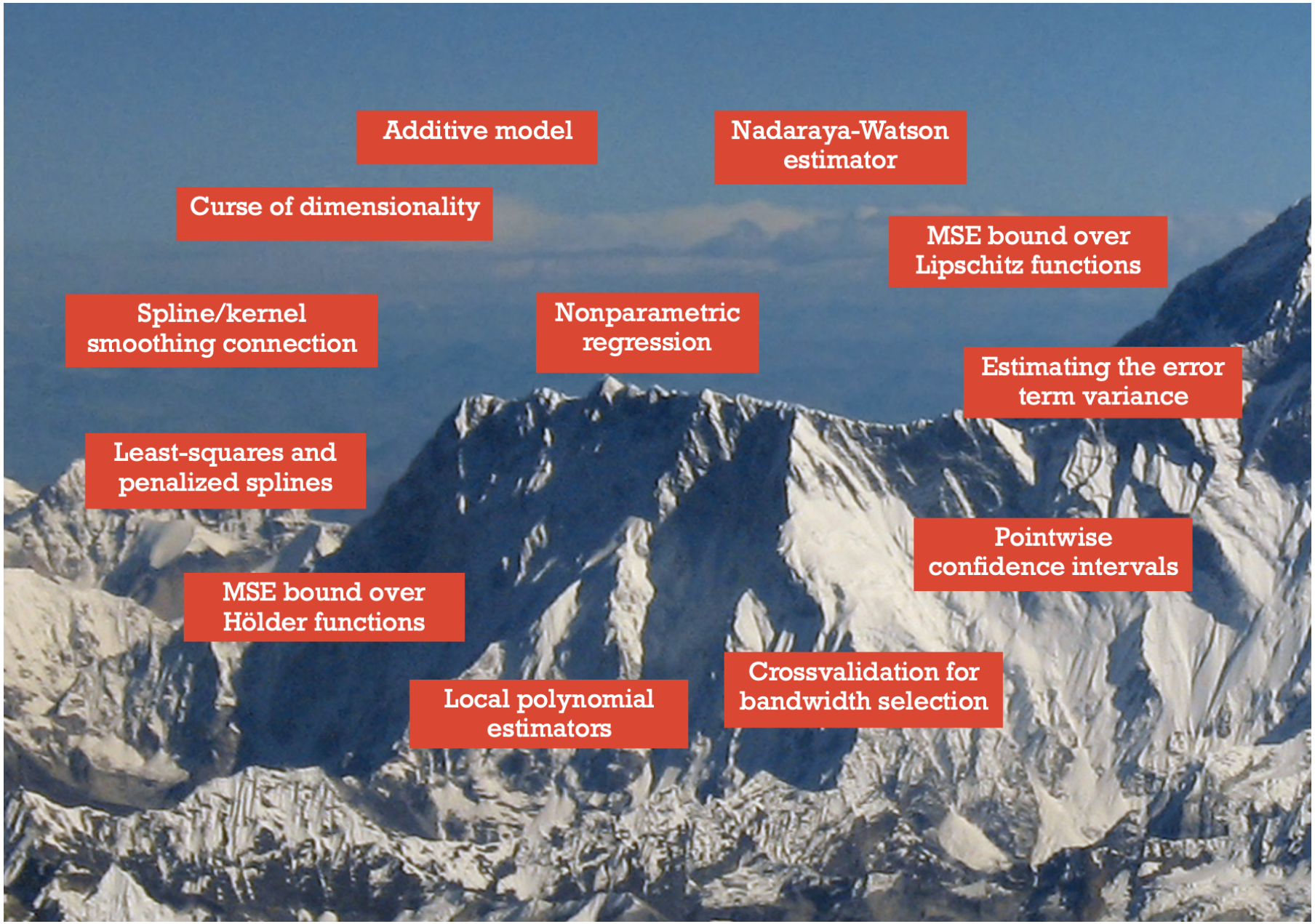
STAT 824 sp 2025 Lec 06 slides

Additive model for nonparametric multiple regression

Karl Gregory

University of South Carolina

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.


Additive model
Nadaraya-Watson estimator
Curse of dimensionality
MSE bound over Lipschitz functions
Spline/kernel smoothing connection
Nonparametric regression
Estimating the error term variance
Least-squares and penalized splines
Pointwise confidence intervals
MSE bound over Hölder functions
Crossvalidation for bandwidth selection
Local polynomial estimators

Observe $(\underset{\sim}{x}_1, y_1), \ldots, (\underset{\sim}{x}_n, y_n) \in [0,1]^d \times \mathbb{R}$

Assume

$$y_i = m(\underset{\sim}{x}_i) + \varepsilon_i, \quad i = 1, \ldots, n$$

$\varepsilon_1, \ldots, \varepsilon_n$ i.i.d error terms, $m : [0,1]^d \to \mathbb{R}$ unknown.

$$\hat{m}_n^{NW}(\underset{\sim}{x}) = \frac{\sum_{i=1}^n y_i \, K\left(h^{-1}(\underset{\sim}{x}_i - \underset{\sim}{x})\right)}{\sum_{i=1}^n K\left(h^{-1}(\underset{\sim}{x}_i - \underset{\sim}{x})\right)}$$

$K : \mathbb{R}^d \to \mathbb{R}$ Kernel function.

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

$$Y = m(X) + \varepsilon, \qquad \text{for some } m : [0,1]^p \to \mathbb{R},$$

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

Multivariate Nadaraya-Watson estimator

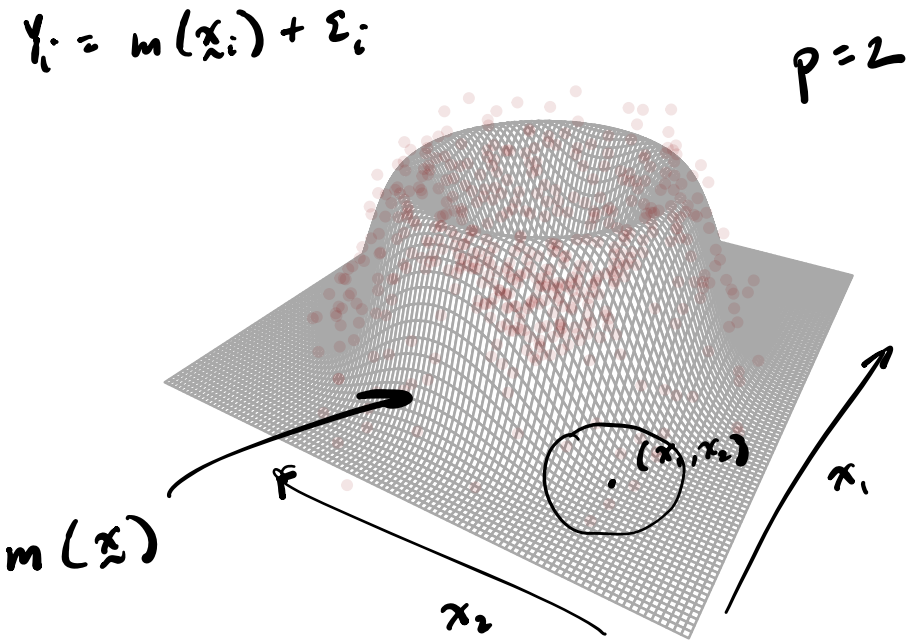
A multivariate version of the Nadaraya-Watson estimator is given by

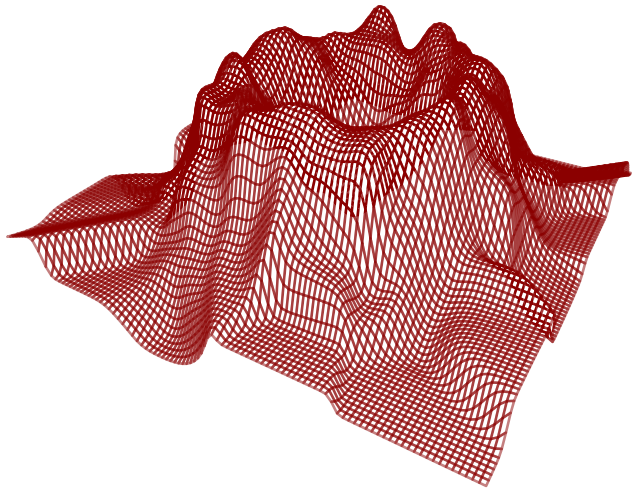
$$\hat{m}_n^{\mathrm{NW}}(x) = \frac{\sum_{i=1}^n Y_i K(h^{-1}(X_i - x))}{\sum_{j=1}^n K(h^{-1}(X_j - x))} \qquad \text{for all } x \in [0,1]^p,$$

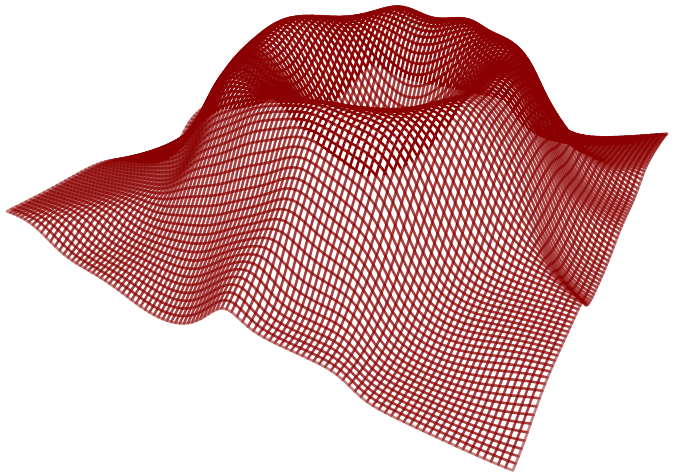
for some kernel function $K : \mathbb{R}^p \to \mathbb{R}$ and bandwidth $h > 0$.

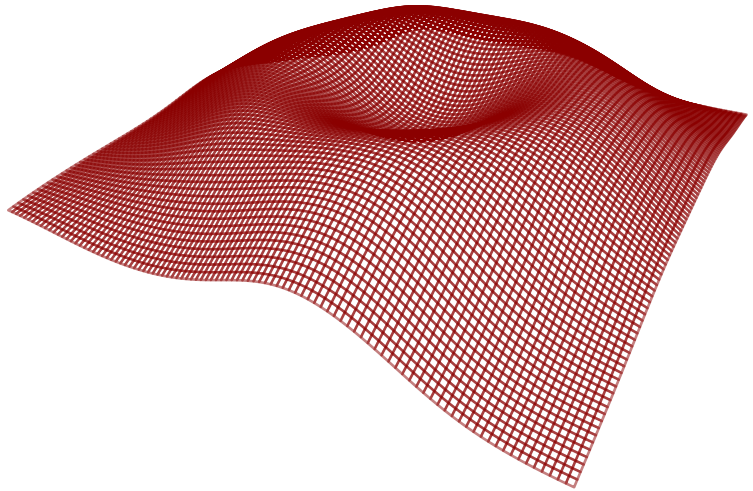
Kernel often like $K(u) = \prod_{j=1}^p G(u_j)$ where G a univariate kernel like

$$G(z) = \phi(z) \quad \text{or} \quad G(z) = \frac{3}{4}(1 - z^2)\mathbf{1}(|z| \le 1).$$


$Y_i = m(\underset{\sim}{x}_i) + \varepsilon_i$
$p = 2$
$m(\underset{\sim}{x})$
x_2
x_1
(x_1, x_2)







Regard $x_1, \ldots, x_n$ as deterministic (fixed)

$$\mathrm{Var}\ \hat{m}_n^{NW}(x) = \mathrm{Var}\left(\frac{\sum_{i=1}^n Y_i K\left(h^{-1}(x_i - x)\right)}{\sum_{i=1}^n K\left(h^{-1}(x_i - x)\right)} \right)$$

$$= \sigma^2 \sum_{i=1}^n \left(\frac{K\left(h^{-1}(x_i - x)\right)}{\sum_{i'=1}^n K\left(h^{-1}(x_{i'} - x)\right)} \right)^2$$

$$= \sigma^2 \frac{\sum_{i=1}^n K^2\left(h^{-1}(x_i - x)\right)}{\left(\sum_{i=1}^n K\left(h^{-1}(x_i - x)\right)\right)^2}$$

$$= \frac{\sigma^2}{nh^p} \frac{\frac{1}{nh^p}\sum_{i=1}^n K^2\left(h^{-1}(x_i - x)\right)}{\left(\frac{1}{nh^p}\sum_{i=1}^n K\left(h^{-1}(x_i - x)\right)\right)^2}$$

$\hat{f}_X(x)$

bounded

Consider the variance $\text{Var}\, \hat{m}_n^{\text{NW}}(x_0)$. We make the following assumptions.

(K1) Let $K(u) \leq K_{\max} < \infty \; \forall \; u \in \mathbb{R}^p$.

(D1) Let $X_1, \ldots, X_n \in [0,1]^p$ be deterministic such that for some $n_0 > 0$

$$0 < c_1 \leq \frac{1}{nh^p} \sum_{i=1}^n K(h^{-1}(X_i - x)) \leq c_1^{-1}$$
$$0 < c_2 \leq \frac{1}{nh^p} \sum_{i=1}^n K^2(h^{-1}(X_i - x)) \leq c_2^{-1}$$

for some c_1, c_2, for all $x \in [0,1]^p$, for all $n \geq n_0$.

Bounds on $\text{Var}\, \hat{m}_n^{\text{NW}}(x_0)$

Under (K1) and (D1), for all $n \geq n_0$, we have

the curse of dimensionality.

$$\text{Var}\, \hat{m}_n^{\text{NW}}(x_0) \in \left(\frac{\sigma^2}{nh^p} \cdot c_1^2 c_2, \; \frac{\sigma^2}{nh^p} \cdot \frac{1}{c_1^2 c_2} \right) \quad \text{for all} \quad x_0 \in [0,1]^p.$$

Exercise: Prove the above and interpret.

Local linear approx around $\underset{\sim}{x}_0$:

$$m(\underset{\sim}{x}) \approx \underbrace{m(\underset{\sim}{x}_0)}_{\theta_0} + \underbrace{[\nabla m(\underset{\sim}{x}_0)]^T}_{\underset{\sim}{\theta}_1}(\underset{\sim}{x} - \underset{\sim}{x}_0)$$

We could also consider local polynomial estimators in the multivariate setting.

Local linear multivariate regression estimator

A multivariate local linear estimator $\hat{m}_n^{\text{LP-1}}(x)$ of $m(x)$ is given by $\hat{\theta}_0(x)$, where

$$(\hat{\theta}_0, \hat{\boldsymbol{\theta}}_1)(x) = \underset{\theta_0 \in \mathbb{R}, \boldsymbol{\theta} \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} (Y_i - \theta_0 - \boldsymbol{\theta}_1^T (X_i - x))^2 K(h^{-1}(X_i - x)).$$

This is also subject to the curse of dimensionality.

Additivity Assumption: $\underset{\sim}{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}$

$$m(\underset{\sim}{x}) = m_1(x_1) + \cdots + m_p(x_p),$$

$m_j : [0,1] \to \mathbb{R}$ each j

Observe $(\underset{\sim}{x}_1, Y_1), \ldots, (\underset{\sim}{x}_n, Y_n)$ $\quad \underset{\sim}{x}_i = \begin{bmatrix} x_{i1} \\ \vdots \\ x_{ip} \end{bmatrix}$

$$Y_i = \mu + m_1(x_{i1}) + \cdots + m_p(x_{ip}) + \varepsilon_i$$

Identifiability issues?

$$\mathbb{E}\, Y_i = \mu + m_1(x_{i1}) + \cdots + m_p(x_{ip})$$

$$= \mu + (m_1(x_{i1}) + c) + \cdots + (m_p(x_{ip}) - c)$$

Impose Identifiability condition:

Require $\frac{1}{n} \sum_{i=1}^{n} m_j(x_{ij}) = 0$ $\quad \left(\mathbb{E}\, m_j(X_j) = 0 \right)$ $j = 1, \ldots, p.$

Then $\bar{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(\mu + \sum_{j=1}^{p} m_j(x_{ij}) + \varepsilon_i\right)$$

$$= \mu + \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i \; .$$

So we estimate μ with $\bar{Y}_n$ ...

But from now on, just center $Y_1, \ldots, Y_n$ and remove μ from model.

Model becomes

$$Y_i = m_1(x_{i1}) + \cdots + m_p(x_{ip}) + \varepsilon_i$$

m_j centered: $\frac{1}{n}\sum_{i=1}^{n} m_j(x_{ij}) = 0 \quad \forall\, j$

A way to mitigate the curse of dimensionality is by assuming *additivity*.

The *additivity* assumption is that $m : [0,1]^p \to \mathbb{R}$ may be written

$$m(x) = m_1(x_1) + \cdots + m_p(x_p)$$

for all $x \in [0,1]^p$ for some functions $m_1, \ldots, m_p : [0,1] \to \mathbb{R}$.

Stone (1985) argued additive functions sufficient for many applications [5].

Additive model

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

$$Y = \mu + m_1(X_1) + \cdots + m_p(X_p) + \varepsilon, \qquad \text{for some } m_1, \ldots, m_p : [0,1] \to \mathbb{R},$$

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

Discuss: Is the additive model identifiable? Examples.

Identifiability condition for additive model

For the sake of identifiability we will assume, without loss of generality, that

$$\mathbb{E} m_j(X_j) = 0 \quad \text{for each} \quad j = 1, \ldots, p.$$

We will make our estimators satisfy the identifiability condition empirically, i.e.

$$n^{-1} \sum_{i=1}^{n} \hat{m}_j(X_{ij}) = 0 \text{ for } j = 1, \ldots, p.$$

for any estimators $\hat{m}_1, \ldots, \hat{m}_p$.

We will always estimate μ with $\bar{Y}_n$.

From now on, assume $\mu = 0$ and that $Y_1, \ldots, Y_n$ are centered, so our model is just

$$Y = m_1(X_1) + \cdots + m_p(X_p) + \varepsilon.$$

Exercise: Let $\mathbf{B} = [\mathbf{B}_1, \mathbf{B}_{-1}]$ have full column rank and let

$$\mathbf{P}_{-1} = \mathbf{B}_{-1}(\mathbf{B}_{-1}^T\mathbf{B}_{-1})^{-1}\mathbf{B}_{-1} \quad \text{and} \quad \mathbf{B}_{1\setminus -1} = (\mathbf{I} - \mathbf{P}_{-1})\mathbf{B}_1.$$

1. Show that
$$\begin{bmatrix} \mathbf{B}_1^T\mathbf{B}_1 & \mathbf{B}_1^T\mathbf{B}_{-1} \\ \mathbf{B}_{-1}^T\mathbf{B}_1 & \mathbf{B}_{-1}^T\mathbf{B}_{-1} \end{bmatrix} \begin{bmatrix} \hat{\alpha}_1 \\ \hat{\alpha}_{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1^T\mathbf{Y} \\ \mathbf{B}_{-1}^T\mathbf{Y} \end{bmatrix}$$
if and only if $\hat{\alpha}_1 = (\mathbf{B}_{1\setminus -1}^T\mathbf{B}_{1\setminus -1})^{-1}\mathbf{B}_{1\setminus -1}\mathbf{Y}$.
2. If $\text{Cov}(\mathbf{Y}) = \sigma^2\mathbf{I}$, give $\text{Var}(\mathbf{b}_1^T\hat{\alpha}_1)$.

Discuss: The purpose of this exercise.

How to estimate $m_1, \dots, m_p$ in additive model.

Backfitting: For the moment assume we have

$$(X_1, \dots, X_p, Y) \in [0,1]^p \times \mathbb{R}$$

and assume

$$Y = m_1(X_1) + \dots + m_p(X_p) + \varepsilon,$$

ε independent of $X_1, \dots, X_p$.

Write

$$m_j(X_j) = Y - \sum_{k \neq j} m_k(X_k) - \varepsilon$$

Take $\mathbb{E}[\,\cdot \mid X_j]$ of both sides. Then

$$(\bigstar) \qquad m_j(X_j) = \mathbb{E}[Y \mid X_j] - \sum_{k \neq j} \mathbb{E}[m_k(X_k) \mid X_j]$$

for all $j = 1, \dots, p$.

Let Π_j denote the operator $\mathbb{E}[\,\cdot \mid X_j]$

Let $m_j := m_j(X_j)$.

Then we can represent (✱) as

$$\begin{bmatrix} I & \Pi_1 & \cdots & \Pi_1 \\ \Pi_2 & I & \cdots & \Pi_2 \\ \vdots & \vdots & \ddots & \vdots \\ \Pi_p & \Pi_p & \cdots & I \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_p \end{bmatrix} = \begin{bmatrix} \Pi_1 Y \\ \Pi_2 Y \\ \vdots \\ \Pi_p Y \end{bmatrix},$$

← Backfitting solves an empirical version of this.

where I gives $I m_j = m_j$, so

$$m_1 + \sum_{j=2}^{p} \Pi_1 m_j = \Pi_1 Y$$

$$m_2 + \sum_{j \neq 2} \Pi_2 m_j = \Pi_2 Y$$

$$\vdots$$

$$m_p + \sum_{j \neq p} \Pi_p m_j = \Pi_p Y$$

Let $S_1, \ldots, S_p$ be $n \times n$ smoother matrices

I_n be $n \times n$ identity

$\hat{\underset{\sim}{m}}_1, \ldots, \hat{\underset{\sim}{m}}_p$ are $n \times 1$ vectors such that

$$\underset{\sim}{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} \qquad \hat{\underset{\sim}{m}}_j = \begin{bmatrix} \hat{m}_j(x_{1j}) \\ \vdots \\ \hat{m}_j(x_{nj}) \end{bmatrix}$$

estimated additive components at design points.

With

$$\begin{bmatrix} I_n & S_1 & \cdots & S_1 \\ S_2 & I_n & \cdots & S_2 \\ \vdots & \vdots & \ddots & \vdots \\ S_p & S_p & \cdots & I_n \end{bmatrix} \begin{bmatrix} \hat{m}_1 \\ \hat{m}_2 \\ \vdots \\ \hat{m}_p \end{bmatrix} = \begin{bmatrix} S_1 y \\ S_2 y \\ \vdots \\ S_p y \end{bmatrix}.$$

Backfitting Algorithm (Gauss-Seidel):

Initialize $\hat{m}_1 = 0, \ldots, \hat{m}_p = 0$.

For $j = 1, \ldots, p$ update

$$\hat{m}_j \leftarrow S_j \left(y - \sum_{k \neq j} \hat{m}_k \right)$$

$$\hat{m}_j \leftarrow \hat{m}_j - \frac{1}{n} 1^T \hat{m}_j \quad \text{(impose identifiability condition)}$$

Until convergence.

Backfitting is a fast and simple way to compute estimators in the additive model.

It also spares us from the numerical issues encountered in the last few slides.

The components of the model $Y = \sum_{j=1}^{p} m_j(X_j) + \varepsilon$ have the interpretation

$$m_j(X_j) = \mathbb{E}[Y|X_j] - \sum_{k \neq j} \mathbb{E}[m_k(X_k)|X_j]$$

for $j = 1, \ldots, p$.

Now, letting Π_j represent conditional expectation given X_j, we have

$$m_j = \Pi_j Y - \sum_{k \neq j} \Pi_j m_k \text{ for } j = 1, \ldots, p \quad (\text{with } m_k := m_k(X_k)).$$

We can write this system of equations as

$$\begin{bmatrix} I & \Pi_1 & \ldots & \Pi_1 \\ \Pi_2 & I & \ldots & \Pi_2 \\ \vdots & \vdots & \ddots & \vdots \\ \Pi_p & \Pi_p & \ldots & I \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_p \end{bmatrix} = \begin{bmatrix} \Pi_1 Y \\ \Pi_2 Y \\ \vdots \\ \Pi_p Y \end{bmatrix}.$$

The *backfitting algorithm* solves an empirical version

$$\begin{bmatrix} \mathbf{I}_n & \mathbf{S}_1 & \dots & \mathbf{S}_1 \\ \mathbf{S}_2 & \mathbf{I}_n & \dots & \mathbf{S}_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_p & \mathbf{S}_p & \dots & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} \hat{\mathbf{m}}_1 \\ \hat{\mathbf{m}}_2 \\ \vdots \\ \hat{\mathbf{m}}_p \end{bmatrix} = \begin{bmatrix} \mathbf{S}_1\mathbf{Y} \\ \mathbf{S}_2\mathbf{Y} \\ \vdots \\ \mathbf{S}_p\mathbf{Y} \end{bmatrix}.$$

of the system of equations on the previous slide, where

- $\mathbf{S}_1, \dots, \mathbf{S}_p$ are smoother matrices associated with univariate smoothers.
- $\hat{\mathbf{m}}_1, \dots, \hat{\mathbf{m}}_p$ are $n \times 1$ with evaluations of estimators at design points.
- $\mathbf{Y} = (Y_1, \dots, Y_n)^T$.

Backfitting algorithm (Gauss–Seidel). See Buja et al. (1989), [1].

Initialize: $\hat{\mathbf{m}}_1, \dots, \hat{\mathbf{m}}_p = \mathbf{0}$. Then iterate: For $j = 1, \dots, p$

1. $\hat{\mathbf{m}}_j \leftarrow \mathbf{S}_j(\mathbf{Y} - \sum_{k \neq j} \hat{\mathbf{m}}_k)$
2. $\hat{\mathbf{m}}_j \leftarrow \hat{\mathbf{m}}_j - n^{-1}\mathbf{1}_n^T \hat{\mathbf{m}}_j$ (centering step for identifiability)

until $\hat{\mathbf{m}}_1, \dots, \hat{\mathbf{m}}_p$ no longer change.

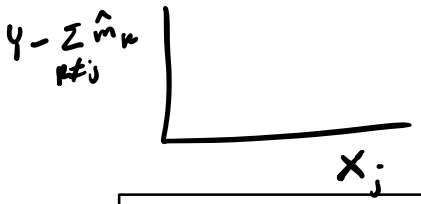
To implement the backfitting algorithm, we just need the smoother matrices.

- Least-squares splines: $\mathbf{S}_j = \mathbf{B}_{nj}(\mathbf{B}_{nj}^T\mathbf{B}_{nj})^{-1}\mathbf{B}_{nj}^T$
- Penalized splines: $\mathbf{S}_j = \mathbf{B}_{nj}(\mathbf{B}_{nj}^T\mathbf{B}_{nj} + \lambda\mathbf{\Omega}_j)^{-1}\mathbf{B}_{nj}^T$

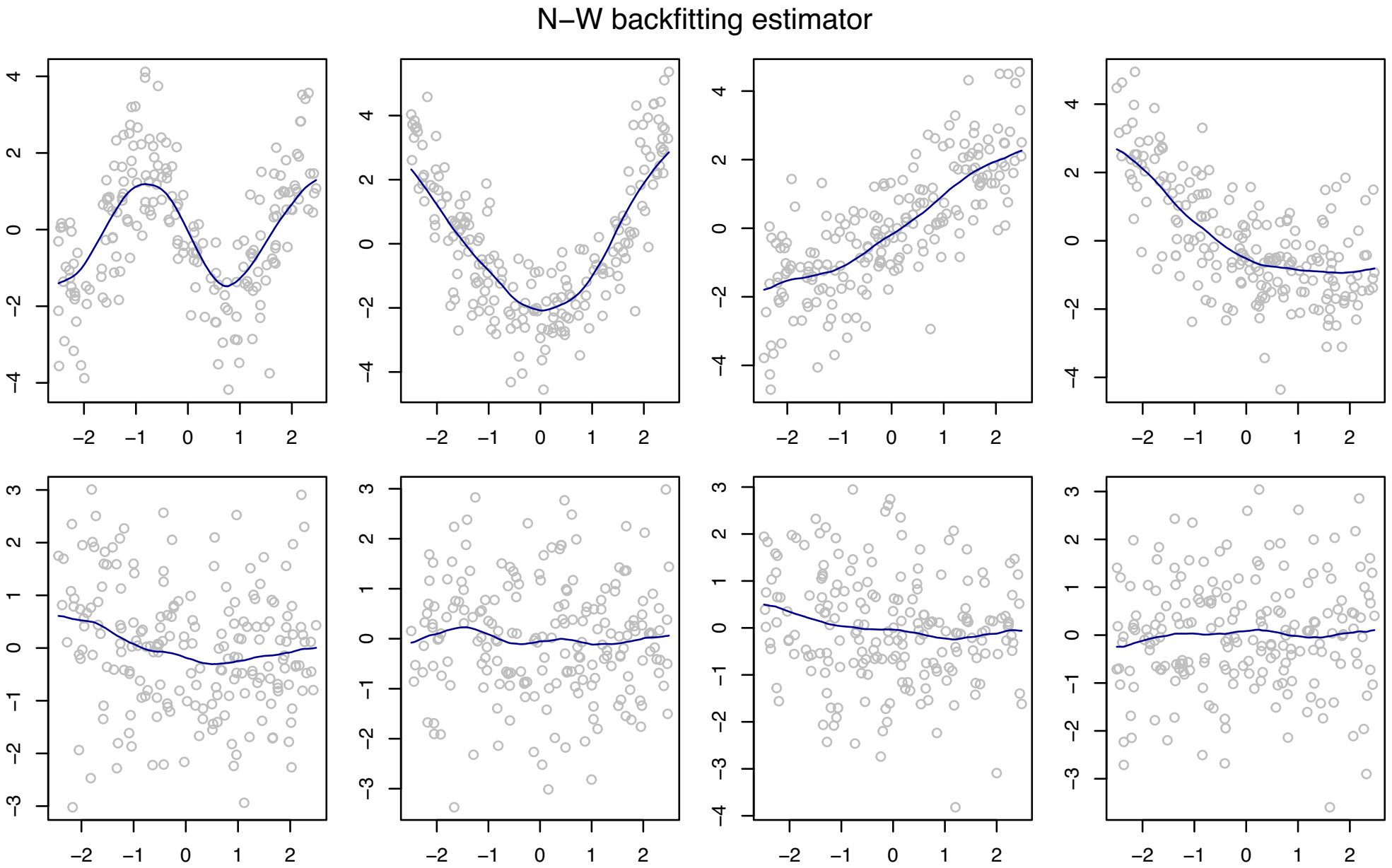
  (Note that there is no need to center the basis functions.)
- Nadaraya-Watson: $\mathbf{S}_j = \left(\frac{K(h^{-1}(X_{kj} - X_{ij}))}{\sum_{\ell=1}^{n} K(h^{-1}(X_{\ell j} - X_{ij}))} \right)_{1 \leq i \leq n, 1 \leq k \leq n}$

Exercise: Demonstrate backfitting with penalized splines.

outer(X[,j], X[,j], "-") = $\left(X_{ij} - X_{kj} \right)_{1 \leq i \leq n, 1 \leq k \leq n}$



Penalized splines backfitting estimator


N–W backfitting estimator

$$X = \begin{bmatrix} \underset{\sim}{x}_1 & \cdots & \underset{\sim}{x}_p \end{bmatrix}$$

$$S_j = \underset{\sim}{x}_j (\underset{\sim}{x}_j^T \underset{\sim}{x}_j)^{-1} \underset{\sim}{x}_j^T \qquad \underset{\sim}{x}$$

Exercise: Try backfitting for multiple linear regression,

$$Y = \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon,$$

$m_1(x_1)$ $\quad$ $m_p(x_p)$

with $\mathbb{E}X_j = 0$ for $j = 1, \ldots, p$.

projections onto $\operatorname{span}\{\underset{\sim}{x}_j\}$

1. What are the smoother matrices $\mathbf{S}_1, \ldots, \mathbf{S}_p$ in this context?
2. How do you obtain the least-squares estimators $\hat{\beta}_1, \ldots, \hat{\beta}_p$ from the output?
3. Implement on some made-up data.

First column of $\underset{\sim}{x}$

$$\hat{\underset{\sim}{m}}_1 = \begin{bmatrix} \hat{m}_1(x_{11}) \\ \vdots \\ \hat{m}_1(x_{n1}) \end{bmatrix} = \hat{\beta}_1 \begin{bmatrix} x_{11} \\ \vdots \\ x_{n1} \end{bmatrix} = \hat{\beta}_1 \underset{\sim}{x}_1$$

$$\hat{\underset{\sim}{m}}_j = S_j \left(\underset{\sim}{y} - \sum_{k \neq j} \hat{\underset{\sim}{m}}_k \right)$$

$$= \underset{\sim}{x}_j (\underset{\sim}{x}_j^T \underset{\sim}{x}_j)^{-1} \underset{\sim}{x}_j^T \left(\underset{\sim}{y} - \sum_{k \neq j} \hat{\beta}_k \underset{\sim}{x}_k \right)$$

$$\hat{\beta}_1 = (\underset{\sim}{x}_1^T \underset{\sim}{x}_1)^{-1} \underset{\sim}{x}_1^T \hat{\underset{\sim}{m}}_1$$

Without Backfitting: Consider splines.

For each j, use basis functions $b_{j1} \cdots b_{jd}$.

Make design matrix

$$\underset{n \times d}{B_j} = \Big(b_{jk}(x_{ij}) \Big)_{1 \le i \le n,\ 1 \le k \le d} \qquad j = 1, \dots, p$$

To impose identifiability condition, empirically center each basis function

$$\bar{b}_{jk}(x) = b_{jk}(x) - \frac{1}{n} \sum_{i=1}^{n} b_{jk}(x_{ij})$$

Make $\bar{B}_j = \big(\bar{b}_{jk}(x_{ij}) \big)_{1 \le i \le n,\ 1 \le k \le d}$

Then consider

$$\left(\hat{m}_1, \dots, \hat{m}_p \right) = \underset{g_1 \in M_1, \dots, g_p \in M_p}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} g_j(x_{ij}) \right)^2,$$

where M_j is all functions of the form $g_j(x) = \sum_{k=1}^{d} \alpha_k \bar{b}_{jk}(x)$

Find LS coefficients:

$$\left(\hat{\alpha}_1, \dots, \hat{\alpha}_p \right) = \underset{\alpha_1, \dots, \alpha_p \in \mathbb{R}^d}{\operatorname{argmin}} \left\| \underset{n \times 1}{Y} - \sum_{j=1}^{p} \underset{n \times d}{\bar{B}_j} \underset{d \times 1}{\alpha_j} \right\|^2$$

$$= \underset{\underset{\sim}{\alpha}_1, \ldots, \underset{\sim}{\alpha}_p \in \mathbb{R}^d}{\operatorname{argmin}} \left\| \underset{\sim}{Y} - \overbrace{\begin{bmatrix} \bar{B}_1 & \cdots & \bar{B}_p \end{bmatrix}}^{\bar{B}} \begin{bmatrix} \underset{\sim}{\alpha}_1 \\ \vdots \\ \underset{\sim}{\alpha}_p \end{bmatrix} \right\|^2$$

$n \times pd$

$pd \times 1$

Can get

$$\begin{pmatrix} \hat{\alpha}_1 \\ \vdots \\ \hat{\alpha}_p \end{pmatrix} = \left(\bar{B}^T \bar{B} \right)^{-1} \bar{B}^T \underset{\sim}{Y}$$

Problem:

$$\bar{B}_j = \overbrace{\left(I_n - \frac{1}{n} \underset{\sim}{1}_n \underset{\sim}{1}_n^T \right)}^{\text{centering}} B_j$$

$\bar{B}_j$: $n \times d$

B_j: $n \times d$ — Full Rank

rank $d-1$

if B_j comes from B-splines

then each row of B_j sums to 1.

$\Rightarrow \left(\bar{B}_j^T \bar{B}_j \right)$ is not invertible.

Silly solution: Just remove one basis function from B_j.

Sparse

$$\left(\hat{\underset{\sim}{\alpha}}_1, \ldots, \hat{\underset{\sim}{\alpha}}_p \right) = \underset{\underset{\sim}{\alpha}_1, \ldots, \underset{\sim}{\alpha}_p \in \mathbb{R}^d}{\operatorname{argmin}} \left\| \underset{\sim}{Y} - \sum_{j=1}^{p} \underset{n \times d}{\overline{B}_j} \underset{d \times 1}{\underset{\sim}{\alpha}_j} \right\|^2 + \lambda \sum_{j=1}^{p} \| \underset{\sim}{\alpha}_j \|_2$$

There are **multitudes** of ways to estimate $m_1, \ldots, m_p$. One is this:

A least-squares splines estimator for the additive model

Least-squares spline estimators $\hat{m}_1^{\text{spl}}, \ldots, \hat{m}_p^{\text{spl}}$ of $m_1, \ldots, m_p$ may be defined as

$$\left(\hat{m}_1^{\text{spl}}, \ldots, \hat{m}_p^{\text{spl}}\right) = \underset{g_j \in \bar{\mathcal{M}}_{nj}}{\text{argmin}} \sum_{i=1}^n \left[Y_i - \sum_{j=1}^p g_j(X_{ij})\right]^2,$$

where for $j = 1, \ldots, p$,

$$\bar{\mathcal{M}}_{nj} = \text{span}\{\bar{b}_{j1}, \ldots, \bar{b}_{jd}\}, \text{ with } \bar{b}_{jl}(x) = b_{jl}(x) - n^{-1} \sum_{i=1}^n b_{jl}(X_{ij}),$$

for $l = 1, \ldots, d$, where $b_{j1}, \ldots, b_{jd}$ are cubic B-spline basis functions.

I have defaulted to cubic splines (we can use splines of other orders).

Exercise:

1. Verify that each $\hat{m}_j^{\text{spl}}$ will satisfy $n^{-1} \sum_{i=1}^n \hat{m}_j^{\text{spl}}(X_{ij}) = 0$.
2. Write the objective function in matrices. Give normal equations.
3. Check whether $\bar{\mathbf{B}}_{jn} = \left(\bar{b}_{j\ell}(X_{ij})\right)_{1 \le i \le n, 1 \le \ell \le d}$ has full rank.

For $\bar{\mathbf{B}}_n = [\bar{\mathbf{B}}_{n1}, \ldots, \bar{\mathbf{B}}_{np}]$ with $\bar{\mathbf{B}}_{nj} = (\bar{b}_{jk}(X_{ij}))_{1 \leq i \leq n, 1 \leq k \leq d}$, the solution to

$$(\bar{\mathbf{B}}_n^T \bar{\mathbf{B}}_n)\alpha = \bar{\mathbf{B}}_n^T \mathbf{Y}$$

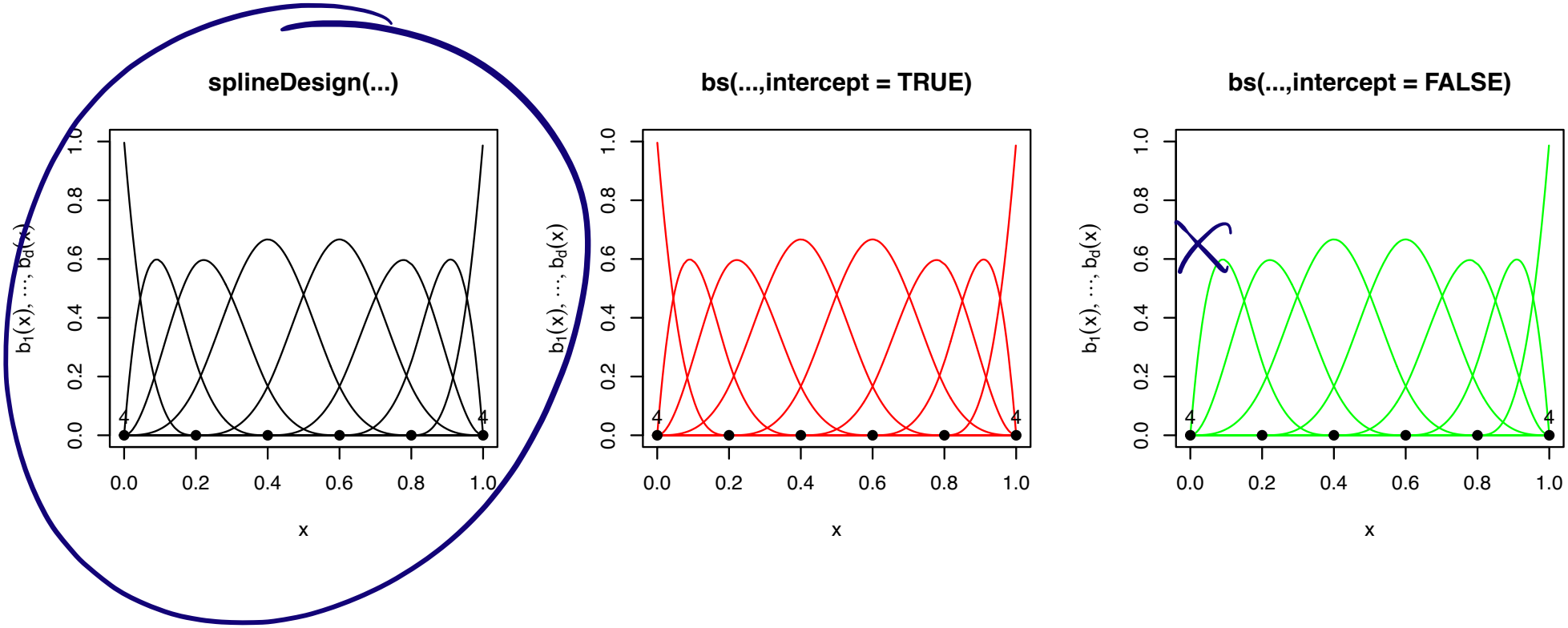
is not unique, because the $\bar{\mathbf{B}}_{n1}, \ldots, \bar{\mathbf{B}}_{np}$ do not have full-column rank—due to:

The B-splines have the property that $\sum_{\ell=1}^{d} b_{j\ell}(x) = 1$ for all $x \in [0, 1]$.

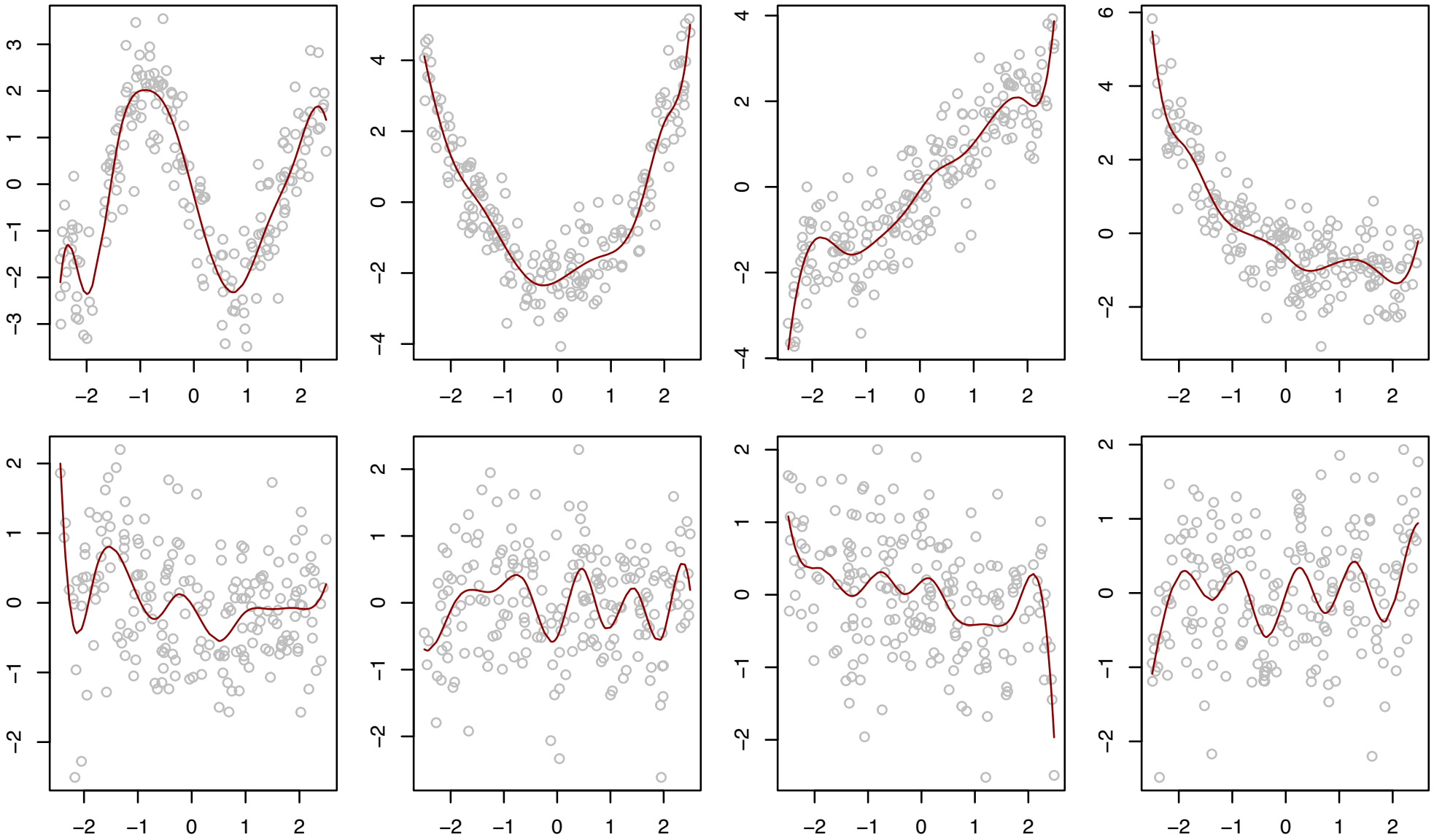
One fix is to discard the first basis function $b_{11}, \ldots, b_{p1}$ for each component...

Illustrate: Write up some code for fitting LS splines in the additive model.

The `bs()` function with `intercept = FALSE` removes the first basis function:



Least–squares splines estimator (directly computed with basis functions centered, 1 removed)



We can also penalize the wiggliness of the fitted functions:

A penalized splines estimator for the additive model

Penalized spline estimators $\hat{m}_1^{\text{pspl}}, \ldots, \hat{m}_p^{\text{pspl}}$ of $m_1, \ldots, m_p$ may be defined as

$$\left(\hat{m}_1^{\text{pspl}}, \ldots, \hat{m}_p^{\text{pspl}}\right) = \underset{g_j \in \bar{\mathcal{M}}_{nj}}{\text{argmin}} \sum_{i=1}^n \left[Y_i - \sum_{j=1}^p g_j(X_{ij})\right]^2 + \lambda \sum_{j=1}^p \int_0^1 [g_j''(x)]^2 dx,$$

for some $\lambda > 0$, where for $j = 1, \ldots, p$,

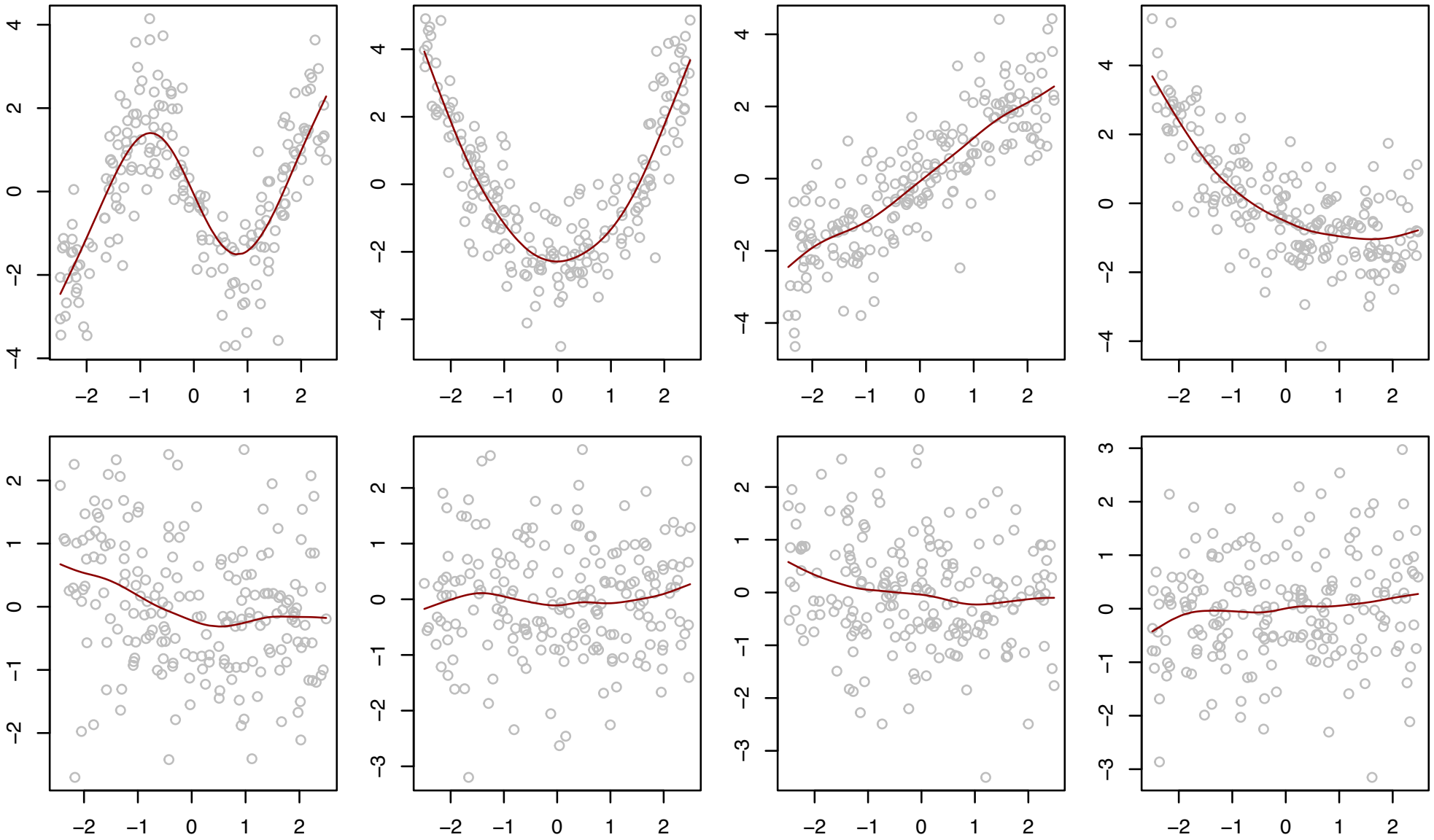
$$\bar{\mathcal{M}}_{nj} = \text{span}\{\bar{b}_{j1}, \ldots, \bar{b}_{jd}\}, \text{ with } \bar{b}_{jl}(x) = b_{jl}(x) - n^{-1} \sum_{i=1}^n b_{jl}(X_{ij}),$$

for $l = 1, \ldots, d$, where $b_{j1}, \ldots, b_{jd}$ are cubic B-spline basis functions.

Exercise:

1. Write the objective function in matrices. Give normal equations. Issues?
2. Show and run sample *R code* for fitting the penalized splines estimator.

Penalized splines estimator (directly computed with basis functions centered, 1 removed)



Now consider the performance of nonparametric estimators in the additive model.

Least-squares splines performance in additive model, Stone (1985), [5]

Suppose $m = m_1 + \cdots + m_p$, where $m_j \in \mathcal{H}(\beta, L)$ for $j = 1, \ldots, p$, and let $\hat{\mathbf{m}}_{1,r}^{\text{spl}}, \ldots, \hat{\mathbf{m}}_{p,r}^{\text{spl}}$ be $n \times 1$ vectors with the fitted values of the least-squares splines estimators of order $r \geq \beta - 1$. Then, provided $X_1, \ldots, X_n$ have a "nice" distribution and $K_n = \alpha n^{\frac{1}{2\beta+1}}$ for some $\alpha > 0$, we have

$$\mathbb{E}\left(\frac{1}{n}\|\hat{\mathbf{m}}_{j,r}^{\text{spl}} - \mathbf{m}_j\|_2^2\right) \leq C \cdot n^{-\frac{2\beta}{2\beta+1}}$$

Basically the mean integrated squared error

I set 1 instead of p.

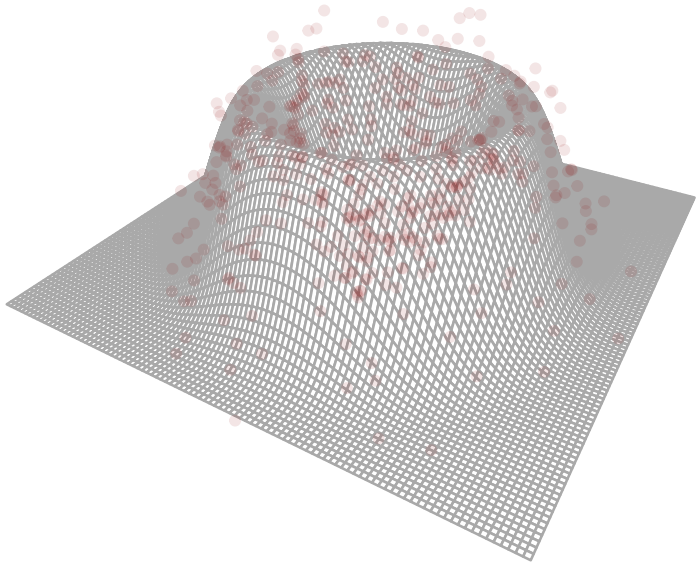
true function at design points

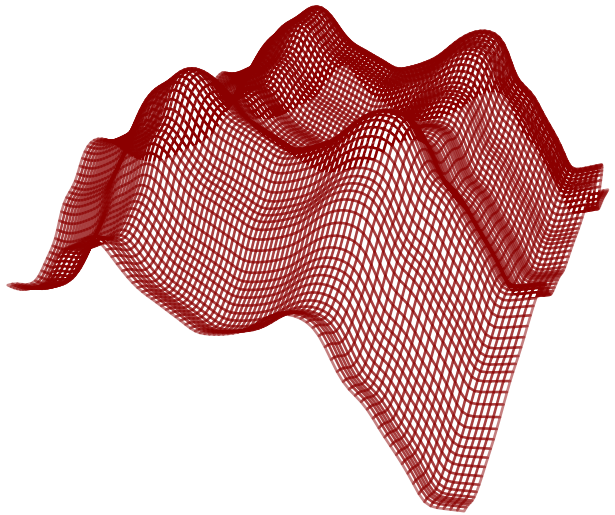
for each $j = 1, \ldots, p$, for some constant $C > 0$ for large enough n.

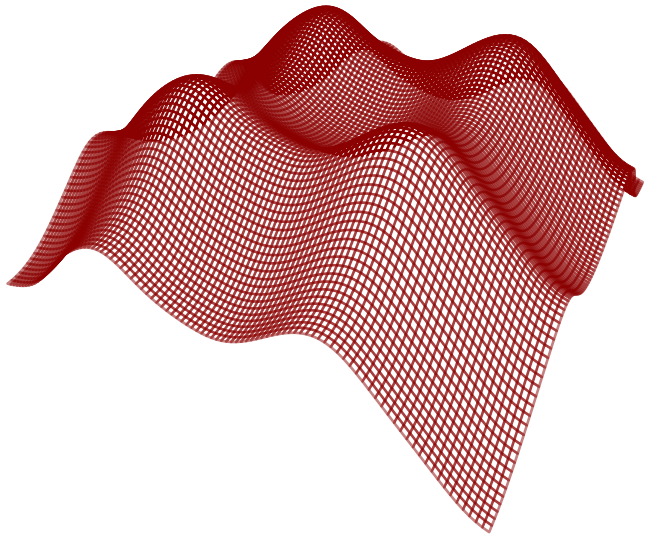
The $\mathbf{m}_1, \ldots, \mathbf{m}_p$ are $n \times 1$ with evaluations of the true functions at design points.

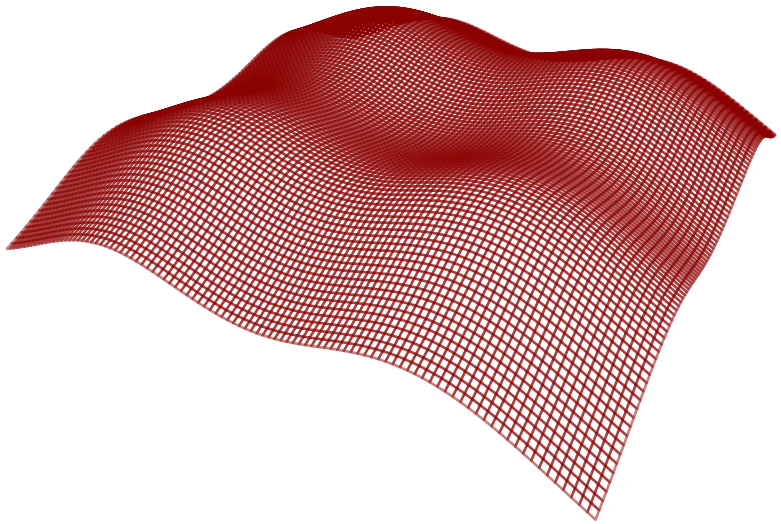
We estimate the additive model components at the univariate nonparametric rate!

Discuss: What if the additivity assumption is false?









Stone (1985) treated p as a constant, absorbing it into C. If we track p, we get

$$\mathbb{E}\left(\frac{1}{n}\|\hat{\mathbf{m}}_{j,r}^{\text{spl}} - \mathbf{m}_j\|_2^2\right) \leq C \cdot p \cdot n^{-\frac{2\beta}{2\beta+1}}.$$

So we see that the dimension, i.e. # of covariates, affects estimation.

The sparse additive model

In large-p settings, we often make a *sparsity assumption*; we assume

$$s := |\mathcal{A}| < p \quad \text{where} \quad \mathcal{A} = \{j : m_j \neq 0\}.$$

This means that some of the functions are equal to zero, giving

$$Y = \sum_{j \in \mathcal{A}} m_j(X_j) + \varepsilon.$$

The covariates with indices in $\mathcal{A}$ are sometimes called the "active" covariates.

Many estimators have been proposed in this setting.

Sparsity via the group lasso. See Huang et al. (2010), [2].

Group lasso estimators $\hat{m}_1^{\mathsf{L}}, \ldots, \hat{m}_1^{\mathsf{L}}$ of $m_1, \ldots, m_p$ can be defined as

$$\hat{m}_j^{\mathsf{L}}(x) = \sum_{k=1}^{d} \hat{\alpha}_{jk}^{\mathsf{L}} \bar{b}_{jk}(x), \quad j = 1, \ldots, p,$$

where $\hat{\boldsymbol{\alpha}}_j^{\mathsf{L}} = (\hat{\alpha}_{j1}^{\mathsf{L}}, \ldots, \hat{\alpha}_{jd}^{\mathsf{L}})^T$, $j = 1, \ldots, p$ are given by

$$(\hat{\boldsymbol{\alpha}}_1^{\mathsf{L}}, \ldots, \hat{\boldsymbol{\alpha}}_p^{\mathsf{L}}) = \underset{\boldsymbol{\alpha}_j \in \mathbb{R}^d}{\operatorname{argmin}} \, \|\mathbf{Y} - \sum_{j=1}^{p} \bar{\mathbf{B}}_{nj} \boldsymbol{\alpha}_j\|_2^2 + \lambda \sum_{j=1}^{p} \|\boldsymbol{\alpha}_j\|_2,$$

with $\bar{\mathbf{B}}_{nj} = (\bar{b}_{jk}(X_{ij}))_{1 \le i \le n, 1 \le k \le d}$, $j = 1, \ldots, p$.

Can get adaptive lasso estimators $\hat{m}_j^{\mathsf{AL}}(x) = \sum_{k=1}^{d} \hat{\alpha}_{jk}^{\mathsf{AL}} \bar{b}_{jk}(x), \quad j = 1, \ldots, p$, with

$$(\hat{\boldsymbol{\alpha}}_1^{\mathsf{AL}}, \ldots, \hat{\boldsymbol{\alpha}}_p^{\mathsf{AL}}) = \underset{\boldsymbol{\alpha}_j \in \mathbb{R}^d}{\operatorname{argmin}} \, \|\mathbf{Y} - \sum_{j=1}^{p} \bar{\mathbf{B}}_{nj} \boldsymbol{\alpha}_j\|_2^2 + \lambda_{\mathsf{A}} \sum_{j=1}^{p} \frac{1}{\|\hat{\boldsymbol{\alpha}}_j^{\mathsf{L}}\|_2} \cdot \|\boldsymbol{\alpha}_j\|_2,$$

A sparse penalized splines estimator. See Meier et al. (2009), [3].

Sparse pen. spline estimators $\hat{m}_1^{\text{spspl}}, \ldots, \hat{m}_p^{\text{spspl}}$ of $m_1, \ldots, m_p$ may be defined as

$$\left(\hat{m}_1^{\text{spspl}}, \ldots, \hat{m}_p^{\text{spspl}}\right) = \underset{g_j \in \bar{\mathcal{M}}_{nj}}{\text{argmin}} \sum_{i=1}^{n} \left[Y_i - \sum_{j=1}^{p} g_j(X_{ij}) \right]^2 + \lambda \sum_{j=1}^{p} \sqrt{n^{-1} \sum_{i=1}^{n} [g_j(X_{ij})]^2 + \xi \int_0^1 [g_j''(x)]^2 dx},$$

for some $\lambda > 0, \xi \geq 0$, where for $j = 1, \ldots, p$,

$$\bar{\mathcal{M}}_{nj} = \text{span}\{\bar{b}_{j1}, \ldots, \bar{b}_{jd}\}, \text{ with } \bar{b}_{jl}(x) = b_{jl}(x) - n^{-1} \textstyle\sum_{i=1}^{n} b_{jl}(X_{ij}),$$

for $l = 1, \ldots, d$, where $b_{j1}, \ldots, b_{jd}$ are cubic B-spline basis functions.

Exercise: Show how this can be formulated as a group lasso problem.

We can also impose sparsity by soft-thresholding the backfitting algorithm.

Soft-thresholded backfitting algorithm. Ravikumar et al. (2009), [4]

Initialize: $\hat{\mathbf{m}}_1, \ldots, \hat{\mathbf{m}}_p = \mathbf{0}$. Then iterate: For $j = 1, \ldots, p$

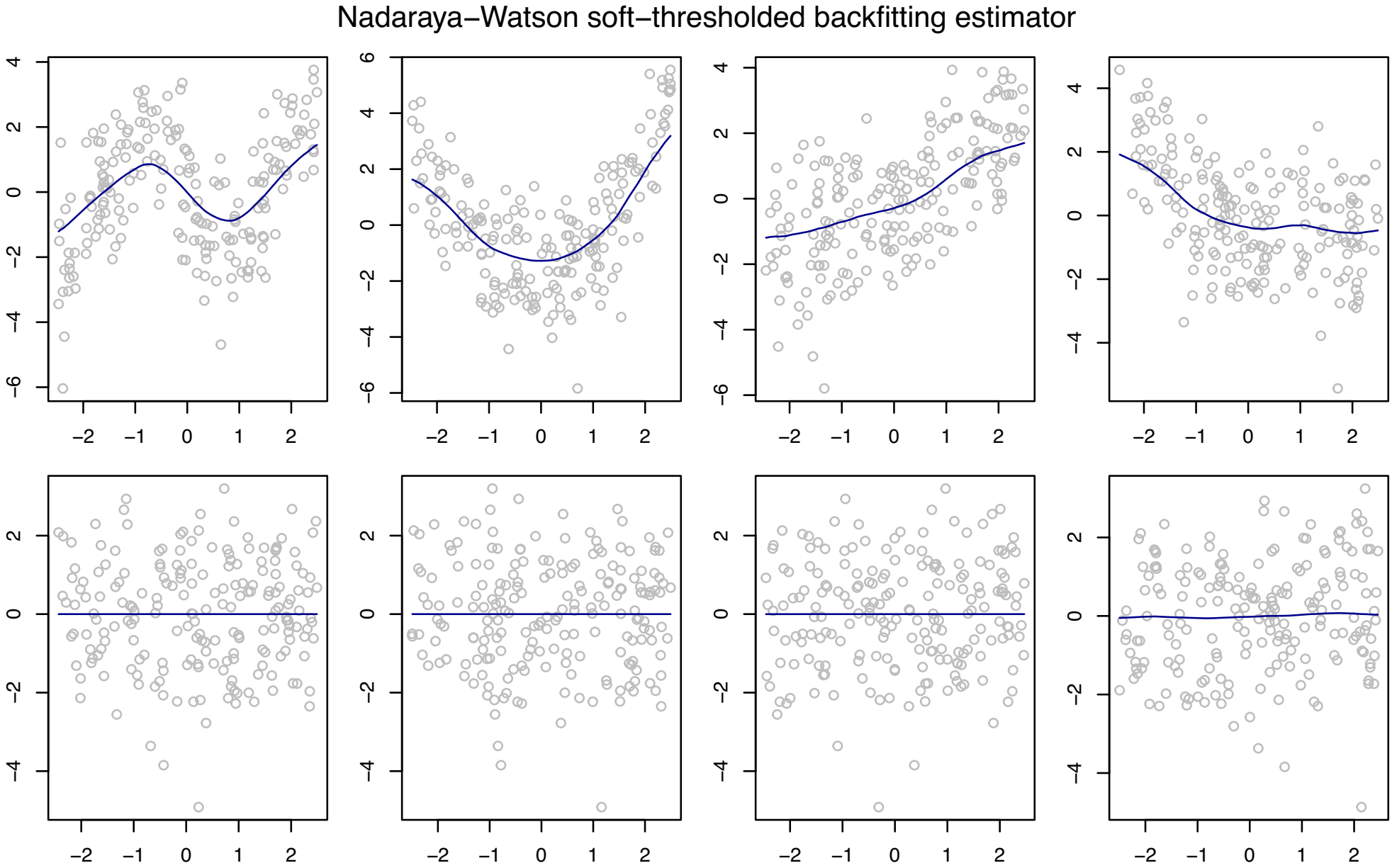
1. $\hat{\mathbf{m}}_j \leftarrow \mathbf{S}_j(\mathbf{Y} - \sum_{k \neq j} \hat{\mathbf{m}}_k)$
2. $\hat{\mathbf{m}}_j \leftarrow \begin{cases} \hat{\mathbf{m}}_j \cdot (\|\hat{\mathbf{m}}_j\|_n - \lambda)/\|\hat{\mathbf{m}}_j\|_n, & \text{if } \|\hat{\mathbf{m}}_j\|_n \geq \lambda \\ \mathbf{0}, & \text{if } \|\hat{\mathbf{m}}_j\|_n < \lambda \end{cases}$
3. $\hat{\mathbf{m}}_j \leftarrow \hat{\mathbf{m}}_j - n^{-1}\mathbf{1}_n^T \hat{\mathbf{m}}_j$ (centering step for identifiability)

until $\hat{\mathbf{m}}_1, \ldots, \hat{\mathbf{m}}_p$ no longer change.

$$\|\hat{\mathbf{m}}_j\|_n = \sqrt{\frac{1}{n}\sum_{i=1}^n \left(\hat{m}_j(x_{ij})\right)^2} = \frac{1}{\sqrt{n}}\|\hat{\mathbf{m}}_j\|$$

In the above $\|\hat{\mathbf{m}}_j\|_n^2$ denotes the mean of the squared entries of $\hat{\mathbf{m}}_j$.

We can apply this to any linear smoother.


Nadaraya–Watson soft–thresholded backfitting estimator


Andreas Buja, Trevor Hastie, and Robert Tibshirani.
Linear smoothers and additive models.
The Annals of Statistics, pages 453–510, 1989.


Jian Huang, Joel L Horowitz, and Fengrong Wei.
Variable selection in nonparametric additive models.
The Annals of Statistics, 38(4):2282, 2010.


Lukas Meier, Sara Van de Geer, Peter Bühlmann, et al.
High-dimensional additive modeling.
The Annals of Statistics, 37(6B):3779–3821, 2009.


Pradeep Ravikumar, John Lafferty, Han Liu, and Larry Wasserman.
Sparse additive models.
Journal of the Royal Statistical Society: Series B (Statistical Methodology), 71(5):1009–1030, 2009.


Charles J Stone.
Additive regression and other nonparametric models.
The Annals of Statistics, pages 689–705, 1985.