B-splines

- A spline function of order m and knots $\xi_1, \ldots, \xi_k$ is a piecewise polynomial that can be expressed as a linear combination of the functions $x^0, \ldots, x^{m-1}, (x-\xi_1)_+^{m-1}, \ldots, (x-\xi_k)_+^{m-1}$, where

$$(x-\xi_i)_+^{m-1} = \begin{cases} (x-\xi_i)^{m-1} & \text{if } x \geq \xi_i; \\ 0 & \text{otherwise,} \end{cases}$$

  for $i = 1, \ldots, k$.

- Plot $f(x) = (x-0.5)_+^2$ for $x \in [0,1]$.

```
f <- function(x, a=0.5){
  ans <- (x-a)^2
  ans[x<a] <- 0
 return(ans)
}
curve(f,0,1)
```

- A spline function of order m and inner knots $\xi_1, \ldots, \xi_k$ on an interval $[a,b]$ can be expressed as a linear combination of the B-spline basis functions on $[a,b]$ of order m with inner knots $\xi_1, \ldots, \xi_k$.

- Each B-spline basis function of order m is characterized by a set of knots $y_1, \ldots, y_{m+1}$ (arranged in ascending order), which is denoted by $N(\cdot|y_1, \ldots, y_{m+1})$. To compute B-spline basis functions, the following recursive formulas are used.

$$\begin{aligned} &N(x|y_1, \ldots, y_{m+1}) \\ &= \frac{x-y_1}{y_m-y_1} N(x|y_1, \ldots, y_m) + \frac{y_{m+1}-x}{y_{m+1}-y_2} N(x|y_2, \ldots, y_{m+1}). \end{aligned} \tag{1}$$

$$N(x|y_1, y_2) = \begin{cases} 1 & \text{if } x \in [y_1, y_2); \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

$$N(x|\underbrace{y_1, \ldots, y_1}_{m \text{ times}}, y_2) = \begin{cases} (y_2-x)^{m-1}/(y_2-y_1)^{m-1} & \text{if } x \in [y_1, y_2); \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

$$N(x|y_1, \underbrace{y_2, \ldots, y_2}_{m \text{ times}}) = \begin{cases} (x-y_1)^{m-1}/(y_2-y_1)^{m-1} & \text{if } x \in [y_1, y_2); \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

- B-spline basis functions on $[a,b]$ of order m with knots ξ_1, …, ξ_k. Let y_1, …, y_{2m+k} be the sequence

$$\underbrace{a,\ldots,a}_{m \text{ times}},\xi_1,\ldots,\xi_k,\underbrace{b,\ldots,b}_{m \text{ times}},$$

  then the $(m+k)$ functions $N(\cdot|y_1,\ldots,y_{m+1}),\ldots,N(\cdot|y_{m+k},\ldots,y_{2m+k})$ are the B-spline basis functions on $[a,b]$ of order m with knots ξ_1, …, ξ_k.

- Note. The B-spline basis functions sum up to 1 and each one takes values in $[0,1]$.

- The function `splineDesign` in R Package `splines` can be used for computing a single B-spline basis function $N(\cdot|y_1,\ldots,y_{m+1})$.

  – Suppose that `y`$= (y_1,\ldots,y_{m+1})$ and `x` $= (x_1,\ldots,x_n)$ are two vectors in R, then

    ```
    splineDesign(y, x, ord=length(y)-1, outer.ok=TRUE)[,1]
    ```

    gives the vector $(N(x_1|y_1,\ldots,y_{m+1}),\ \ldots,\ N(x_n|y_1,\ldots,y_{m+1}))$.

- Example 1. Check whether `splineDesign` gives

$$N(x|0,0,0,1) = \begin{cases} (1-x)^{3-1}/(1-0)^{3-1} = (1-x)^2 & \text{if } x \in [0,1); \\ 0 & \text{otherwise.} \end{cases}$$

  R commands:

```
require("splines")
y <- c(0,0,0,1)
f <- function(x){
 return(splineDesign(y, x, ord=length(y)-1, outer.ok=TRUE)[,1])
}
curve(f,0,1)

n0001 <- function(x){ (1-x)^2 }
curve(n0001, 0, 1, add=T, col=2)  #Plot the N(|0,0,0,1) function on [0,1]
```

- The function `bs` in R Package `splines` can be used for computing the $(m+k)$ B-spline functions on $[a,b]$ of order m with knots ξ_1, …, ξ_k.

– Suppose that `knots`$= (\xi_1, \ldots, \xi_k)$ and `x` $= (x_1, \ldots, x_n)$ are vectors in R, then B-spline basis functions on [`a`,`b`] of order `m` with knots $\xi_1, \ldots, \xi_k$ evaluated at `x` form a $n \times$ (`m`$+k$) matrix X, where the j-th column of X is the j-th B-spline basis function evaluated at `x`. The matrix X can be computed using the R command

```
bs(x, knots=knots, deg=m-1, Boundary.knots=c(a,b), intercept=TRUE)
```

- Example 2. Compute the B-spline basis functions on $[0, 1]$ of order three with knots 0.1 and 0.2 using `bs` and `splineDesign`. Plot the five B-spline basis functions.

```
x <- (1:1000)/1001
knotlist=c(0.1, 0.2)
ord=3
knot_all=c(rep(0,ord), knotlist, rep(1,ord))
nb = length(knotlist)+ord
plot(x,x,type="n",ylab="")
s=0
for ( i in 1:nb){
  y = knot_all[i:(i+ord)]
  f <- function(x){
   n <- length(x)
   return(splineDesign(y, x, ord=length(y)-1, outer.ok=T)[,1])
  }
  f1 <- function(x){
    bx=bs(x, deg=ord-1, knots=knotlist, Boundary.knots=c(0,1), intercept=T)
    return(bx[,i])
  }
  lines(x,f(x), type="l");  lines(x, f1(x), col=i+1)
  s=s+sum(abs(f(x)-f1(x)))
}
s
```

- Spline functions can approximated smooth functions well if the number of knots are large enough.

- Exercise 1. Write a function with input x and $y = (y_1, \ldots, y_{m+1})$ (assuming $y_1 \le \ldots \le y_{m+1}$ and $m \in \{1, 2\}$) and output $N(x|m, y_1, \ldots, y_{m+1})$ using the recursive formulas in (1) – (4). Use your function to compute $N(x|2, y_1, y_2, y_3)$ at $x =$`seq(0, 1, length=11)` for $(y_1, y_2, y_3) = (0, 0, 0.1)$ and $(y_1, y_2, y_3) = (0, 0.1, 0.2)$. Compare the results with those obtained using `splineDesign`.

- Exercise 2. Let B_1, …, B_{m+k} be the B-spline basis functions on $[0, 1]$ of order m with knots ξ_1, …, ξ_k. Check whether each of $B_1(x)$, …, $B_{m+k}(x)$ can be expressed as linear combinations of the functions 1, x, …, x^{m-1}, $(x - \xi_1)_+^{m-1}$, …, $(x - \xi_k)_+^{m-1}$ for $x =$`(1:1000)/1001`, $m = 4$, $k = 3$, and $(\xi_1, \ldots, \xi_3) =$`(1:3)/4`.

- Exercise 3. Let $f(x) = x \sin(20x)$ for $x \in [0, 1]$. Suppose that $n = 1000$, $(X_1, \ldots, X_n) =$ `seq(0, 1, length=n)`, and $Y_i = f(X_i)$ for $i = 1$, …, n.

  (a) Find the ISE of estimating f based on (X_1, Y_1), …, (X_n, Y_n) by approximating f using the best linear combination of B-spline basis functions on $[0, 1]$ of order 4 with k equally spaced knots (`(1:k)/(k+1)`) for $k = 1$, 2, …, 7. Which k gives the best ISE?

  (b) Consider estimating f based on (X_1, Y_1), …, (X_n, Y_n) by approximating f using the best linear combination of 1, x, …, x^m. Let m_0 be the smallest m such that the resulting ISE is smaller than the best ISE found in Part (a). Find m_0.

Exercise 4. Define a function `f` in R:

```
f0 <- function(x){
 ans <- x*sin(20*x)
 ans[x<0] <- 0
 return(ans)
}

f <- function(x){ f0(2*(x-0.5))}
curve(f,0,1)
```

It is expected that we can approximate `f` well using a cubic spline (order 4) with knots `(1:13)/14`.

(a) Generate data from a nonparametric regression model as follows:

```
n <- 1000
x <- seq(0,1,length=n)
y <- f(x) + rnorm(n, sd=0.02)
```

Fit a cubic spline to the data with knots `(1:13)/14` using truncated power basis functions. Remove the insignificant knots using backward elimination. How many knots are left in the model?

(b) Let `knots0` be the remaining knots from Part (a). Generate data from a nonparametric regression model as follows:

```
n <- 1000
x <- seq(0,1,length=n)
y <- f(x)
```

Fit a cubic spline to the data with knots `=knots0` using B-spline basis functions on $[0, 1]$. Find the ISE. Denote this ISE by ISE.SP. Recall that we can also approximate `f` well on $[0, 0.5]$ using a polynomial of high degree. Is it possible to use a polynomial of degree 10 to approximate `f` so that the ISE is smaller or equal to ISE.SP?

- Exercise 5. Let $f(x) = x \sin(20x)$ for $x \in [0, 1]$. Suppose that $n = 1000$, $(X_1, \ldots, X_n) =$ `seq(0, 1, length=n)`, and $Y_i = f(X_i)+\varepsilon_i$ for $i = 1, \ldots, n$, where ε_i's are IID $N(0, \sigma^2)$ variables with $\sigma = 0.2$. Consider estimating f based on $(X_1, Y_1), \ldots, (X_n, Y_n)$ by approximating f using

  (i) B-spline basis functions with order 4 and k equally spaced knots and

  (ii) polynomial basis functions $1, x, \ldots, x^m$,

  where k and m are chosen using cross-validation. The selection ranges for k and m are $\{1, 2, \ldots, 7\}$ and $\{1, 2, \ldots, 12\}$. Find the IMSEs for (i) and (ii) based on 10000 simulations. Use `set.seed` to make sure that the simulation data for (i) and (ii) are the same.