Smoothing splines

Statistical learning reading group

Alexander Ly

Psychological Methods
University of Amsterdam

Amsterdam, 8 March 2016

Overview

Regression

- There exists a true function f^* such that $y = f^*(x) + \epsilon$.
  Goal: Give a *single* best guess $\hat{f}(x)$ of $f^*(x)$ based on finite samples $\binom{x_1}{y_1}, \dots, \binom{x_n}{y_n}$.

Regression

- There exists a true function f^* such that $y = f^*(x) + \epsilon$. Goal: Give a *single* best guess $\hat{f}(x)$ of $f^*(x)$ based on finite samples $\binom{x_1}{y_1}, \ldots, \binom{x_n}{y_n}$.
- Step 1: Define "best guess" aka define a loss function

$$E(f^*(x) - \hat{f}(x))^2 \tag{1}$$

Regression

- There exists a true function f^* such that $y = f^*(x) + \epsilon$. Goal: Give a *single* best guess $\hat{f}(x)$ of $f^*(x)$ based on finite samples $\binom{x_1}{y_1}, \ldots, \binom{x_n}{y_n}$.
- Step 1: Define "best guess" aka define a loss function

$$E(f^*(x) - \hat{f}(x))^2 \tag{1}$$

- Step 2: Define a candidate collection of functions $\mathcal{F}$

Regression

- There exists a true function f^* such that $y = f^*(x) + \epsilon$. Goal: Give a *single* best guess $\hat{f}(x)$ of $f^*(x)$ based on finite samples $\binom{x_1}{y_1}, \ldots, \binom{x_n}{y_n}$.
- Step 1: Define "best guess" aka define a loss function

$$E(f^*(x) - \hat{f}(x))^2 \tag{1}$$

- Step 2: Define a candidate collection of functions $\mathcal{F}$
- Step 3: Calculate the (empirical) loss for each single candidate $\tilde{f}$ in $\mathcal{F}$

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{f}(x_i))^2 \tag{2}$$

Regression

- There exists a true function f^* such that $y = f^*(x) + \epsilon$. Goal: Give a *single* best guess $\hat{f}(x)$ of $f^*(x)$ based on finite samples $\binom{x_1}{y_1}, \ldots, \binom{x_n}{y_n}$.
- Step 1: Define "best guess" aka define a loss function

$$E(f^*(x) - \hat{f}(x))^2 \tag{1}$$

- Step 2: Define a candidate collection of functions $\mathcal{F}$
- Step 3: Calculate the (empirical) loss for each single candidate $\tilde{f}$ in $\mathcal{F}$

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{f}(x_i))^2 \tag{2}$$

- Step 4: Minimise: Take as best guess:

$$\hat{f}(x) = \underset{\tilde{f}\in\mathcal{F}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{f}(x_i))^2 \tag{3}$$

Example of $\mathcal{F}$: Linear regression

Trick: frame problem in terms of matrix algebra:

$$y = X\theta + \epsilon \tag{4}$$

observed $y \in \mathbb{R}^n$, observed design Matrix $X \in \mathbb{R}^{n \times p}$,
parameters $\theta \in \mathbb{R}^p$
Pro:

Con:

Example of $\mathcal{F}$: Linear regression

Trick: frame problem in terms of matrix algebra:

$$y = X\theta + \epsilon \tag{4}$$

observed $y \in \mathbb{R}^n$, observed design Matrix $X \in \mathbb{R}^{n \times p}$, parameters $\theta \in \mathbb{R}^p$

Pro:

- Computationally: No need to calculate the loss for each $f \in \mathcal{F}$. Solve by matrix algebra $\hat{\theta} = (X^T X)^{-1} X^T y$

Con:

Example of $\mathcal{F}$: Linear regression

Trick: frame problem in terms of matrix algebra:

$$y = X\theta + \epsilon \tag{4}$$

observed $y \in \mathbb{R}^n$, observed design Matrix $X \in \mathbb{R}^{n \times p}$, parameters $\theta \in \mathbb{R}^p$

Pro:

- Computationally: No need to calculate the loss for each $f \in \mathcal{F}$. Solve by matrix algebra $\hat{\theta} = (X^T X)^{-1} X^T y$
- Unique minimiser: is the plugin $\hat{f}(x_{\text{new}}) = \hat{\theta} x_{\text{new}}$

Con:

Example of $\mathcal{F}$: Linear regression

Trick: frame problem in terms of matrix algebra:

$$y = X\theta + \epsilon \tag{4}$$

observed $y \in \mathbb{R}^n$, observed design Matrix $X \in \mathbb{R}^{n \times p}$, parameters $\theta \in \mathbb{R}^p$

Pro:

- Computationally: No need to calculate the loss for each $f \in \mathcal{F}$. Solve by matrix algebra $\hat{\theta} = (X^T X)^{-1} X^T y$
- Unique minimiser: is the plugin $\hat{f}(x_{\text{new}}) = \hat{\theta} x_{\text{new}}$

Con:

- Misspecification The true f^* is most likely not linear, thus, $f^* \notin \mathcal{F}$

Other examples of $\mathcal{F}$:

The $p \ll n$ “regime” (i.e., matrix trick is okay):

The $n \ll p$ “regime” (i.e., no uniqueness):

Other examples of $\mathcal{F}$:

The $p \ll n$ "regime" (i.e., matrix trick is okay):

- Polynomials $\mathcal{F}_m := \left\{ f(x) = \theta_{m-1} x^{m-1} + \ldots + \theta_1 x + \theta_0 \right\}$ with m parameters

The $n \ll p$ "regime" (i.e., no uniqueness):

Other examples of $\mathcal{F}$:

The $p \ll n$ "regime" (i.e., matrix trick is okay):

- Polynomials $\mathcal{F}_m := \left\{ f(x) = \theta_{m-1} x^{m-1} + \ldots + \theta_1 x + \theta_0 \right\}$ with m parameters
- Piecewise polynomials $\mathcal{F}_{m,K}$ with $mK + m$ parameters

The $n \ll p$ "regime" (i.e., no uniqueness):

Other examples of $\mathcal{F}$:

The $p \ll n$ “regime” (i.e., matrix trick is okay):

- Polynomials $\mathcal{F}_m := \left\{ f(x) = \theta_{m-1} x^{m-1} + \ldots + \theta_1 x + \theta_0 \right\}$ with m parameters
- Piecewise polynomials $\mathcal{F}_{m,K}$ with $mK + m$ parameters
- Polynomial splines with $m + K$ parameters

The $n \ll p$ “regime” (i.e., no uniqueness):

Other examples of $\mathcal{F}$:

The $p \ll n$ "regime" (i.e., matrix trick is okay):

- Polynomials $\mathcal{F}_m := \left\{ f(x) = \theta_{m-1} x^{m-1} + \ldots + \theta_1 x + \theta_0 \right\}$ with m parameters
- Piecewise polynomials $\mathcal{F}_{m,K}$ with $mK + m$ parameters
- Polynomial splines with $m + K$ parameters
- Natural splines with K parameters

The $n \ll p$ "regime" (i.e., no uniqueness):

Other examples of $\mathcal{F}$:

The $p \ll n$ "regime" (i.e., matrix trick is okay):

- Polynomials $\mathcal{F}_m := \left\{ f(x) = \theta_{m-1} x^{m-1} + \ldots + \theta_1 x + \theta_0 \right\}$ with m parameters
- Piecewise polynomials $\mathcal{F}_{m,K}$ with $mK + m$ parameters
- Polynomial splines with $m + K$ parameters
- Natural splines with K parameters
- Spoiler: Relationship number of parameters and smoothing

The $n \ll p$ "regime" (i.e., no uniqueness):

Other examples of $\mathcal{F}$:

The $p \ll n$ "regime" (i.e., matrix trick is okay):

- Polynomials $\mathcal{F}_m := \left\{ f(x) = \theta_{m-1} x^{m-1} + \ldots + \theta_1 x + \theta_0 \right\}$ with m parameters
- Piecewise polynomials $\mathcal{F}_{m,K}$ with $mK + m$ parameters
- Polynomial splines with $m + K$ parameters
- Natural splines with K parameters
- Spoiler: Relationship number of parameters and smoothing

The $n \ll p$ "regime" (i.e., no uniqueness):

- Smoothing splines with "uncountably many parameters"

Other examples of $\mathcal{F}$:

The $p \ll n$ “regime” (i.e., matrix trick is okay):

- Polynomials $\mathcal{F}_m := \left\{ f(x) = \theta_{m-1} x^{m-1} + \ldots + \theta_1 x + \theta_0 \right\}$ with m parameters
- Piecewise polynomials $\mathcal{F}_{m,K}$ with $mK + m$ parameters
- Polynomial splines with $m + K$ parameters
- Natural splines with K parameters
- Spoiler: Relationship number of parameters and smoothing

The $n \ll p$ “regime” (i.e., no uniqueness):

- Smoothing splines with “uncountably many parameters”
- Spoiler: Relationship degree of freedom and tuning parameter

Key in the $p \ll n$ regime

To apply the "matrix trick" in case of $p \ll n$ (polynomials, piecewise polynomials, polynomial splines and natural splines) use basis functions (i.e., transform x).

- Use powers of x for non-linear behaviour:

$$g_j(x) = x^j \quad (5)$$

Key in the $p \ll n$ regime

To apply the "matrix trick" in case of $p \ll n$ (polynomials, piecewise polynomials, polynomial splines and natural splines) use basis functions (i.e., transform x).

- Use powers of x for non-linear behaviour:

$$g_j(x) = x^j \tag{5}$$

- Use indicator functions for local behaviour:

$$g_j(x) = 1_{(\xi_{j-1},\xi_j]}(x) := \begin{cases} 1 & \text{if } x \in (\xi_{j-1}, \xi_j] \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Key in the $p \ll n$ regime

To apply the "matrix trick" in case of $p \ll n$ (polynomials, piecewise polynomials, polynomial splines and natural splines) use basis functions (i.e., transform x).

- Use powers of x for non-linear behaviour:

$$g_j(x) = x^j \tag{5}$$

- Use indicator functions for local behaviour:

$$g_j(x) = 1_{(\xi_{j-1},\xi_j]}(x) := \begin{cases} 1 & \text{if } x \in (\xi_{j-1}, \xi_j] \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

- Combination of the two

Polynomial regression

$$\mathcal{F}_m := \left\{ f(x) = \theta_{m-1} x^{m-1} + \ldots + \theta_0 \right\} = \left\{ f(x) = \sum_{j=0}^{m-1} \theta_j g_j(x) \right\},$$

thus $g_j(x) = x^j$.

Polynomial regression

$$\mathcal{F}_m := \left\{ f(x) = \theta_{m-1} x^{m-1} + \ldots + \theta_0 \right\} = \left\{ f(x) = \sum_{j=0}^{m-1} \theta_j g_j(x) \right\},$$

thus $g_j(x) = x^j$. Solution: Take $\hat{f}$ with

$$\hat{\theta} = (X^T X)^{-1} X^T y \tag{7}$$

where the design matrix is

$$X = \begin{pmatrix} g_0(x_1) & g_1(x_1) & \ldots & g_{m-1}(x_1) \\ g_0(x_2) & g_1(x_2) & \ldots & g_{m-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ g_0(x_n) & g_1(x_n) & \ldots & g_{m-1}(x_n) \end{pmatrix} = \begin{pmatrix} 1 & x_1^1 & \ldots & x_1^{m-1} \\ 1 & x_2^1 & \ldots & x_2^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n^1 & \ldots & x_n^{m-1} \end{pmatrix}$$

Polynomial regression

Solution: Take $\hat{f}$ with

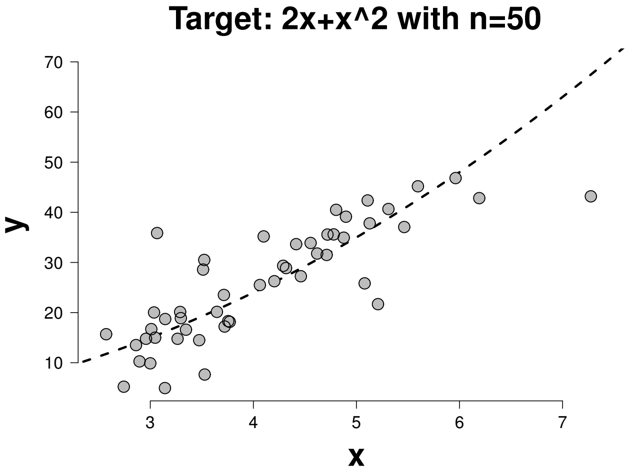
$$\hat{\theta} = (X^T X)^{-1} X^T y \tag{7}$$

where the design matrix is

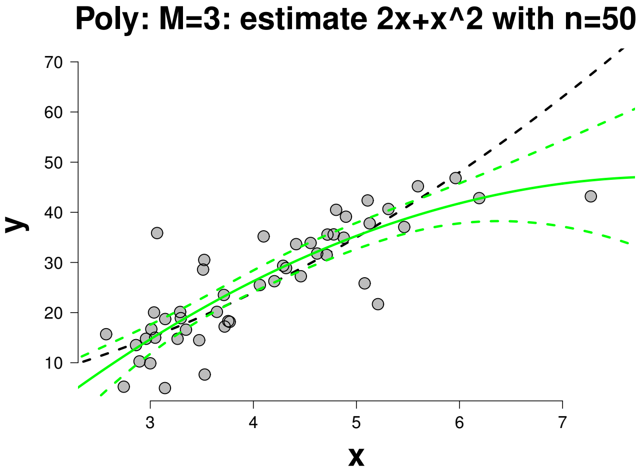
$$X = \begin{pmatrix} g_0(x_1) & g_1(x_1) & \dots & g_{m-1}(x_1) \\ g_0(x_2) & g_1(x_2) & \dots & g_{m-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ g_0(x_n) & g_1(x_n) & \dots & g_{m-1}(x_n) \end{pmatrix} = \begin{pmatrix} 1 & x_1^1 & \dots & x_1^{m-1} \\ 1 & x_2^1 & \dots & x_2^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n^1 & \dots & x_n^{m-1} \end{pmatrix}$$

Pick the order m by hand or by cross validation

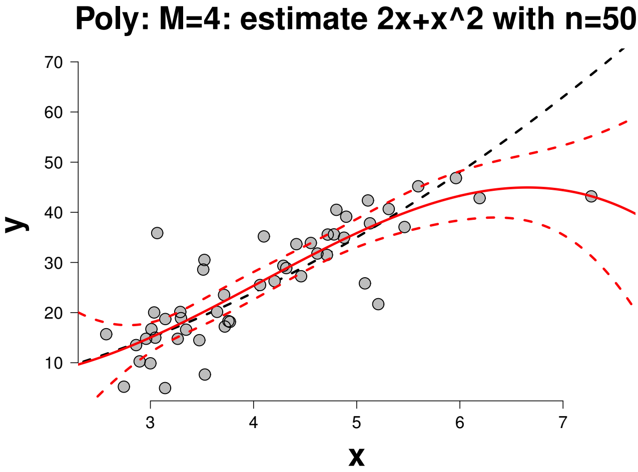
Target:



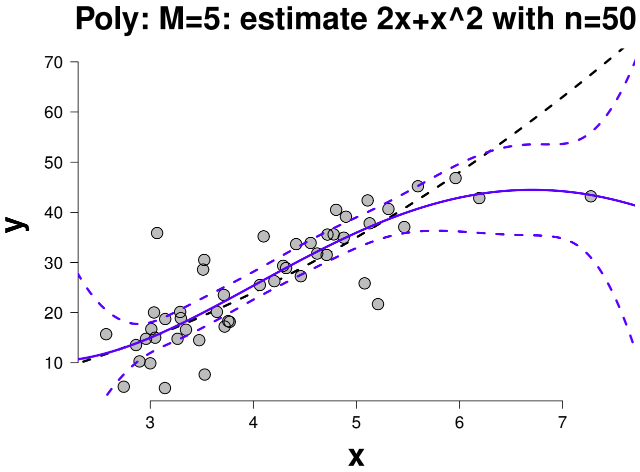
Polynomial regression:



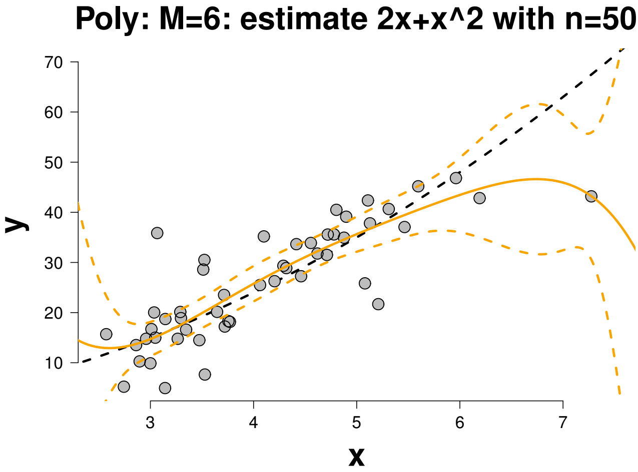
Polynomial regression:



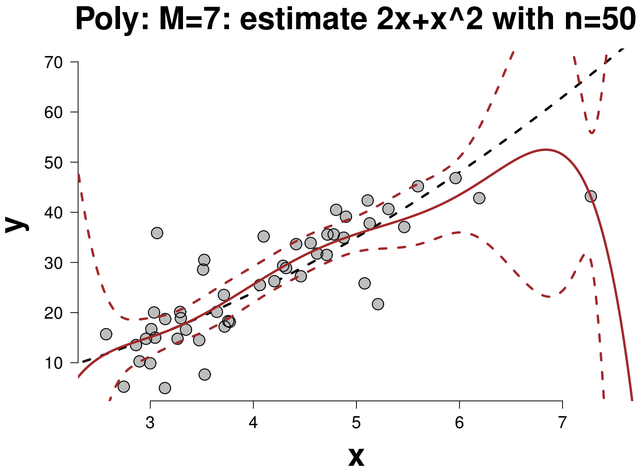
Polynomial regression:



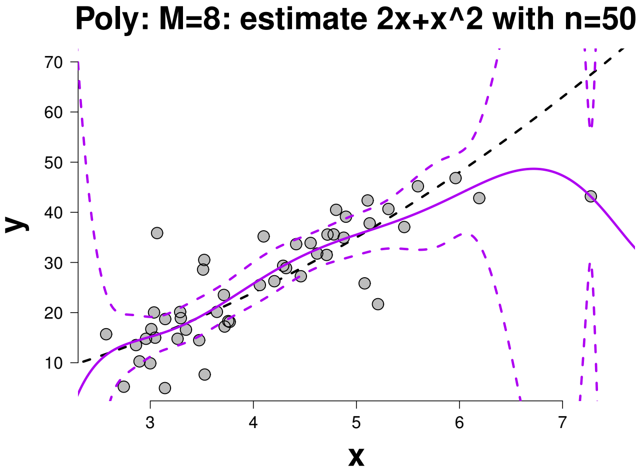
Polynomial regression:



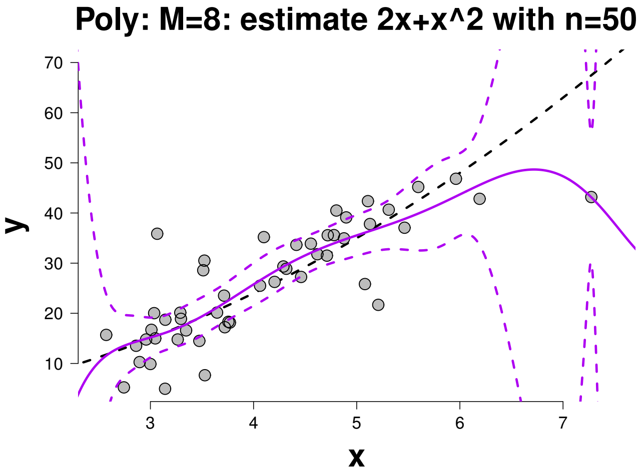
Polynomial regression:



Polynomial regression:



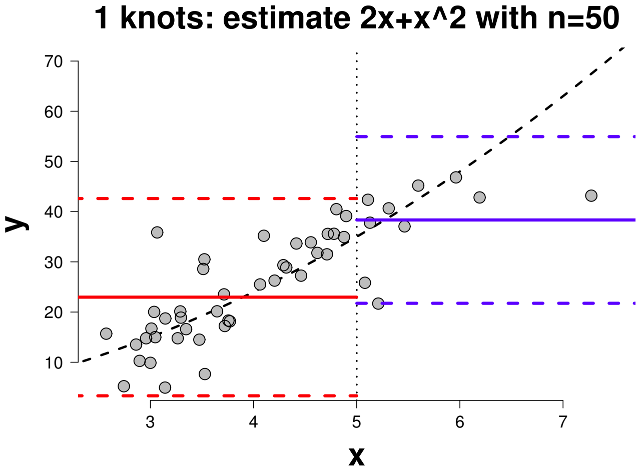
Polynomial regression:



Great in the middle (low bias), bad in the tails (high variance).

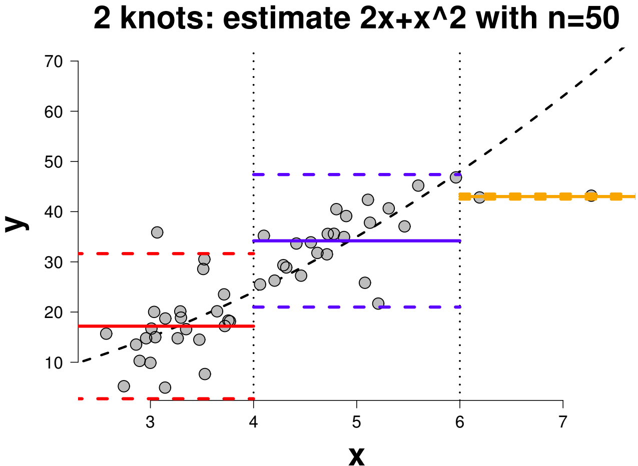
Example: Piecewise constants

Introduce knots $\xi_1, \ldots, \xi_K$ yielding $K + 1$ bins. Fit a constant function locally.



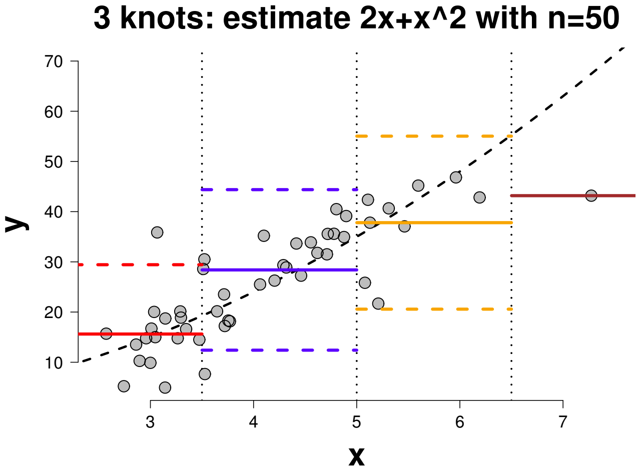
Example: Piecewise constants

Introduce knots $\xi_1, \ldots, \xi_K$ yielding $K + 1$ bins. Fit a constant function locally.



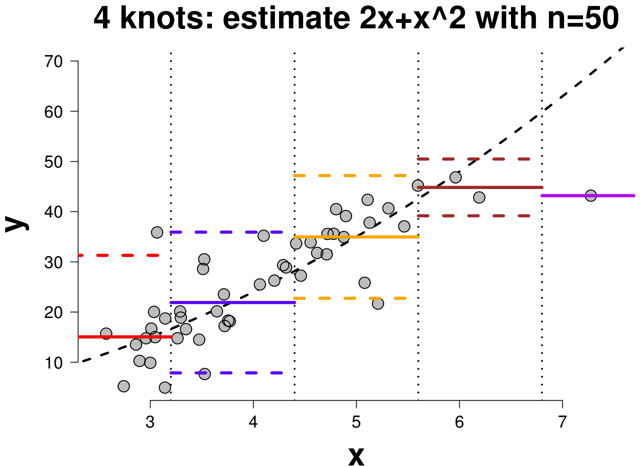
Example: Piecewise constants

Introduce knots $\xi_1, \ldots, \xi_K$ yielding $K+1$ bins. Fit a constant function locally.



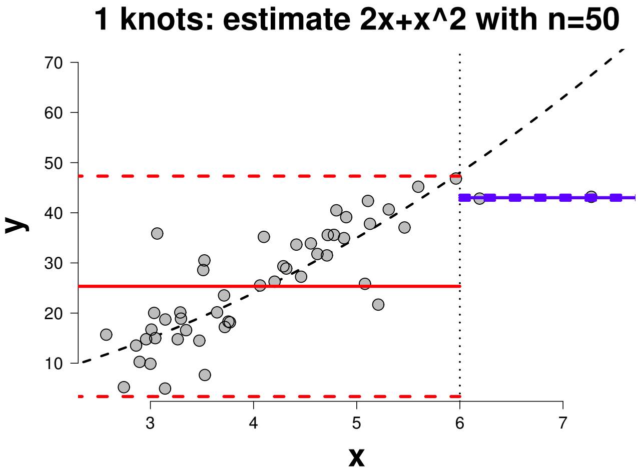
Example: Piecewise constants

Introduce knots $\xi_1, \ldots, \xi_K$ yielding $K+1$ bins. Fit a constant function locally.



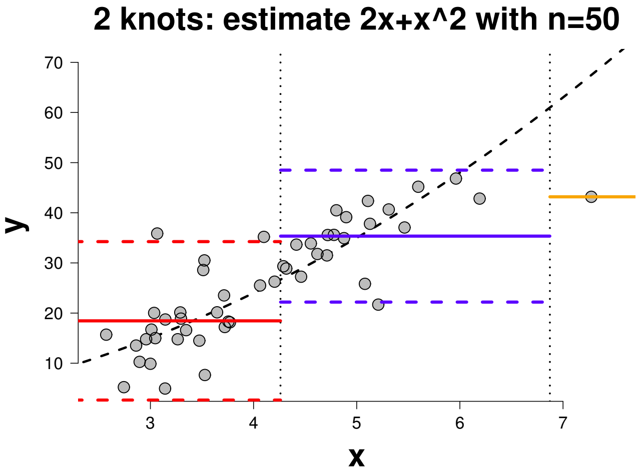
Example: Piecewise constants

Depends on *K* and where



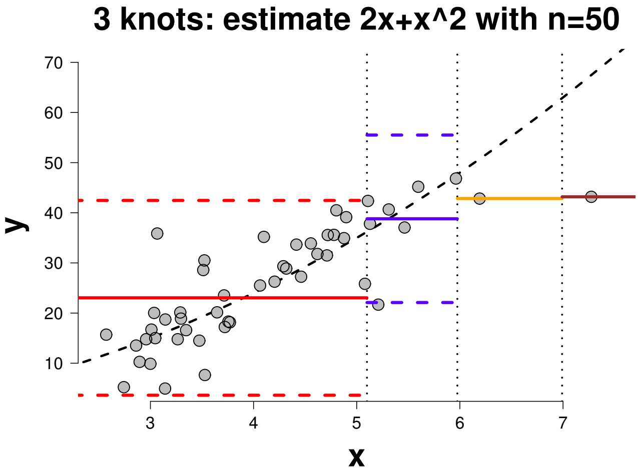
Example: Piecewise constants

Depends on *K* and where



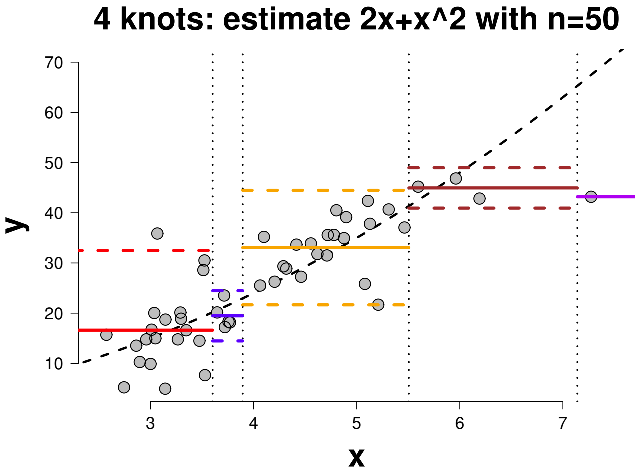
Example: Piecewise constants

Depends on *K* and where



Example: Piecewise constants

Depends on *K* and where



Basis functions

Piecewise constants:

$$\mathcal{F}_K := \left\{ f(x) = \sum_{k=0}^{K} \theta_k g_k(x) \right\},$$

thus $g_k(x) = \mathbf{1}_{(\xi_{k-1},\xi_k]}(x)$.

Basis functions

Piecewise constants:

$$\mathcal{F}_K := \left\{ f(x) = \sum_{k=0}^{K} \theta_k g_k(x) \right\},$$

thus $g_k(x) = \mathbf{1}_{(\xi_{k-1},\xi_k]}(x)$. Solution: Take $\hat{f}$ with

$$\hat{\theta} = (X^T X)^{-1} X^T y \tag{8}$$

where the design matrix is

$$X = \begin{pmatrix} g_0(x_1) & g_1(x_1) & \dots & g_{m-1}(x_1) \\ g_0(x_2) & g_1(x_2) & \dots & g_{m-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ g_0(x_n) & g_1(x_n) & \dots & g_{m-1}(x_n) \end{pmatrix} = \begin{pmatrix} 0 & 1 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

each row has only one “1”.

Global function, local modification

Piecewise constants:

$$\mathcal{F}_K := \left\{ f(x) = \sum_{k=0}^{K} \theta_k g_k(x) \right\},$$

thus $g_k(x) = \mathbf{1}_{(\xi_{k-1},\xi_k]}(x)$.

- Take the $g_0(x)$ just the whole range with a global parameter θ_0.

Global function, local modification

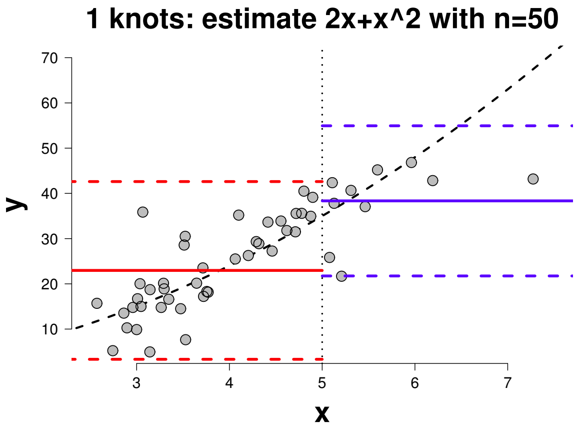
Piecewise constants:

$$\mathcal{F}_K := \left\{ f(x) = \sum_{k=0}^{K} \theta_k g_k(x) \right\},$$

thus $g_k(x) = \mathbf{1}_{(\xi_{k-1},\xi_k]}(x)$.

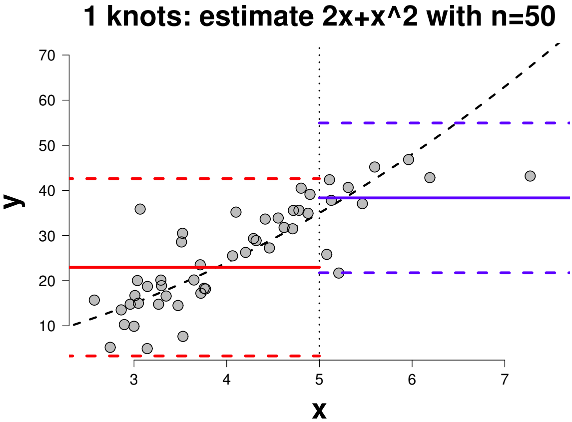
- Take the $g_0(x)$ just the whole range with a global parameter θ_0.
- Consider θ_k only the local modification of the kth interval $(\xi_{k-1}, \xi_k]$

Global function, local modification



- Local: $\theta_0 \approx 22$ on the zeroth interval
- Local: $\theta_1 \approx 39$ on the first interval

Global function, local modification



- Global: $\theta_0 \approx 22$ on the global interval
- Local: $\theta_1 \approx 17$ modification on the first interval

Piecewise polynomials

$$\mathcal{F}_{m,K} = f(x) = \begin{cases} \sum_{j=0}^{m-1} \theta_{j,1} x^j & \text{if } x \leq \xi_1 \\ \sum_{j=0}^{m-1} \theta_{j,2} x^j & \text{if } \xi_1 < x \leq \xi_2 \\ \vdots & \vdots \\ \sum_{j=0}^{m-1} \theta_{j,k} x^j & \text{if } \xi_{k-1} < x \leq \xi_k \\ \vdots & \vdots \\ \sum_{j=0}^{m-1} \theta_{j,K} x^j & \text{if } \xi_{K-1} < x \leq \xi_K \end{cases} \tag{9}$$

with $m(K+1)$ parameters. Thus,

$$\mathcal{F}_{m,k} = \left\{ f(x) = \sum_{j=0,k=1}^{m-1,K} \theta_{j,k} g_{j,k}(x) \right\} \tag{10}$$

where $g_{j,k}(x) = x^j \, 1_{(\xi_{k-1},\xi_k]}(x)$.

Basis functions

Piecewise polynomials

$$\mathcal{F}_{m,K} := \left\{ f(x) = \sum_{j,k} \theta_{j,k} g_{j,k}(x) \right\},$$

thus $g_{j,k}(x) = x^j \mathbf{1}_{(\xi_{k-1},\xi_k]}(x)$.

Basis functions

Piecewise polynomials

$$\mathcal{F}_{m,K} := \left\{ f(x) = \sum_{j,k} \theta_{j,k} g_{j,k}(x) \right\},$$

thus $g_{j,k}(x) = x^j \mathbf{1}_{(\xi_{k-1},\xi_k]}(x)$. Solution: Take $\hat{f}$ with

$$\hat{\theta} = (X^T X)^{-1} X^T y \tag{11}$$

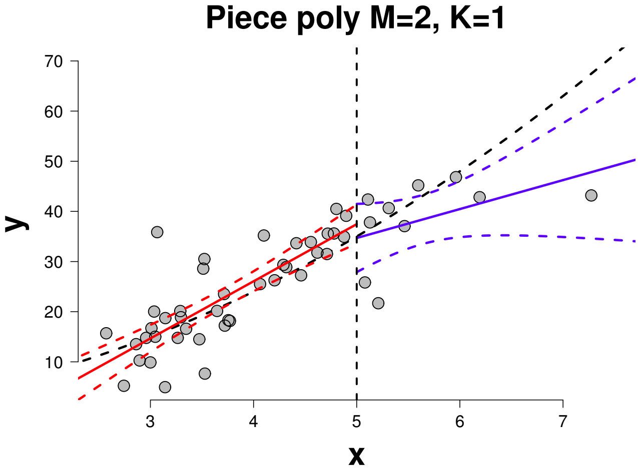
where the design matrix is

$$X = \begin{pmatrix} g_0(x_1) & g_1(x_1) & \dots & g_{m-1}(x_1) \\ g_0(x_2) & g_1(x_2) & \dots & g_{m-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ g_0(x_n) & g_1(x_n) & \dots & g_{m-1}(x_n) \end{pmatrix} = \begin{pmatrix} 0 & x & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & x^{m-1} \end{pmatrix}$$

each row has only one monomial "x^j".

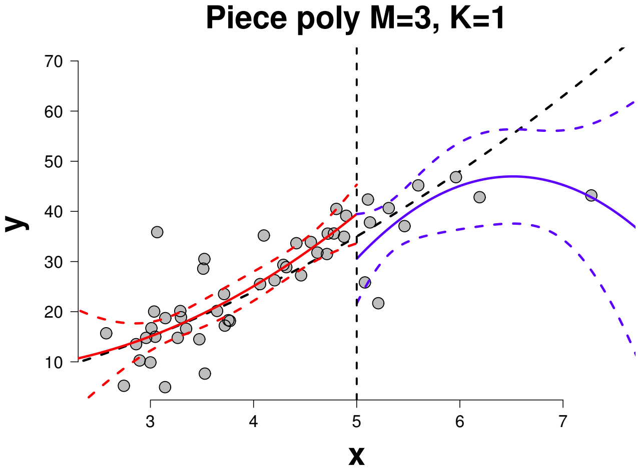
Example: Piecewise polynomials

$K = 1$ knot



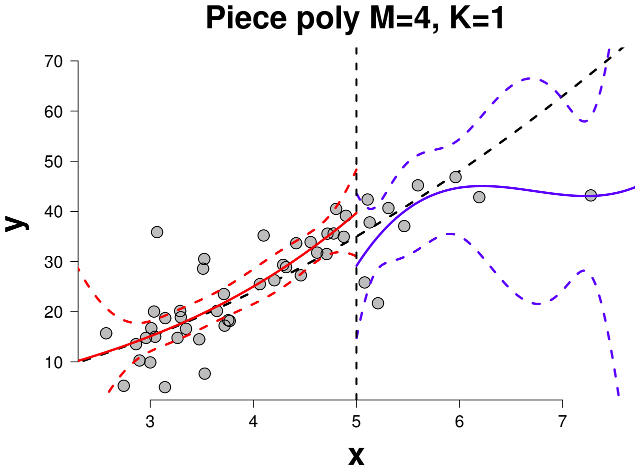
Example: Piecewise polynomials

$K = 1$ knot



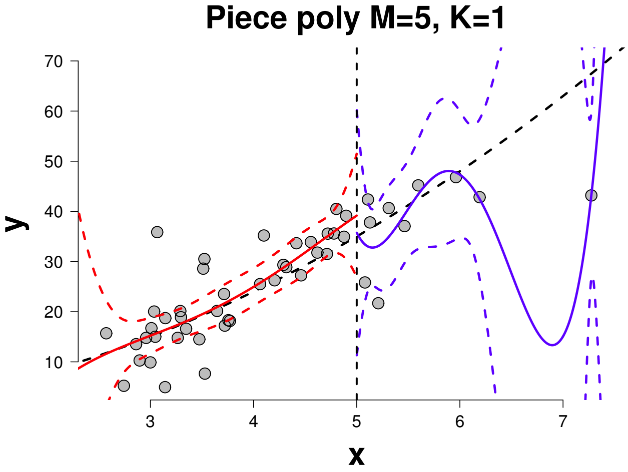
Example: Piecewise polynomials

$K = 1$ knot



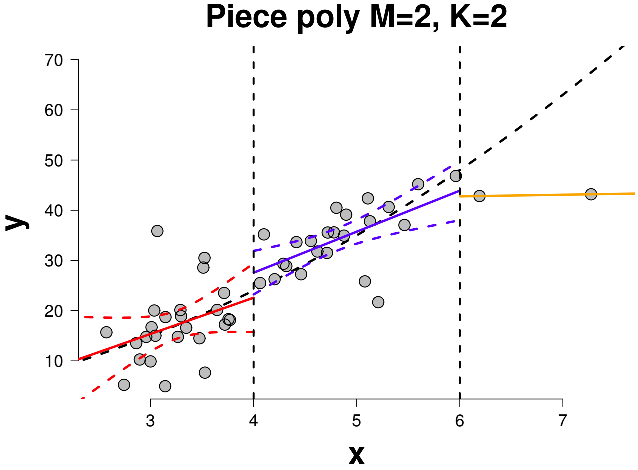
Example: Piecewise polynomials

$K = 1$ knot



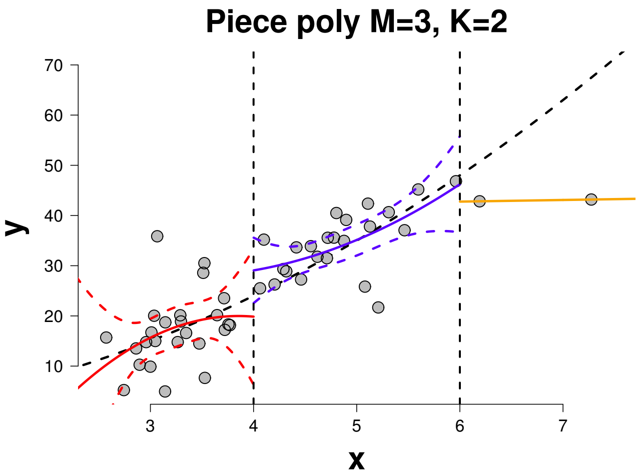
Example: Piecewise polynomials

Depends on *K* and where



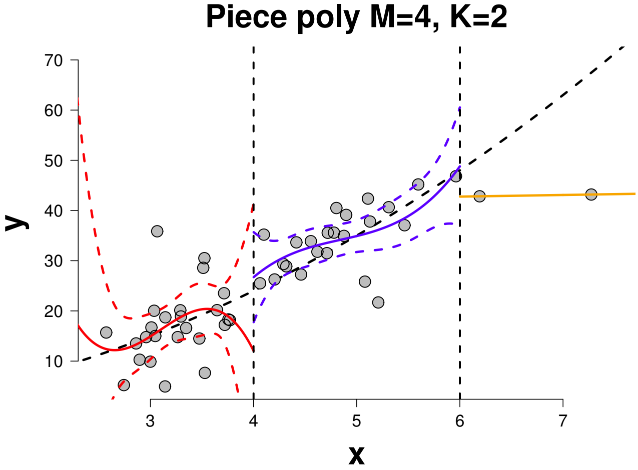
Example: Piecewise polynomials

Depends on *K* and where



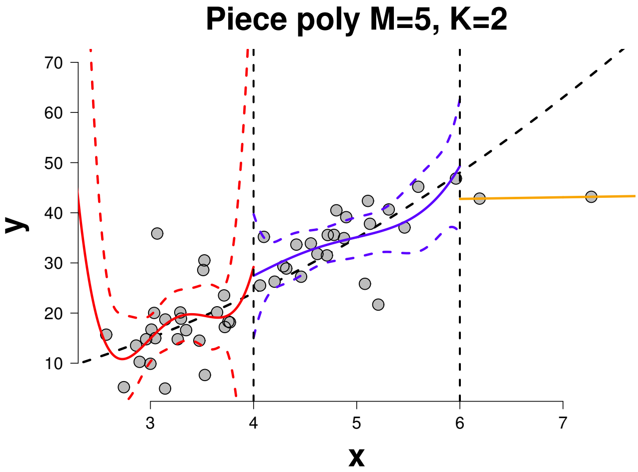
Example: Piecewise polynomials

Depends on K and where



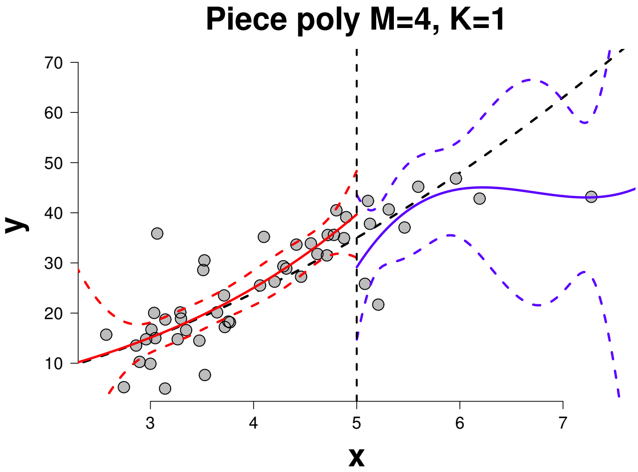
Example: Piecewise polynomials

Depends on *K* and where



Splines

Splines are piecewise polynomials that are smooth.



Splines

Splines are piecewise polynomials that are smooth. A polynomial spline of order m with K number of knots is has the basis functions:

- Global polynomial of order m

$$g_0(x) = x^0, \ldots, g_{m-1}(x) = x^{m-1} \tag{12}$$

Splines

Splines are piecewise polynomials that are smooth. A polynomial spline of order m with K number of knots is has the basis functions:

- Global polynomial of order m

$$g_0(x) = x^0, \dots, g_{m-1}(x) = x^{m-1} \tag{12}$$

- Local modifications:

$$g_{m+1}(x) = (x - \xi_1)_+^{m-1}, \dots, g_{m+K}(x) = (x - \xi_K)_+^{m-1} \tag{13}$$

thus, $m + K$ parameters.

Splines

Splines are piecewise polynomials that are smooth. A polynomial spline of order m with K number of knots is has the basis functions:

- Global polynomial of order m

$$g_0(x) = x^0, \ldots, g_{m-1}(x) = x^{m-1} \tag{12}$$

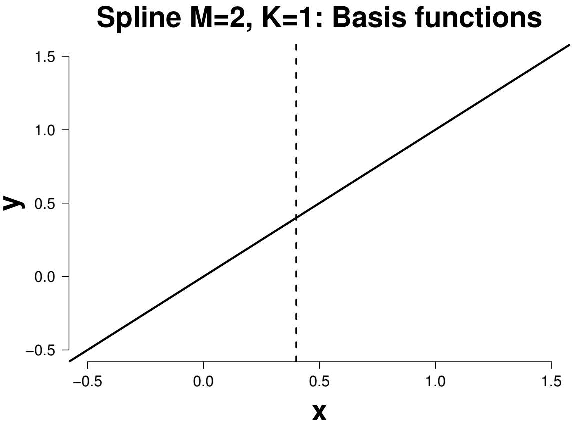
- Local modifications:

$$g_{m+1}(x) = (x - \xi_1)_+^{m-1}, \ldots, g_{m+K}(x) = (x - \xi_K)_+^{m-1} \tag{13}$$

  thus, $m + K$ parameters.
- Note $m + K < m(K + 1)$. Example cubic spline with two knots: $4 + 2$ vs 12 parameters.

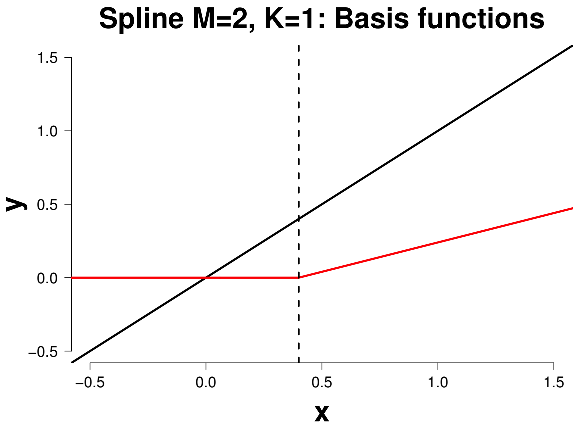
Example: Basis functions $M = 2$, $K = 1$

Knot at $\xi_1 = 0.4$
Global $\theta_1 = 1$



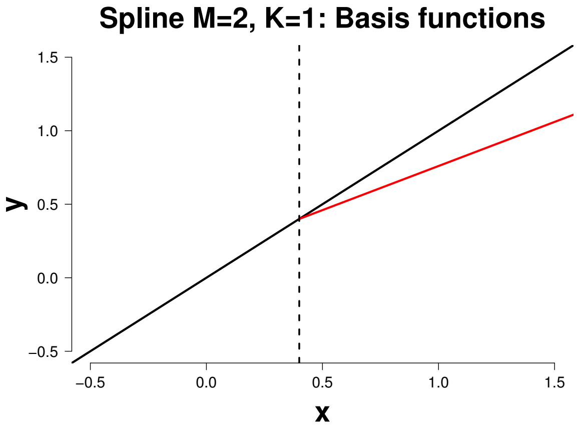
Example: Basis functions $M = 2$, $K = 1$

Knot at $\xi_1 = 0.4$
Global $\theta_1 = 1$ and local $\theta_{1+1} = 0.4$



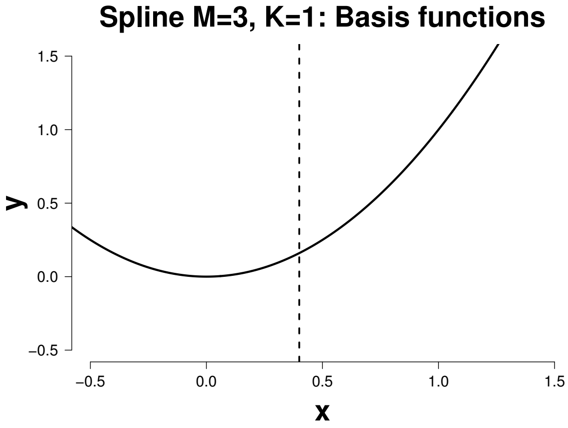
Example: Basis functions $M = 2$, $K = 1$

Knot at $\xi_1 = 0.4$
Modification: subtract $\theta_{1+1} = 0.4$ locally



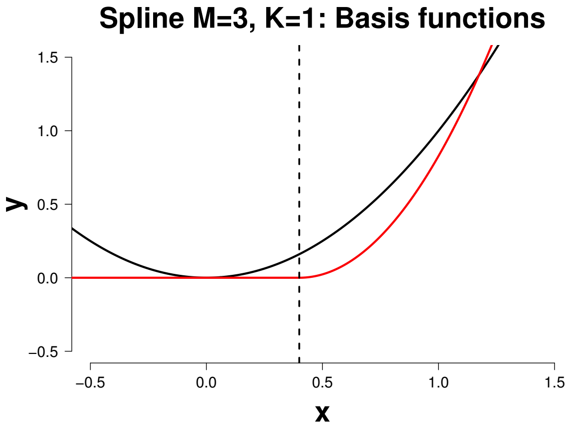
Example: Basis functions $M = 3$, $K = 1$

Knot at $\xi_1 = 0.4$
Global $\theta_2 = 1$



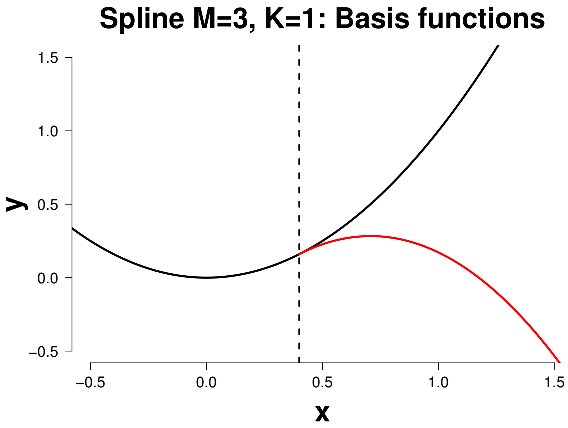
Example: Basis functions $M = 3$, $K = 1$

Knot at $\xi_1 = 0.4$
Global $\theta_2 = 1$ and local $\theta_{2+1} = 2.3$



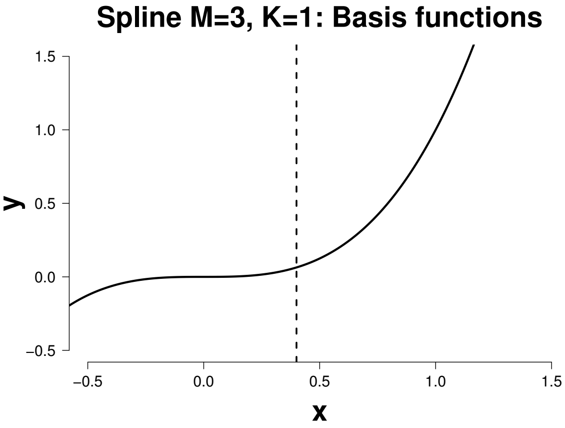
Example: Basis functions $M = 3$, $K = 1$

Knot at $\xi_1 = 0.4$
Modification: subtract $\theta_{2+1} = 2.3$ locally



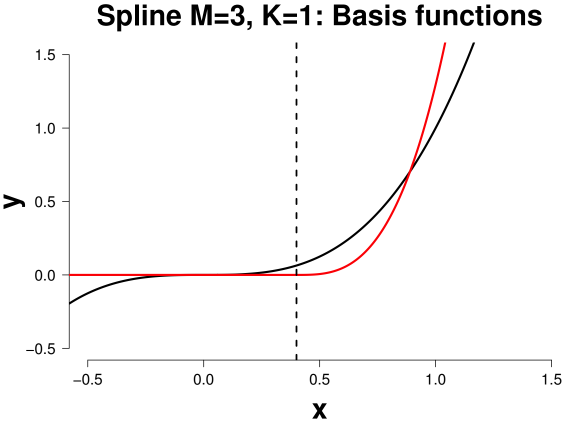
Example: Basis functions $M = 4$, $K = 1$

Knot at $\xi_1 = 0.4$
Global $\theta_3 = 1$



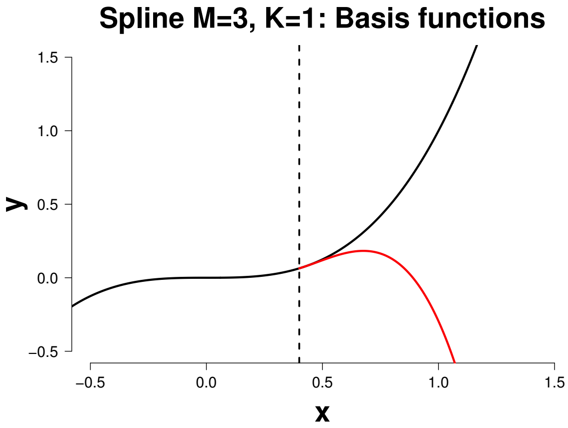
Example: Basis functions $M = 4$, $K = 1$

Knot at $\xi_1 = 0.4$
Global $\theta_3 = 1$ and local $\theta_{3+1} = 6$



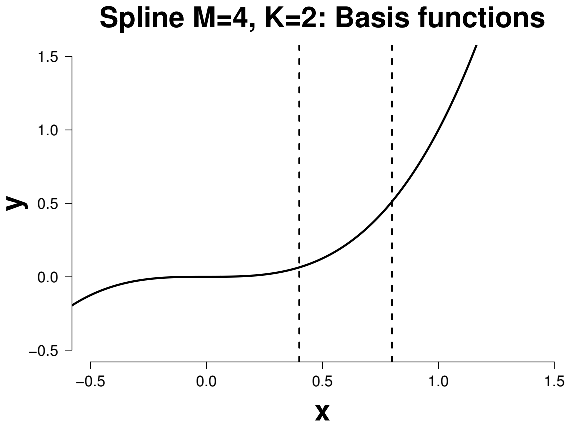
Example: Basis functions $M = 4$, $K = 1$

Knot at $\xi_1 = 0.4$
Modification: subtract $\theta_{3+1} = 6$ locally



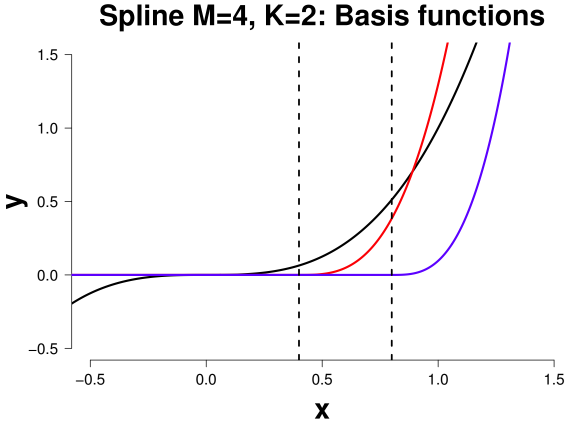
Example: Basis functions $M = 4$, $K = 2$

Knot at $\xi_1 = 0.4$ and $\xi_2 = 0.8$
Global $\theta_3 = 1$



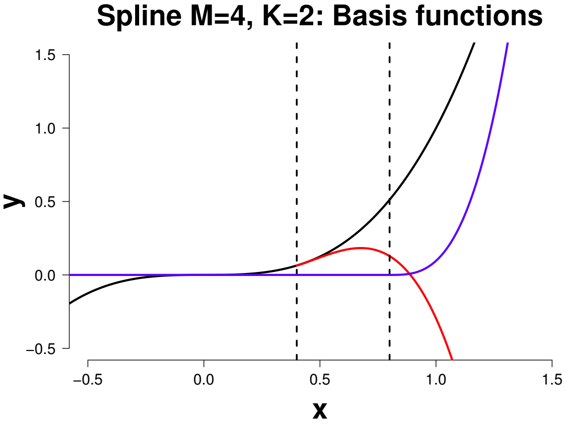
Example: Basis functions $M = 4$, $K = 2$

Knot at $\xi_1 = 0.4$ and $\xi_2 = 0.8$
Global $\theta_3 = 1$ and local $\theta_{3+1} = 6$, $\theta_{3+2} = 12$



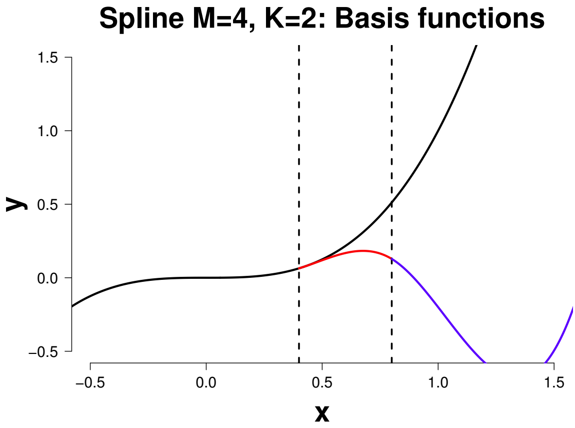
Example: Basis functions $M = 4$, $K = 2$

Knot at $\xi_1 = 0.4$ and $\xi_2 = 0.8$
Modification: subtract $\theta_{3+1} = 6$ “locally” from ξ_1 onwards



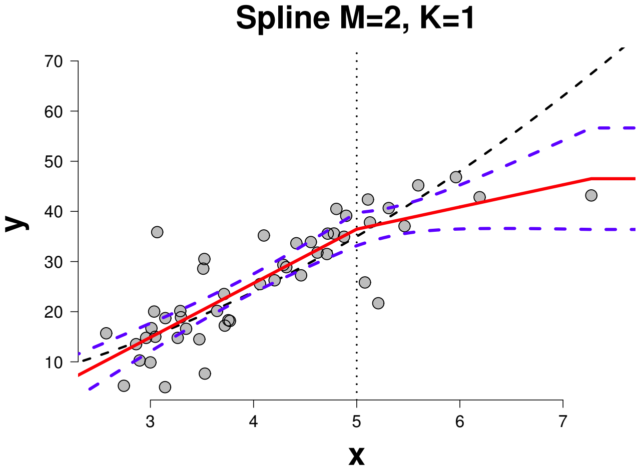
Example: Basis functions $M = 4$, $K = 2$

Knot at $\xi_1 = 0.4$ and $\xi_2 = 0.8$
Modification: add $\theta_{3+2} = 12$ “locally” from ξ_2 onwards



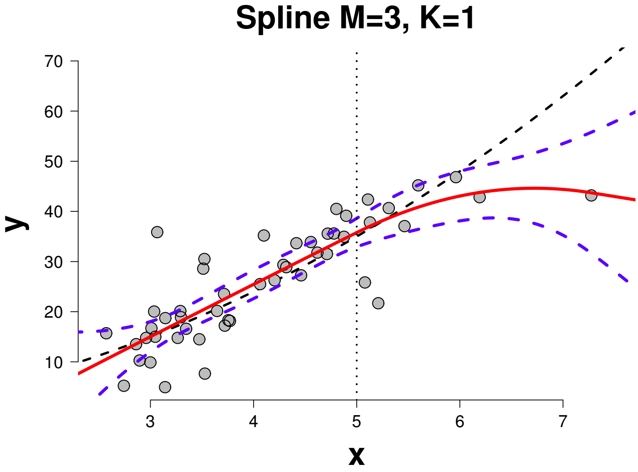
Example: Polynomials splines

$K = 1$ knot



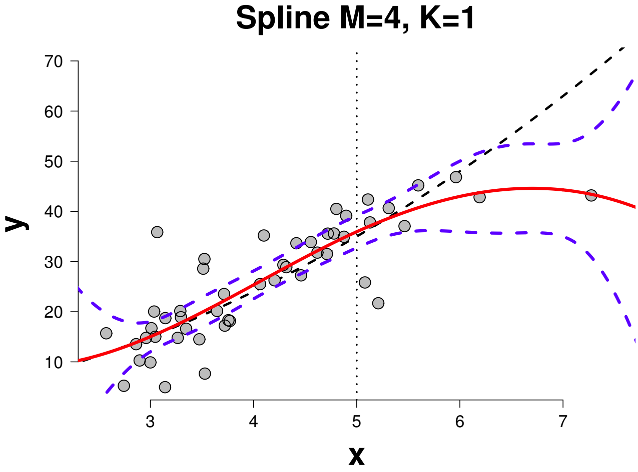
Example: Polynomials splines

$K = 1$ knot



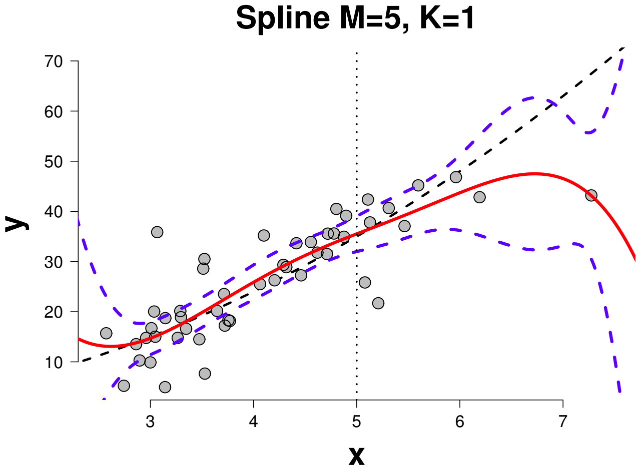
Example: Polynomials splines

$K = 1$ knot



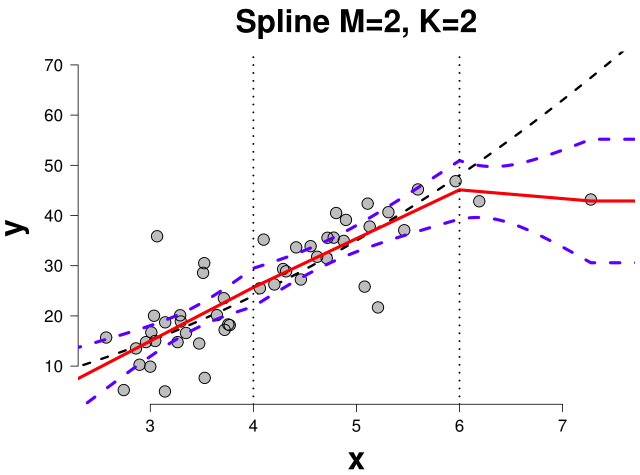
Example: Polynomials splines

$K = 1$ knot



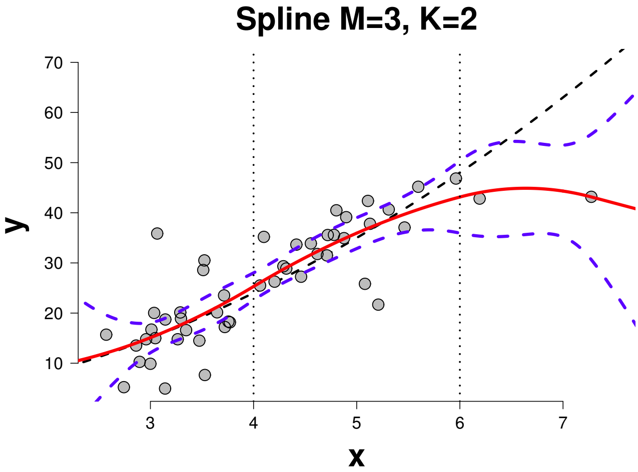
Example: Polynomials splines

Depends on *K* and where



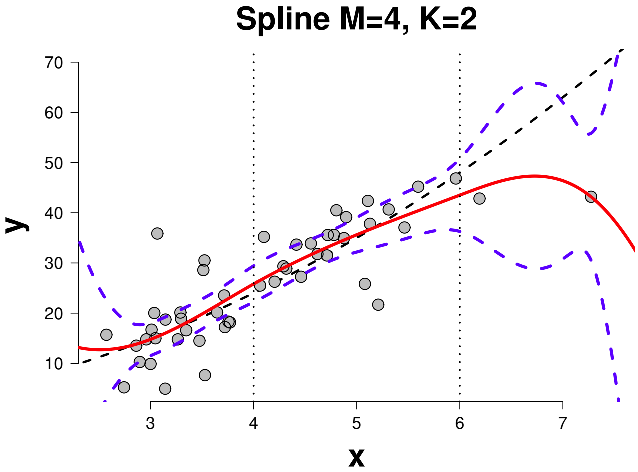
Example: Polynomials splines

Depends on *K* and where



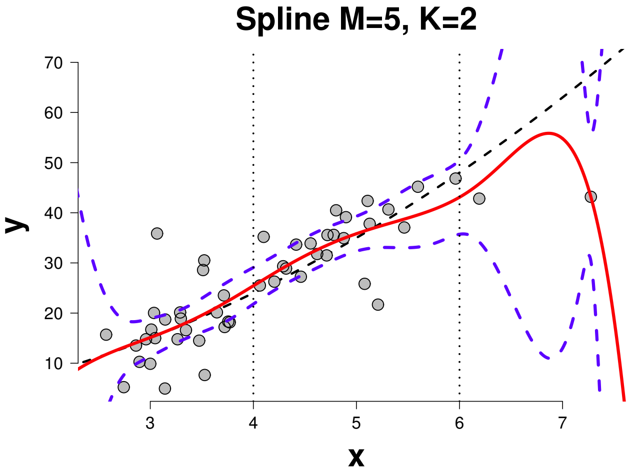
Example: Polynomials splines

Depends on *K* and where



Example: Polynomials splines

Depends on *K* and where



Natural splines

- Tail behaviour still bad. (high variance)

Natural splines

- Tail behaviour still bad. (high variance)
- Natural spline: Take polynomial of lower order $M/2 - 1$ for function past the end points

Natural splines

- Tail behaviour still bad. (high variance)
- Natural spline: Take polynomial of lower order $M/2 - 1$ for function past the end points
- Natural spline has K parameters! Whatever M may be

Natural splines

- Tail behaviour still bad. (high variance)
- Natural spline: Take polynomial of lower order $M/2 - 1$ for function past the end points
- Natural spline has K parameters! Whatever M may be
- "Solves": How many knots. (specify the number of parameters)

Natural splines

- Tail behaviour still bad. (high variance)
- Natural spline: Take polynomial of lower order $M/2 - 1$ for function past the end points
- Natural spline has K parameters! Whatever M may be
- "Solves": How many knots. (specify the number of parameters)
- Solution to where: use quantile of observed X

Natural splines

- Tail behaviour still bad. (high variance)
- Natural spline: Take polynomial of lower order $M/2 - 1$ for function past the end points
- Natural spline has K parameters! Whatever M may be
- "Solves": How many knots. (specify the number of parameters)
- Solution to where: use quantile of observed X
- Still have to choose the order M

Natural splines basis

Natural splines are polynomial splines that have lower order "tails" A natural spline of order *m* with *K* number of knots is has *K* number of basis functions:

- Global polynomial of order *m*

$$N_0(x) = x^0, \ldots, N_{m-3}(x) = x^{m-3} \tag{14}$$

Natural splines basis

Natural splines are polynomial splines that have lower order "tails" A natural spline of order m with K number of knots is has K number of basis functions:

- Global polynomial of order m

$$N_0(x) = x^0, \ldots, N_{m-3}(x) = x^{m-3} \tag{14}$$

Natural splines basis

Natural splines are polynomial splines that have lower order "tails" A natural spline of order m with K number of knots is has K number of basis functions:

- Global polynomial of order m

$$N_0(x) = x^0, \ldots, N_{m-3}(x) = x^{m-3} \tag{14}$$

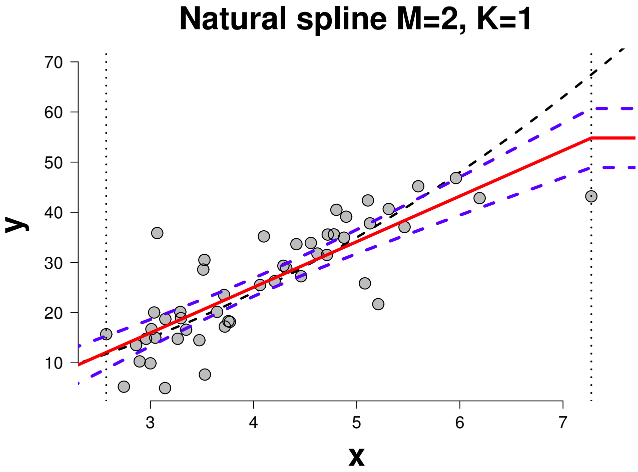
- Local modifications:

$$N_{k+2}(x) = d_k(x, \xi_k) - d_{K-1}(x, \xi_{K-1}) \text{ for } k = 1, \ldots, K - m + 1 \tag{15}$$

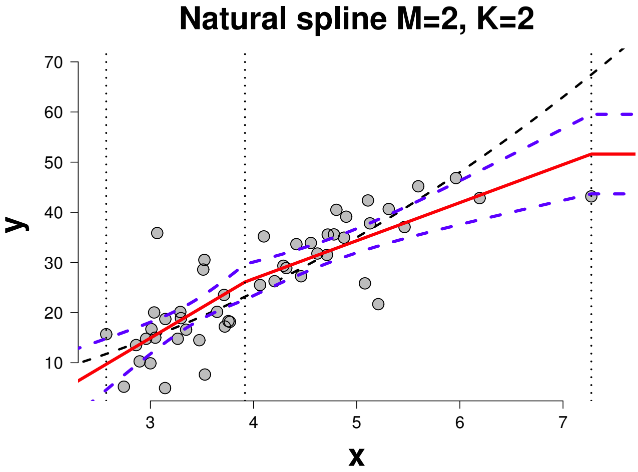
where

$$d_k(x, \xi_k) = \frac{(x - \xi_k)_+^3 - (x - \xi_K)_+^3}{\xi_K - \xi_k} \tag{16}$$

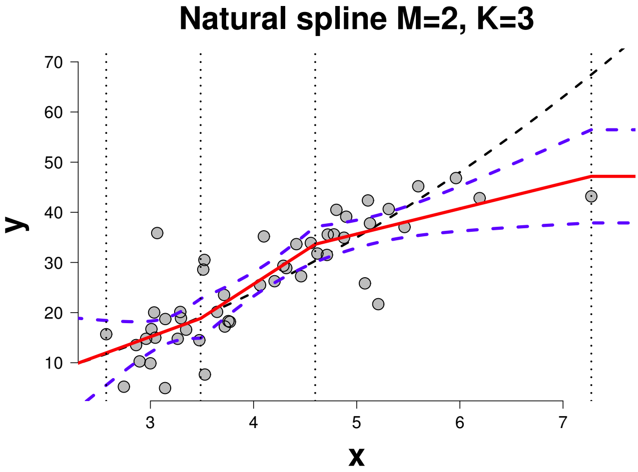
Example: Natural splines



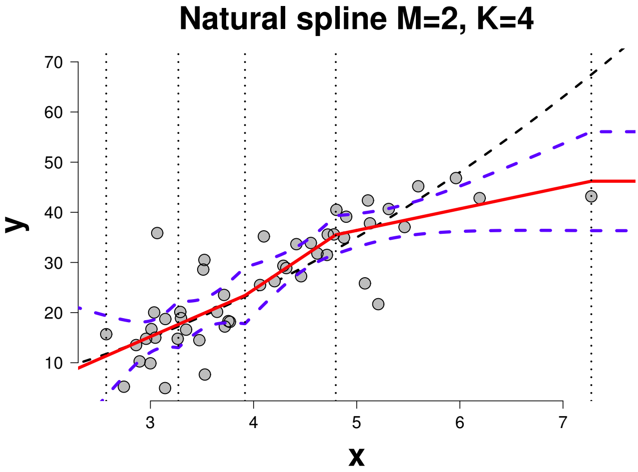
Example: Natural splines



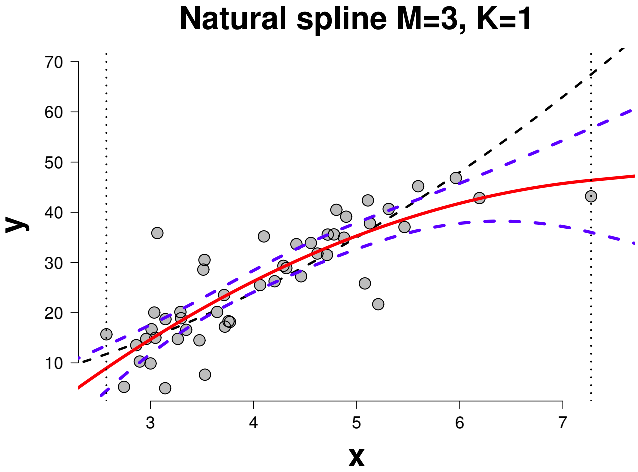
Example: Natural splines



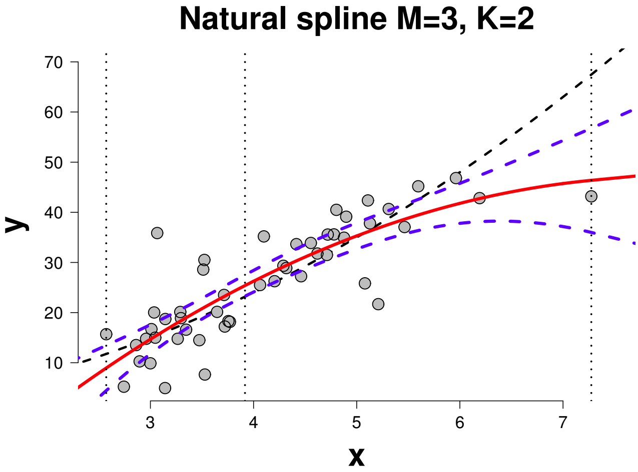
Example: Natural splines



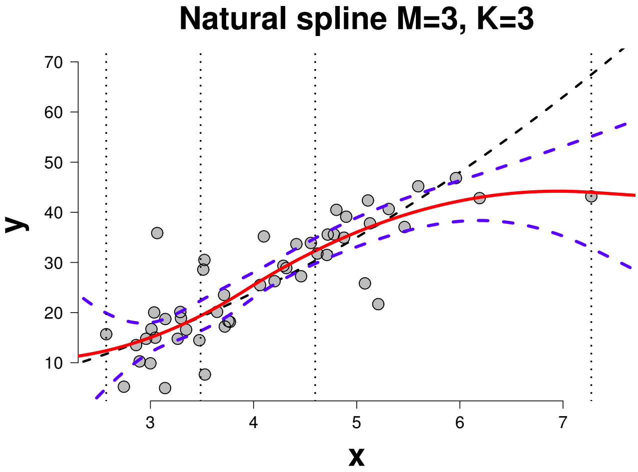
Example: Natural splines



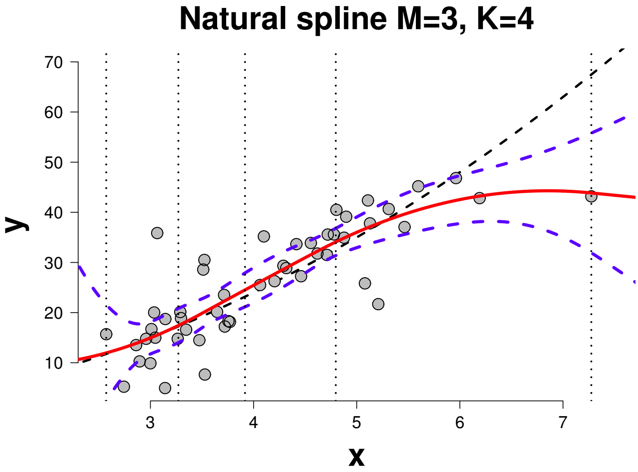
Example: Natural splines



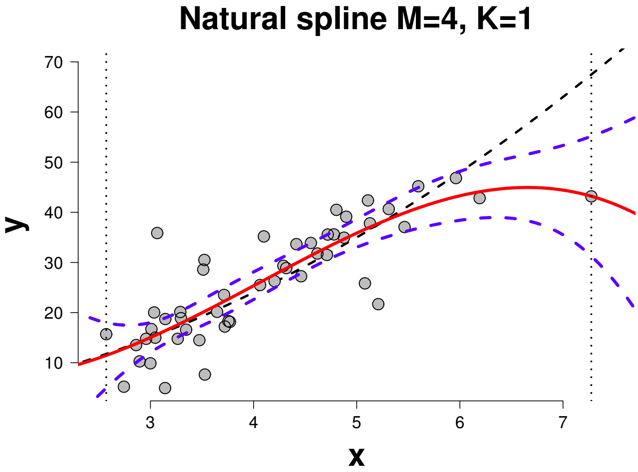
Example: Natural splines



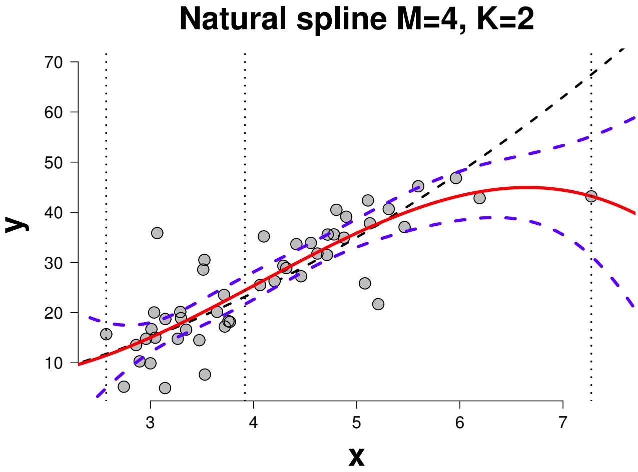
Example: Natural splines



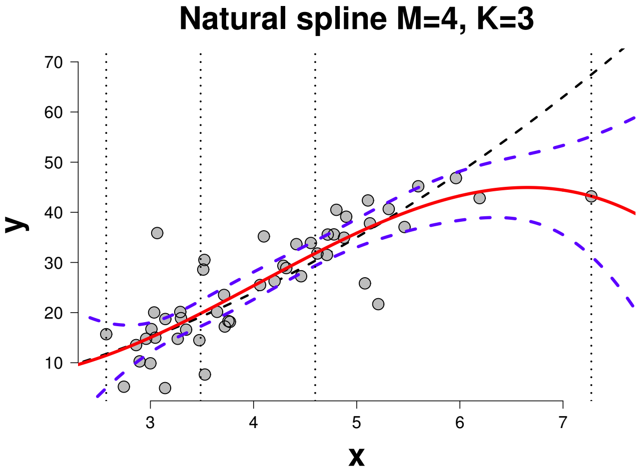
Example: Natural splines



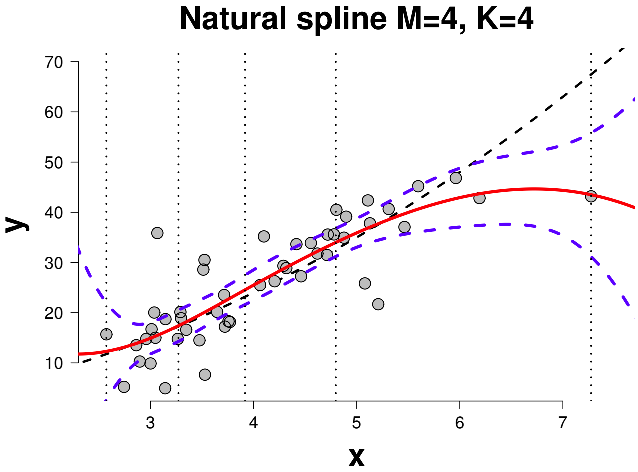
Example: Natural splines



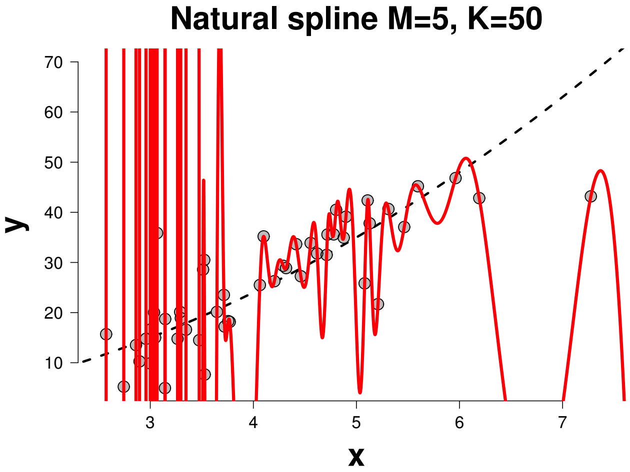
Example: Natural splines



Example: Natural splines



Example: Natural splines



Uniqueness and regularisation

- Previous: $p \ll n$ regime. Matrix "trick": small to big with limit at $p = n$ and note the interpolation.

Uniqueness and regularisation

- Previous: $p \ll n$ regime. Matrix "trick": small to big with limit at $p = n$ and note the interpolation.
- Problem: when $p > n$ then also have $Y = X(\theta_{(0)} + u) + \epsilon$, where $Xu = 0$. There are many u s.t. $Xu = 0$, thus, non-uniqueness.

Uniqueness and regularisation

- Previous: $p \ll n$ regime. Matrix "trick": small to big with limit at $p = n$ and note the interpolation.
- Problem: when $p > n$ then also have $Y = X(\theta_{(0)} + u) + \epsilon$, where $Xu = 0$. There are many u s.t. $Xu = 0$, thus, non-uniqueness.
- Solution: Choose the solution s.t. $\theta_{(0)} + u$ is small. In other words, instead of minimising $\sum_{i=1}^{n}(y_i - \tilde{f}(x_i))^2$ minimise the following instead

$$\hat{f}(x) = \underset{\tilde{f}\in\mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^{n}(y_i - \tilde{f}(x_i))^2 + \lambda\,\text{penalty}(\tilde{f}). \tag{17}$$

for some fixed $\lambda > 0$.

Uniqueness and regularisation

- Previous: $p \ll n$ regime. Matrix "trick": small to big with limit at $p = n$ and note the interpolation.
- Problem: when $p > n$ then also have $Y = X(\theta_{(0)} + u) + \epsilon$, where $Xu = 0$. There are many u s.t. $Xu = 0$, thus, non-uniqueness.
- Solution: Choose the solution s.t. $\theta_{(0)} + u$ is small. In other words, instead of minimising $\sum_{i=1}^{n}(y_i - \tilde{f}(x_i))^2$ minimise the following instead

$$\hat{f}(x) = \underset{\tilde{f} \in \mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^{n}(y_i - \tilde{f}(x_i))^2 + \lambda \, \text{penalty}(\tilde{f}). \qquad (17)$$

  for some fixed $\lambda > 0$.
- Example: Lasso/ridge/elastic nets. Here: smoothing splines (directly on the function, not on the parameters).

Smoothing spline set-up

- Big to small, start with $n \ll p$ and regularise:

$$\hat{f}(x) = \underset{\tilde{f} \in \mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \tilde{f}(x_i))^2 + \lambda \int [f^{(m/2)}(x)]^2 \mathrm{d}x \tag{18}$$

Smoothing spline set-up

- Big to small, start with $n \ll p$ and regularise:

$$\hat{f}(x) = \underset{\tilde{f} \in \mathcal{F}}{\text{argmin}} \sum_{i=1}^{n} (y_i - \tilde{f}(x_i))^2 + \lambda \int [f^{(m/2)}(x)]^2 \mathrm{d}x \quad (18)$$

- Candidate set: $\mathcal{F}_\lambda$ are all functions that have a bounded squared $m/2$ derivative.

Smoothing spline set-up

- Big to small, start with $n \ll p$ and regularise:

$$\hat{f}(x) = \underset{\tilde{f} \in \mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \tilde{f}(x_i))^2 + \lambda \int [f^{(m/2)}(x)]^2 \mathrm{d}x \quad (18)$$

- Candidate set: $\mathcal{F}_\lambda$ are all functions that have a bounded squared $m/2$ derivative.
- Problem: Infinite-dimensional optimisation problem over all functions f

Smoothing spline set-up

- Big to small, start with $n \ll p$ and regularise:

$$\hat{f}(x) = \underset{\tilde{f} \in \mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \tilde{f}(x_i))^2 + \lambda \int [f^{(m/2)}(x)]^2 \mathrm{d}x \quad (18)$$

- Candidate set: $\mathcal{F}_\lambda$ are all functions that have a bounded squared $m/2$ derivative.
- Problem: Infinite-dimensional optimisation problem over all functions f
- Remarkable: There is unique minimiser: an m order natural spline with knots at the observations $x_1, \ldots, x_n$.

Smoothing spline set-up

- Big to small, start with $n \ll p$ and regularise:

$$\hat{f}(x) = \underset{\tilde{f} \in \mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \tilde{f}(x_i))^2 + \lambda \int [f^{(m/2)}(x)]^2 \mathrm{d}x \quad (18)$$

- Candidate set: $\mathcal{F}_\lambda$ are all functions that have a bounded squared $m/2$ derivative.
- Problem: Infinite-dimensional optimisation problem over all functions f
- Remarkable: There is unique minimiser: an m order natural spline with knots at the observations $x_1, \ldots, x_n$.
- Note: This spline is sum of finite number of basis functions (i.e., $n = K$ parameters). These basis functions are decided by the data $x_1, \ldots, x_n$.

Smoothing splines basis

A smoothing spline has basis functions decided by the data $x_1, \ldots, x_n$

- Global polynomial of order m

$$N_0(x) = x^0, \ldots, N_{m-3}(x) = x^{m-3} \tag{19}$$

Smoothing splines basis

A smoothing spline has basis functions decided by the data $x_1, \ldots, x_n$

- Global polynomial of order m

$$N_0(x) = x^0, \ldots, N_{m-3}(x) = x^{m-3} \tag{19}$$

- Local modifications:

$$N_{i+2}(x) = d_i(x, x_i) - d_{n-1}(x, x_{n-1}) \text{ for } i = 1, \ldots, n-m+1 \tag{20}$$

where

$$d_k(x, x_i) = \frac{(x - x_i)_+^3 - (x - x_n)_+^3}{x_n - x_i} \tag{21}$$

Return of the “matrix trick”

Thus the candidate solution is of the form

$$\tilde{f}(x) = \sum_{i=1}^{n} N_i(x)\theta_j \tag{22}$$

hence

$$MSE(\tilde{f}) = (y - N\theta)^T(y - N\theta) + \lambda\theta^T\Omega_n\theta, \tag{23}$$

where N is the design matrix $\{N_{ij}\} = N_j(x_i)$ and

$$\{\Omega_n\}_{ji} = \int N_j^{(m/2)}(x)N_i^{(m/2)}(x)\mathrm{d}x \tag{24}$$

Return of the "matrix trick"

Thus the candidate solution is of the form

$$\tilde{f}(x) = \sum_{i=1}^{n} N_i(x)\theta_j \tag{22}$$

hence

$$MSE(\tilde{f}) = (y - N\theta)^T(y - N\theta) + \lambda\theta^T\Omega_n\theta, \tag{23}$$

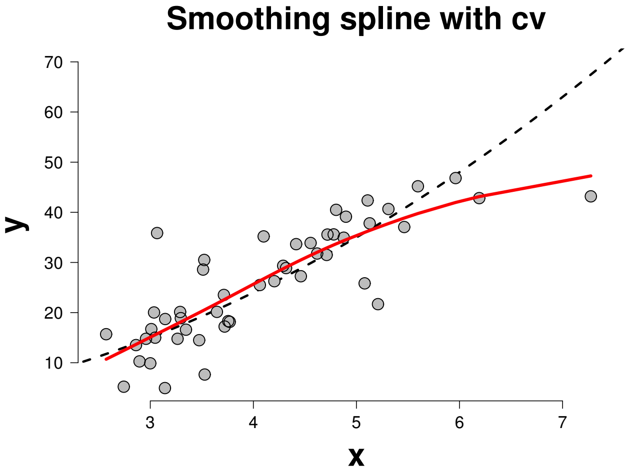
where N is the design matrix $\{N_{ij}\} = N_j(x_i)$ and

$$\{\Omega_n\}_{ji} = \int N_j^{(m/2)}(x)N_i^{(m/2)}(x)\mathrm{d}x \tag{24}$$

Minimisation

$$\hat{\theta} = (N^TN + \lambda\Omega_n)^{-1}N^Ty \tag{25}$$

Example: Smoothing spline with cross validation



Choosing the λ and degrees of freedom

- Recall: $n \ll p$ regime solution: Natural splines $\hat{f}(X) = X\hat{\theta}$ with K knots:

$$\hat{f}(X) = \underbrace{X(X^T X)^{-1} X^T}_{H_\xi} y \tag{26}$$

where H_ξ is a symmetric, positive semidefinite matrix.

Choosing the λ and degrees of freedom

- Recall: $n \ll p$ regime solution: Natural splines $\hat{f}(X) = X\hat{\theta}$ with K knots:

$$\hat{f}(X) = \underbrace{X(X^TX)^{-1}X^T}_{H_\xi} y \tag{26}$$

where H_ξ is a symmetric, positive semidefinite matrix.

- Compare: Smoothing spline

$$\hat{f}(N) = \underbrace{N(N^TN + \lambda\Omega_n)^{-1}N^T}_{S_\lambda} y \tag{27}$$

where S_λ is a symmetric, positive semidefinite matrix.

Choosing the λ and degrees of freedom

- Recall: $n \ll p$ regime solution: Natural splines $\hat{f}(X) = X\hat{\theta}$ with K knots:

$$\hat{f}(X) = \underbrace{X(X^TX)^{-1}X^T}_{H_\xi} y \tag{26}$$

  where H_ξ is a symmetric, positive semidefinite matrix.
- Compare: Smoothing spline

$$\hat{f}(N) = \underbrace{N(N^TN + \lambda\Omega_n)^{-1}N^T}_{S_\lambda} y \tag{27}$$

  where S_λ is a symmetric, positive semidefinite matrix.
- $\mathrm{trace}(H_\xi) = K$, the dimension of the space H_ξ projects to

Choosing the λ and degrees of freedom

- Recall: $n \ll p$ regime solution: Natural splines $\hat{f}(X) = X\hat{\theta}$ with K knots:

$$\hat{f}(X) = \underbrace{X(X^T X)^{-1} X^T}_{H_\xi} y \tag{26}$$

  where H_ξ is a symmetric, positive semidefinite matrix.
- Compare: Smoothing spline

$$\hat{f}(N) = \underbrace{N(N^T N + \lambda\Omega_n)^{-1} N^T}_{S_\lambda} y \tag{27}$$

  where S_λ is a symmetric, positive semidefinite matrix.
- $\text{trace}(H_\xi) = K$, the dimension of the space H_ξ projects to
- Take $df = \text{trace}(S_\lambda)$. Note as $\lambda \to \infty$ this lowers the dimension.

Further relationships

- Projections
- RKHS
- Gaussian processes
- Bayesian nonparametric regression