Scenario $(\underline{x}_1, y_1) \dots (\underline{x}_n, y_n)$ TS

Want to generate prediction rule $f(\underline{x})$ that predicts Y for new query $\underline{x}$

We know: optimal rule is $f(\underline{x}) = E(Y | \underline{x})$

How do we estimate $E(Y|x)$ from TS?

Consider single predictor case

Option 1: Kernel smoothing

Option 2: Expansion based methods

Basic idea of expansion based prediction methods:

- Choose "dictionary" of basis functions $B_1(x) \dots B_K(x)$ for which (hopefully)

  $$E(Y|x) \approx \sum_{j=1}^{K} a_j B_j(x)$$

- Find $\hat{\underline{a}}$ that minimizes the resubstution error:

  sum of squared prediction errors for training sample

  $$\hat{\underline{a}} = \underset{\underline{a}}{\operatorname{argmin}} \| \underline{y} - X \underline{a} \|^2$$

Where $X_{ij} = B_j(i)$: X $n \times K$

i-th of $X = (B_1(x_i) \dots B_K(x_i))$

j-th column of $X = (B_j(x_1) \dots B_j(x_n))^T$

Examples for dictionaries

① Polynomials

$B_i(x) = x^i \quad i = 1 \dots K-1$

$B_K(x) = 1$

$\sum a_j B_j(x)$ is polynomial of degree $K-1$
order K

② Piecewise constant functions

Pick "Knot positions" $c_1 \leq c_2 \cdots \leq c_{K-1}$

$c_0 = -\infty \qquad c_K = +\infty$

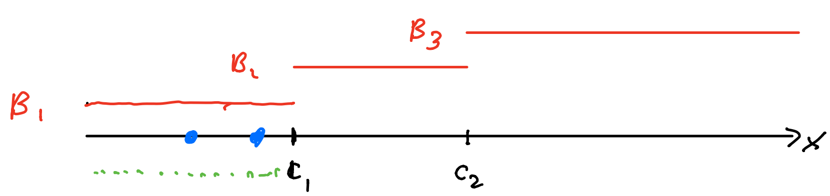
$$B_i(x) = I((c_{i-1}, c_i], x)$$

$\Vert$

1 if $x \in [c_{i-1}, c_i]$

0 otherwise

Do we need a column of $(1 \dots 1)$?



$\Rightarrow \sum B_j(x) = 1 \quad \forall x$

Form $X = \begin{bmatrix} B_1(x_1) & B_2(x_1), & B_2(x_1) \\ & \vdots & \\ B_1(x_n) & B_2(x_n) & B_2(x_n) \end{bmatrix}$ each row adds up to 1 adding a column of all "1"s is unnecessy and would make X rank deficient

Obvious chooce for c_1, c_2:

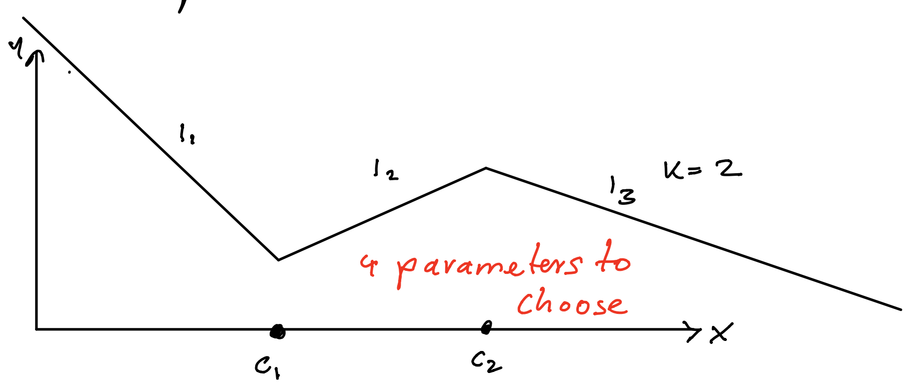
choose c_1 = 33rd percentile of $x_1 \ldots x_n$

c_2 = 66 th " $x_1 \ldots x_n$

$\hat{a}_1$ = mean of the y_i for which $x_i \in (-\infty, c_1]$

③ Continuous piecewise linear functions ("order 2" splines)

Choose knot positions $c_1 \le c_2 \cdots \le c_K$



K knots $\Rightarrow$ how many parameters are there?

$(K+1)$ intervals $\Rightarrow$ $2(K+1)$ coefficients of the $(K+1)$ line segments

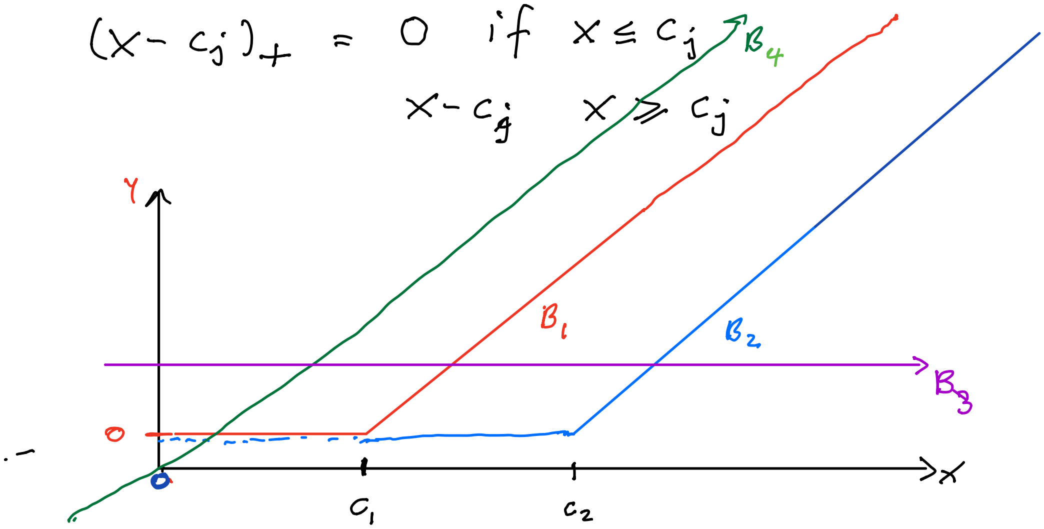
K constraints

$2(K+1) - K = K+2$ free parameters

Basis functions

$$B_j(x) = (x - c_j)_+$$

$$(x - c_j)_+ = \begin{cases} 0 & \text{if } x \le c_j \\ x - c_j & x \ge c_j \end{cases}$$



$B_j(x) = (x - c_j)_+ \quad j = 1 \dots K$

$B_{K+1}(x) = 1(x)$ constant ~~and~~ function

$B_{K+2}(x) = x$

Note: Any linear combination $\sum a_j B_j(x)$ is a continuous piecewise linear function

Conclusion: Any piecewise linear continuous function with knots $c_1 \dots c_K$ can be written as a linear combination of $B_1(x) \dots B_{K+2}(x)$.

4 Cubic splines

piecewise 3rd degree polynomials, twice continuously differentiable

For K knots $K+1$ polynomial pieces $\Rightarrow$

$4(K+1)$ coefficients ond $3K$ constraints

$\Rightarrow 4(K+1) - 3K$ degrees of freedom

$= K+4$

choice of basis functions

$B_j(x) = (x - c_j)_+^3 \quad j = 1 \dots K$

$B_{K+1}(x) = 1$

$B_{K+2}(x) = x$

$B_{K+3}(x) = x^2$

$B_{K+4}(x) = x^3$

Choices :

- which dictionary
- How to choose # of Knots
- How to choose not positions
- How to choose the coefficients

Finding expansion based rules with good predictive performance

Suppose we have chosen basis functions $B_1(x) \dots B_u(x)$

(In the case of order 2 splines, we have chosen the # of Knots and the Knot positions)

Looking at prediction rules of the form

$$f(x) = \sum_{j=1}^{K} a_j B_j(x)$$

How to find $\hat{a}_1 \ldots \hat{a}_K = \underline{\hat{a}}$

$$\underline{\hat{a}} = \underset{\underline{a}}{\operatorname{argmin}} \| y - X\underline{a} \|^2$$

$$X_{ij} = B_j(x_i)$$

Question: Can we do better?

In general "yes"

Why not piecewise quadratic.

K knots $K+1$ segments

$3(K+1)$ parameters

→ continuity K constraints

derivative cont K constraint

$df = 3K + 3 - 2K \quad K+3$

$$B_i = (x - c_i)_+^2 \qquad i = 1 \ldots K$$

$$B_{K+1} = 1$$

$$B_{K+2} = x$$

$$B_{K+3} = x^2$$