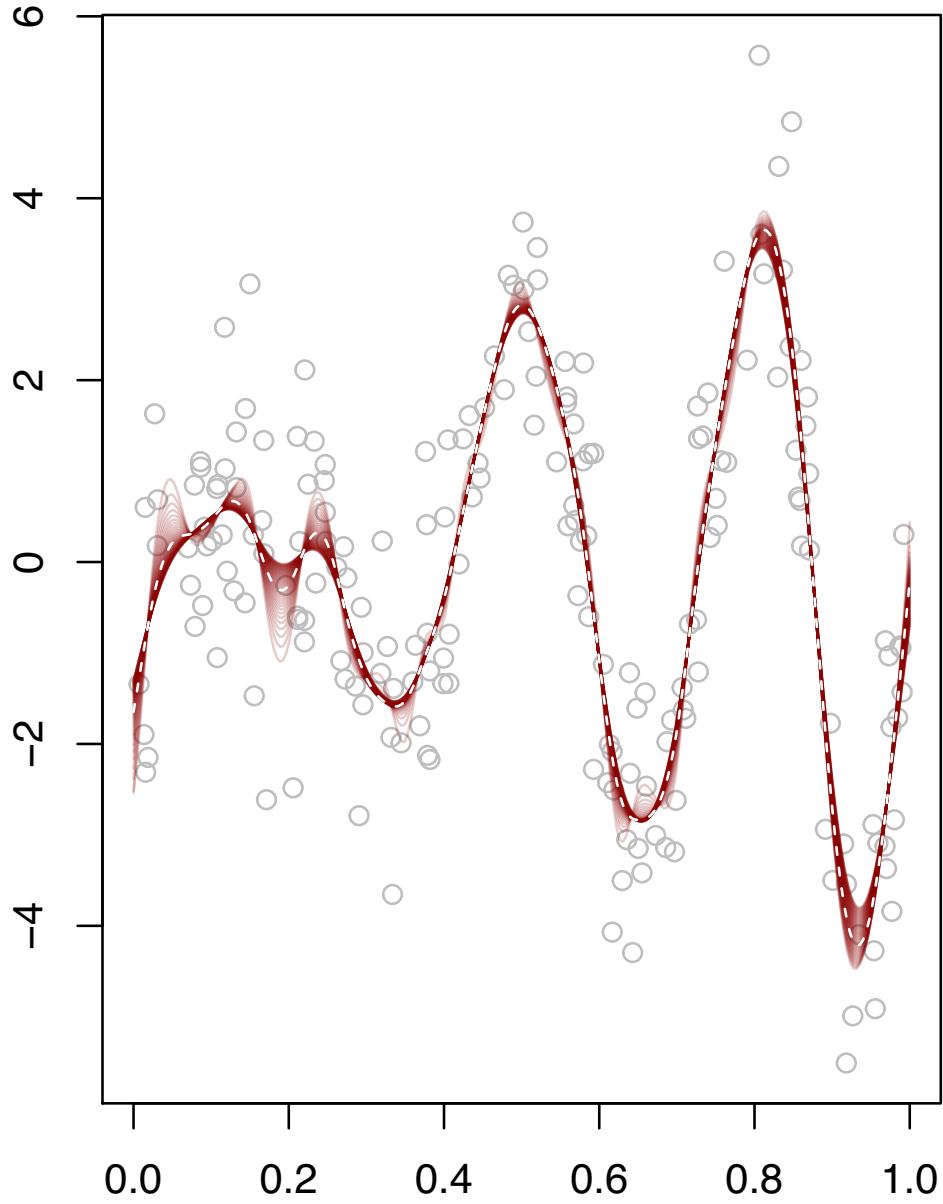
$$\text{tr}(B (B^T B)^{-1} B^T) = \text{tr}(B^T B (B^T B)^{-1}) = \text{tr}(I_d) = d.$$

Since  $\hat{m}_n^{\text{pspl}}$  is a linear estimator, we have

$$\text{CV}_n(h) = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{m}_{n,-i}^{\text{pspl}}(X_i))^2 = \frac{1}{n} \sum_{i=1}^n \left[ \frac{Y_i - \hat{m}_n^{\text{pspl}}(X_i)}{1 - S_{ii}} \right]^2,$$

where  $S_{ii}$  is the element  $i$  on the diagonal of the smoother matrix  $\mathbf{S}$ .

Illustrate in R.



Suppose  $X_i = i/n$ ,  $i = 1, \dots, n$ , for now, and let  $\boldsymbol{\mu} = (m(X_1), \dots, m(X_n))^T$ .

Trend filtering estimator (good reference is Tibshirani's paper, [3])

The *trend filtering estimator* of  $\boldsymbol{\mu}$  is given by  $y_i = m\left(\frac{i}{n}\right) + \varepsilon_i$

$$\hat{\boldsymbol{\mu}} = \operatorname{argmin}_{\mathbf{u} \in \mathbb{R}^n} \|\mathbf{Y} - \mathbf{u}\|_2^2 + \lambda \|\mathbf{D}^{(k+1)} \mathbf{u}\|_1, \quad \|\tilde{\mathbf{x}}\|_1 = \sum_i |x_i|$$

where

$$D^{(1)} = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix} \quad D^{(2)}$$

and  $D^{(k+1)} = D^{(1)} D^{(k)}$ ,  $k \geq 1$ , adjusting the dimension of  $D^{(1)}$  as needed.

To estimate  $m$  at any  $x \in [0, 1]$ , we can just linearly interpolate  $\boldsymbol{\mu}$ .

Accommodate unequally spaced inputs with a modification to  $\mathbf{D}^{(k+1)}$ . See [3].

**Exercise:** Study the penalties  $\lambda \|\mathbf{D}^{(1)} \mathbf{u}\|_1$ ,  $\lambda \|\mathbf{D}^{(2)} \mathbf{u}\|_1$ ,  $\lambda \|\mathbf{D}^{(3)} \mathbf{u}\|_1$ , and  $\lambda \|\mathbf{D}^{(4)} \mathbf{u}\|_1$  and consider their effects on  $\hat{\boldsymbol{\mu}}$ .



$$\|D^{(1)} \tilde{u}\|_2 = \left\| \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_n \end{bmatrix} \right\|_2$$


$$= \left\| \begin{bmatrix} u_1 - u_2 \\ u_2 - u_3 \\ \vdots \\ u_{n-1} - u_n \end{bmatrix} \right\|_2 = |u_1 - u_2| + |u_2 - u_3| + \dots$$

$$= |u_2 - u_1| + |u_3 - u_2| + \dots$$

permutation changes

$$D^{(2)} = \begin{matrix} D^{(1)} & D^{(1)} \\ (n-2) \times (n-1) & (n-1) \times n \end{matrix}$$

$$\|D^{(2)} \tilde{u}\|_2 = |u_1 - 2u_2 + u_3| + |u_2 - 2u_3 + u_4|$$

$$= |(u_3 - u_2) - (u_2 - u_1)| + \dots$$

permutation changes + change

$$\|D^{(3)} \tilde{u}\|_2 = |(u_{n-1} - u_n) - (u_1 - u_2)| - |(u_3 - u_2) - (u_2 - u_1)| + \dots$$

permutation changes + change + change

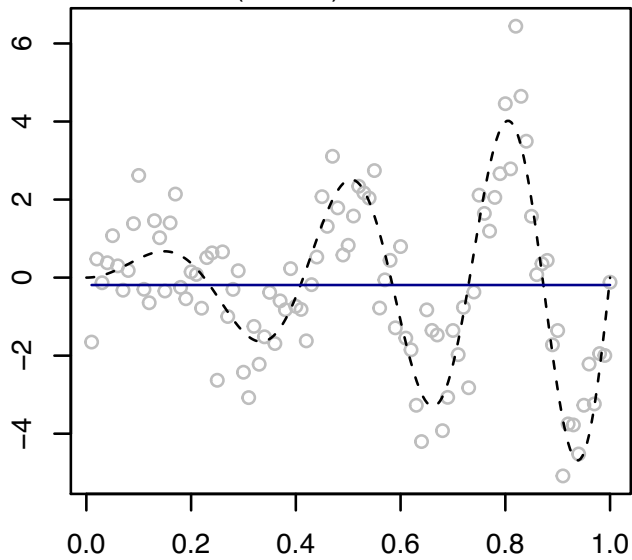
The following is known as a *generalized lasso* minimization problem:

$$\hat{\beta} = \operatorname{argmin}_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda \|\mathbf{D}\beta\|_1.$$

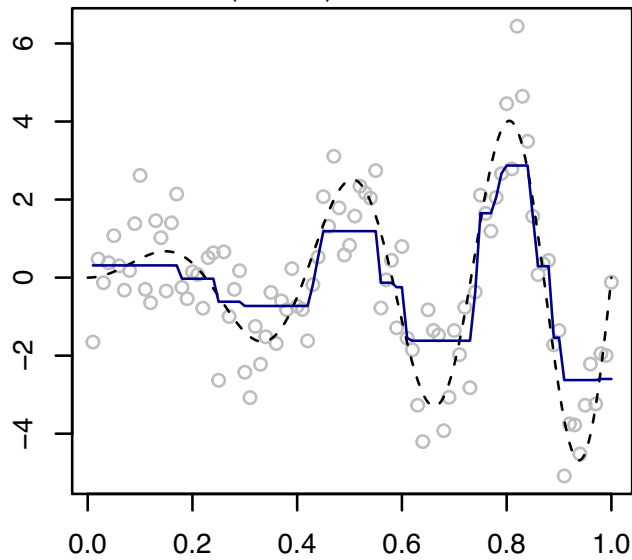
It can be solved with the `genlasso` package.

**Exercise:** Fit the trend filtering estimator on some data for  $k = 0, 1, 2, 3$  and plot.

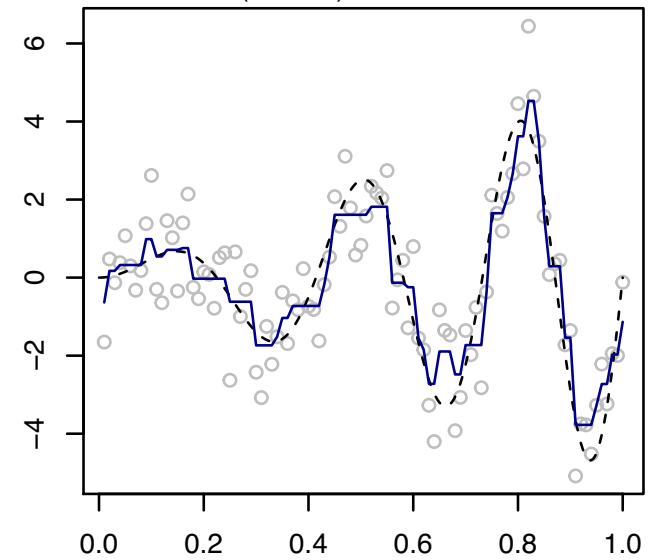
TF (order 0) with df = 1



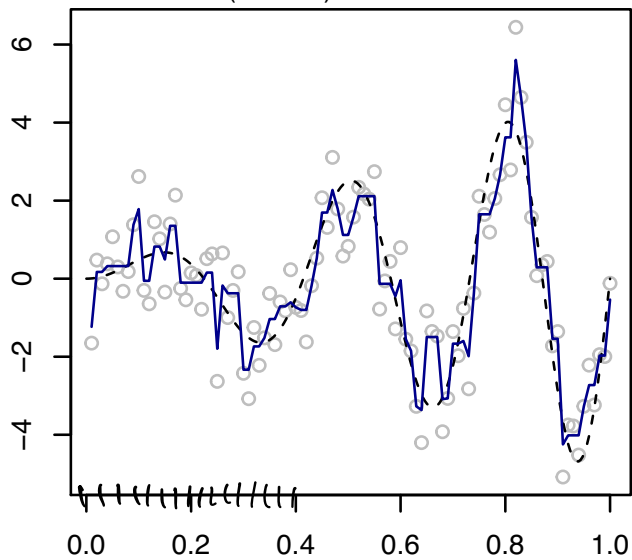
TF (order 0) with df = 21



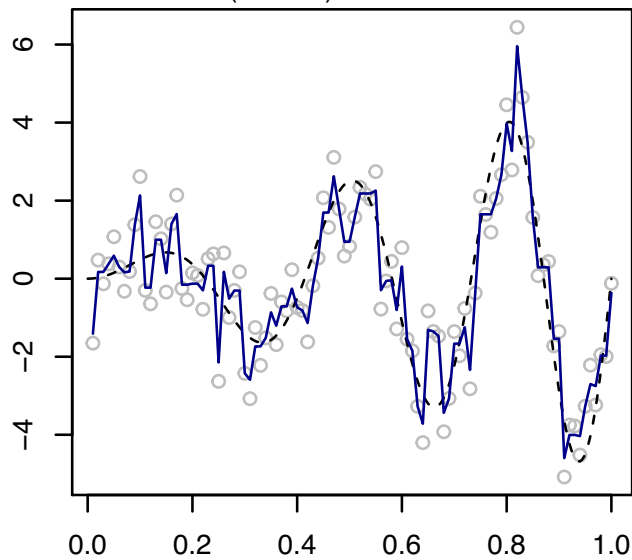
TF (order 0) with df = 40



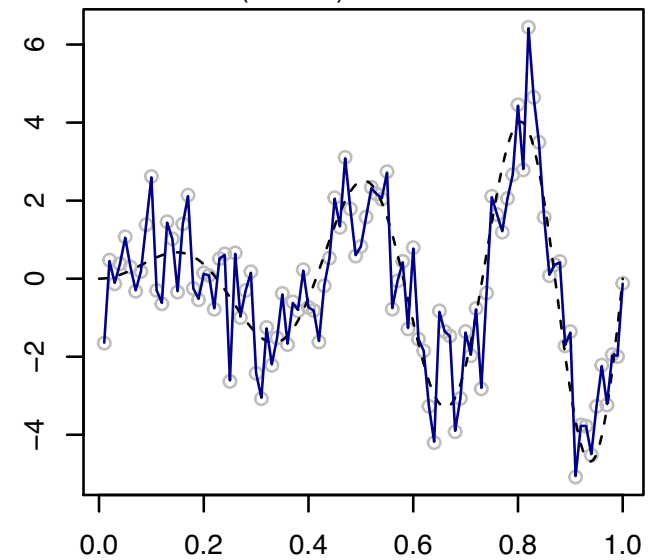
TF (order 0) with df = 60



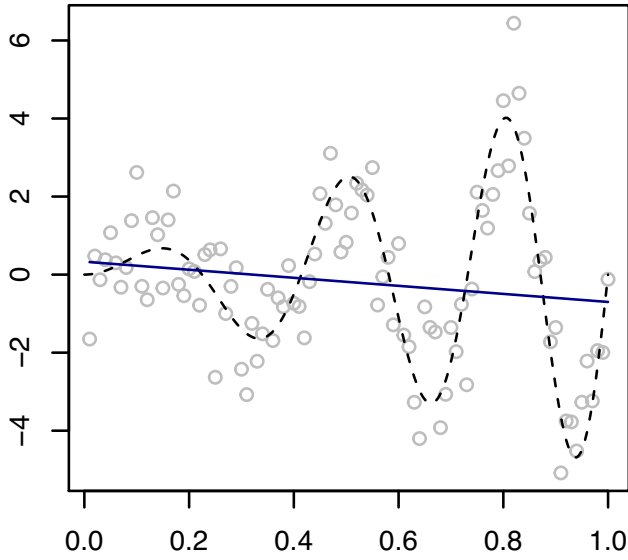
TF (order 0) with df = 79



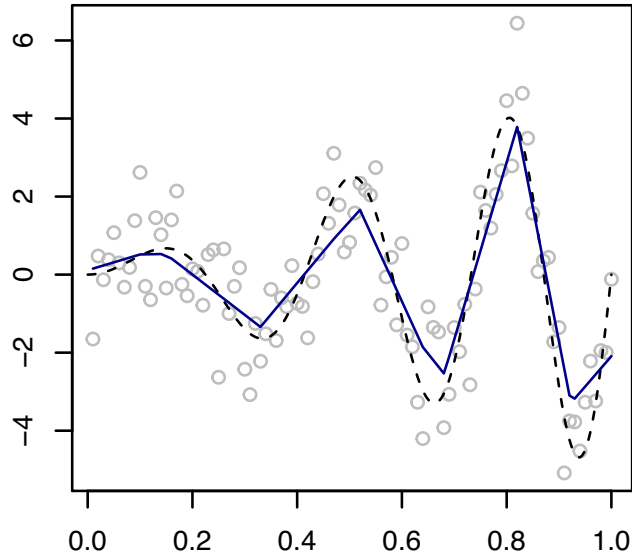
TF (order 0) with df = 99



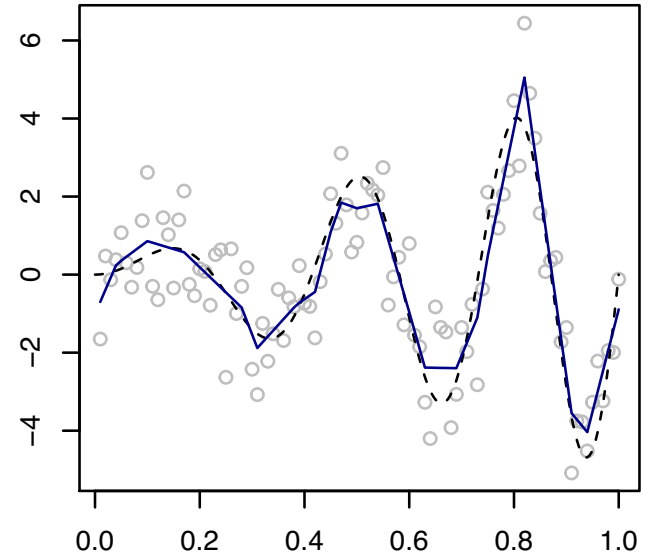
TF (order 1) with df = 2



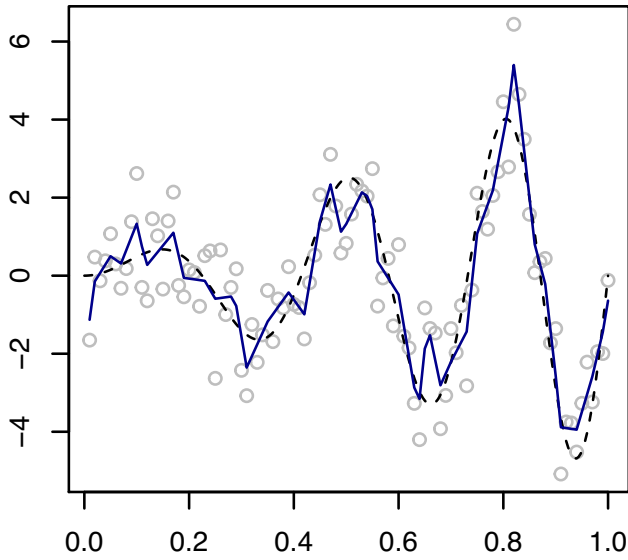
TF (order 1) with df = 15



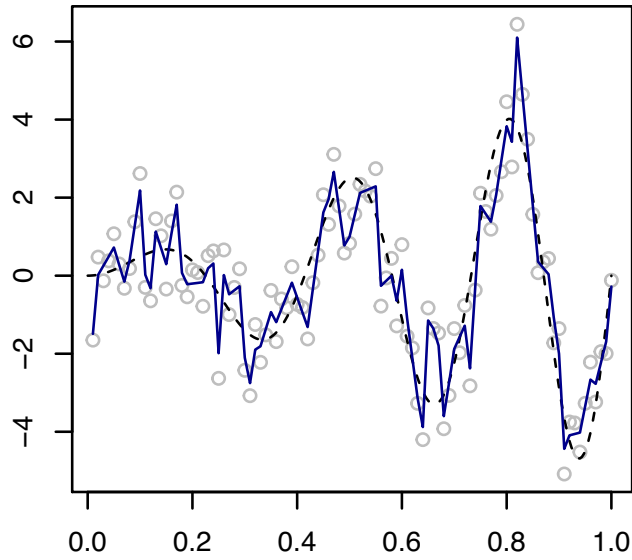
TF (order 1) with df = 23



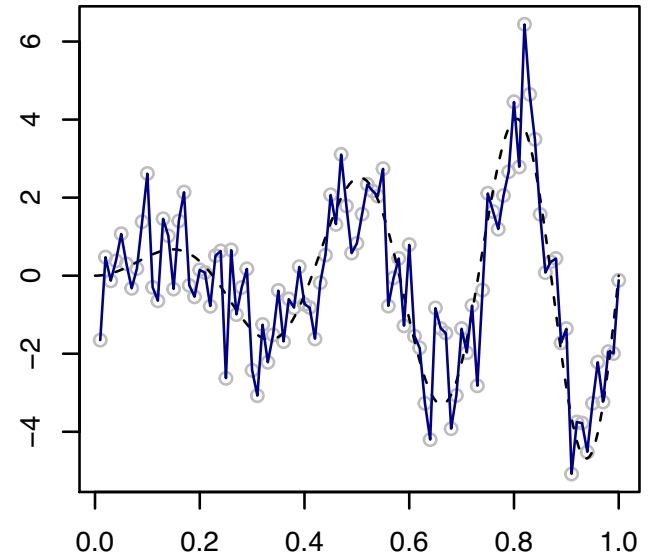
TF (order 1) with df = 46



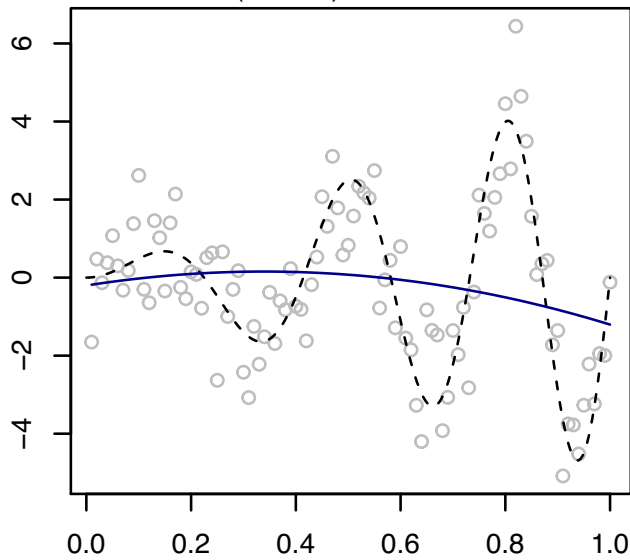
TF (order 1) with df = 68



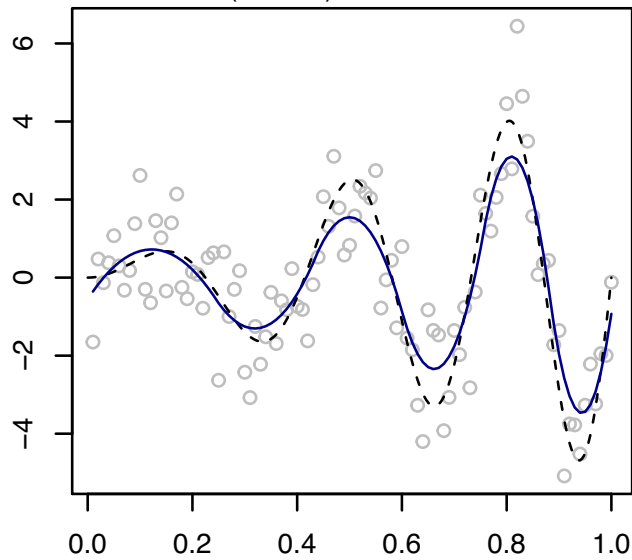
TF (order 1) with df = 99



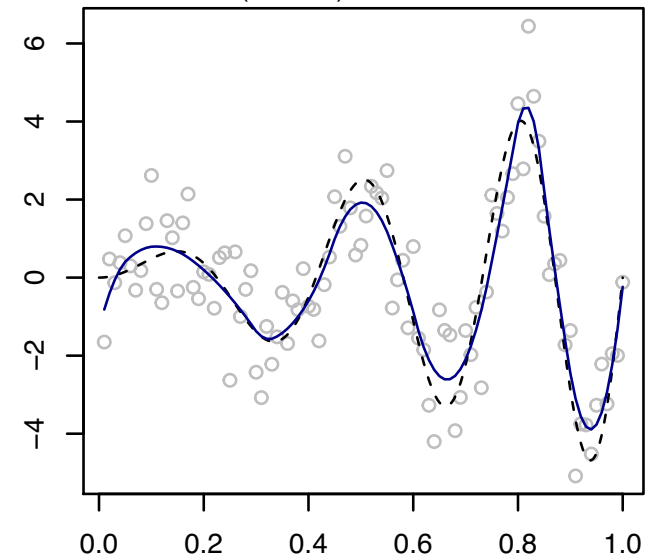
TF (order 2) with df = 3



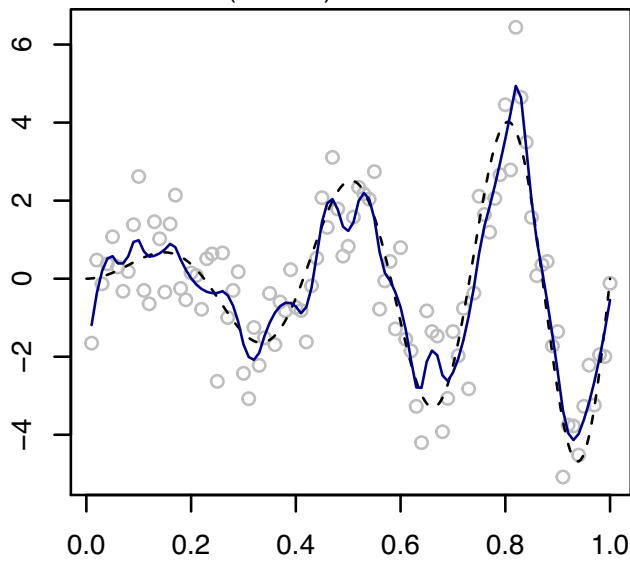
TF (order 2) with df = 12



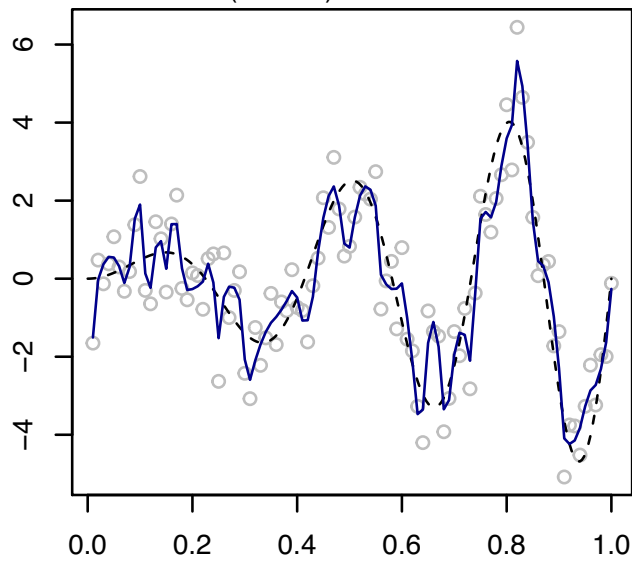
TF (order 2) with df = 19



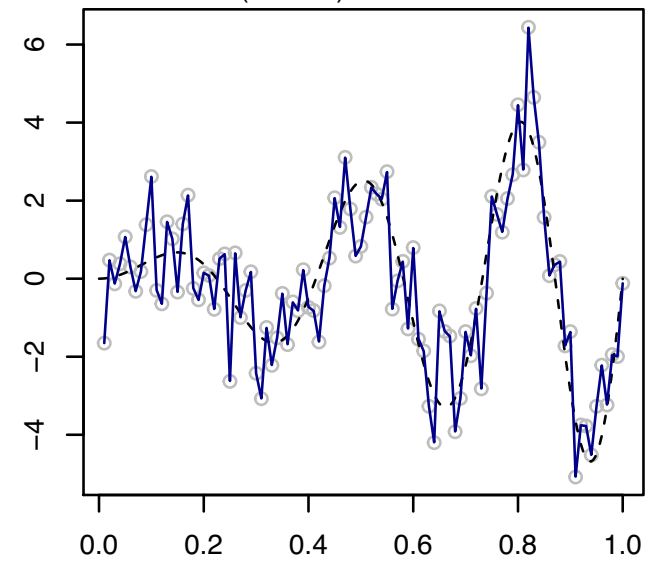
TF (order 2) with df = 39



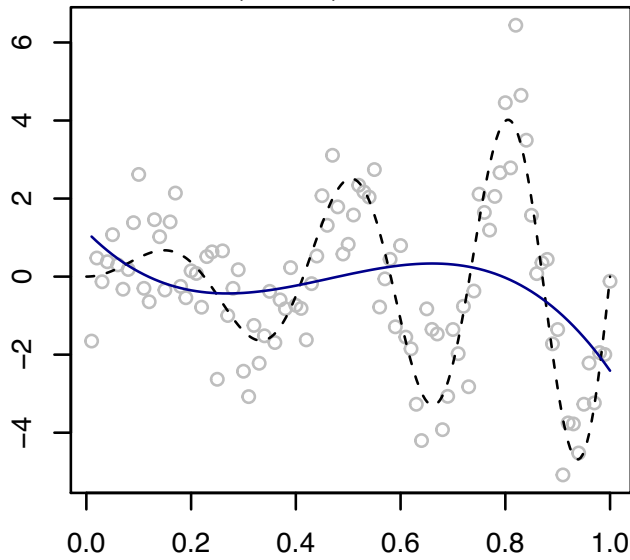
TF (order 2) with df = 64



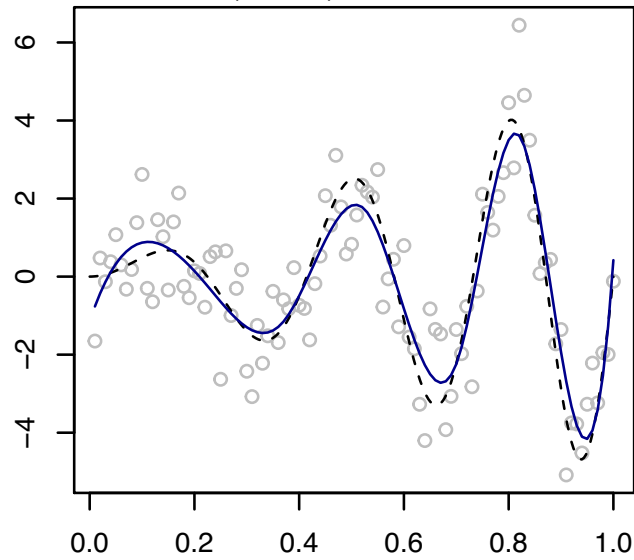
TF (order 2) with df = 99



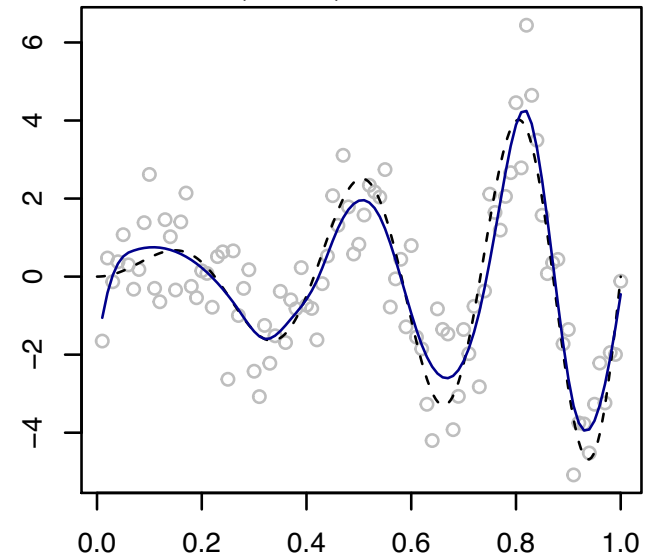
TF (order 4) with df = 4



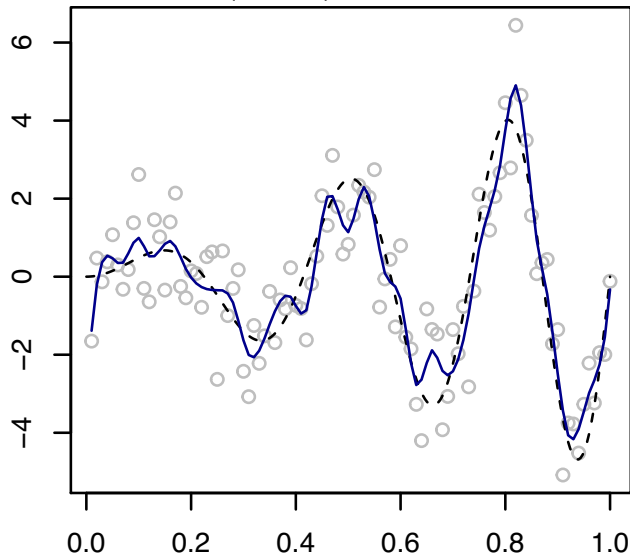
TF (order 4) with df = 12



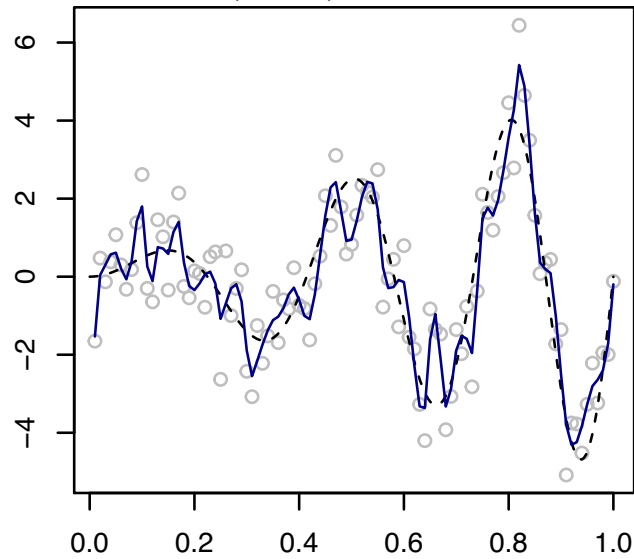
TF (order 4) with df = 18



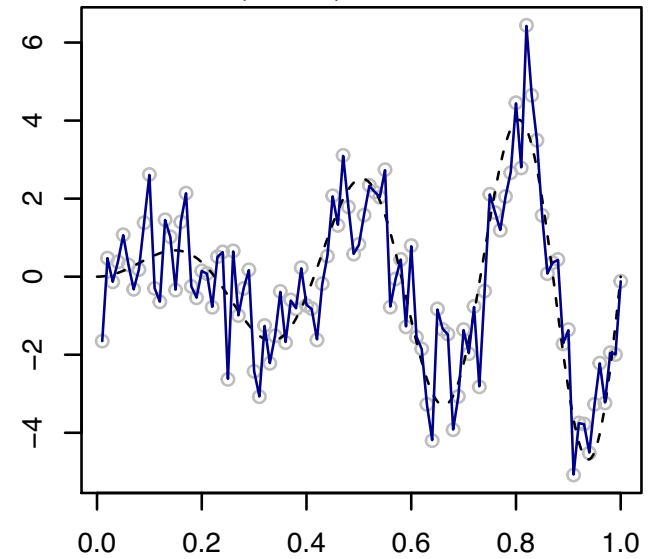
TF (order 4) with df = 35







TF (order 4) with df = 55



TF (order 4) with df = 99



-  David Ruppert, Matt P Wand, and Raymond J Carroll.  
*Semiparametric regression*.  
Number 12. Cambridge university press, 2003.
-  Bernard W Silverman.  
Spline smoothing: the equivalent variable kernel method.  
*The Annals of Statistics*, pages 898–916, 1984.
-  Ryan J Tibshirani et al.  
Adaptive piecewise polynomial estimation via trend filtering.  
*The Annals of statistics*, 42(1):285–323, 2014.
-  Grace Wahba.  
*Spline models for observational data*.  
SIAM, 1990.