

Density estimation based on basis function approximation

- Suppose that X_1, \dots, X_n are IID data with density function f . The problem of interest is to estimate f based on X_1, \dots, X_n based on basis function approximation.
- Suppose that B_1, \dots, B_m is a set of basis functions such that a smooth function can be approximated well by $\sum_{j=1}^m a_j B_j$ for some (a_1, \dots, a_m) .
- A quick way to estimate f with $f \approx \sum_{j=1}^m a_j B_j$ is to use the method of moment approach to estimate $a = (a_1, \dots, a_m)$.

– Idea: for $k = 1, \dots, m$,

$$\frac{1}{n} \sum_{i=1}^n B_k(X_i) \approx \int f(x) B_k(x) dx \approx \sum_{j=1}^m a_j \int B_j(x) B_k(x) dx$$

Solve $a = (a_1, \dots, a_m)$ so that

$$\frac{1}{n} \sum_{i=1}^n B_k(X_i) = \sum_{j=1}^m a_j \int B_j(x) B_k(x) dx$$

- Example 1. Perform density estimation using spline basis approximation and method of moments. The data are IID data generated from a mixture distribution with probability density $f = 0.5f_1 + 0.5f_2$, where

$$f_1(x) = \begin{cases} \frac{1}{c_1 \sqrt{2\pi\sigma_1^2}} e^{-(x-\mu_1)^2/2\sigma_1^2}; & \text{if } x \in (0, 1); \\ 0 & \text{otherwise,} \end{cases}$$

$$\mu_1 = 0.2, \sigma_1 = 0.1,$$

$$f_2(x) = \begin{cases} \frac{1}{c_2 \sqrt{2\pi\sigma_2^2}} e^{-(x-\mu_2)^2/2\sigma_2^2}; & \text{if } x \in (0, 1); \\ 0 & \text{otherwise,} \end{cases}$$

$\mu_2 = 0.7, \sigma_2 = 0.2$, and c_1 and c_2 are constants so that $\int f_1(x) dx = 1 = \int f_2(x) dx$. For spline approximation, use cubic spline basis functions with inner knots $1/5, 2/5, 3/5, 4/5$ and boundary knots $0, 1$. Let B_1, \dots, B_8 denote those basis functions. The true density f is approximated using linear combination of B_1, \dots, B_8 .

```
###generate data of size 1000 (stored in x) from density f
set.seed(1)
mu1=.2
mu2=.7
n <- 1000
m <- n*10
z <- rnorm(m,mean=mu1,sd=.1); x <- z[(z>0)&(z<1)]
x <- x[1:n]
z <- rnorm(m,mean=mu2,sd=.2); x2 <- z[(z>0)&(z<1)]
x2 <- x2[1:n]
z <- sample(0:1, size = n, replace=T)
```

```

x[z==1] <- x2[z==1]

#### compute the matrix whose (i,j)th element is the integral of B_iB_j
require("splines")
knotlist <- (1:4)/5
nb <- length(knotlist)+4
M <- matrix(0, nb, nb)
for (i in 1:nb){
  for (j in i:nb){
    tem <- function(u){
      bx <- bs(u, knots = knotlist, Boundary.knots = c(0,1), intercept=T)
      return( bx[,i]*bx[,j])
    }
    M[i,j] <- integrate(tem, 0, 1)$value
    if (j > i) { M[j,i] <- M[i,j] }
  }
}

#### compute fhat, the estimator of f using method of moments
moments <- apply(bs(x, knots = knotlist, Boundary.knots=c(0,1), intercept=T), 2, mean)
ahat <- solve( M, moments)
fhat <- function(u){
  ans <- bs(u, knots = knotlist, Boundary.knots = c(0,1), intercept=T) %*% ahat
  return( as.numeric(ans) )
}

##### compare fhat with the true density f
k0=pnorm(1, mean=mu1,sd=.1)-pnorm(0,mean=mu1,sd=.1)
k1=pnorm(1, mean=mu2,sd=.2)-pnorm(0,mean=mu2,sd=.2)
f <- function(x){
  ans <- 0.5*dnorm(x, mean=mu1, sd=.1)/k0 + 0.5 *dnorm(x, mean=mu2, sd=.2)/k1
  ans[x>1]=0
  ans[x<0]=0
  return(ans)
}
curve(f,0,1)
curve(fhat,0,1, add=T, col=2)
## compute ISE
tem <- function(u){ (fhat(u)-f(u))^2 }
integrate(tem,0,1)
#0.006720843

##### obtain a normalized version
k2 <- integrate(fhat,0,1)$value
fhat1 <- function(u){ fhat(u)/k2 }
curve(fhat1,0,1, add=T, col=3)

##### Check spline approximation accuracy using the given basis functions

```

```

x0 <- (1:1000)/1001
y1 <- f(x0)
bx <- bs(x0,knots=knotlist, Boundary.knots = c(0,1), intercept = T)
y1.lm <- lm(y1~bx-1)
lines(x0, y1.lm$fitted, col=4)
f.reg <- function(u){
  bx <- bs(u,knots=knotlist, Boundary.knots = c(0,1), intercept = T)
  ans <- bx%% y1.lm$coefficients
  return(ans[,1])
}

tem <- function(u){ (f.reg(u)-f(u))^2 }
integrate(tem,0,1)
#0.001820952

```

- Suppose that $\log f$ can be approximated using $\sum_{j=1}^m a_j B_j$, where B_1, \dots, B_m are basis functions, then an approximation of f is given by

$$f_a(x) = \frac{\exp(\sum_{j=1}^m a_j B_j(x))}{\int \exp(\sum_{j=1}^m a_j B_j(x)) dx},$$

where $a = (a_1, \dots, a_m)$. Note that $\int f_a(x) dx = 1$. Suppose that there exist constants c_1, \dots, c_m such that

$$1 = \sum_{j=1}^m c_j B_j(x), \quad (1)$$

then

$$\ln f_a(x) = \sum_{j=1}^m (a_j - \lambda(a) c_j) B_j(x), \quad (2)$$

where

$$\lambda(a) = \ln \left(\int e^{\sum_{j=1}^m a_j B_j(x)} dx \right).$$

Then the coefficients a_1, \dots, a_m can be estimated using maximum likelihood and m can be determined using likelihood cross-validation.

- Suppose that B_1, \dots, B_m are B-spline basis functions, we have

$$1 = \sum_{j=1}^m B_j(x),$$

so (1) holds with $c_j = 1$ for all j .

- Leave-one-out likelihood cross-validation. Let $\hat{f}_{-i,m}$ be the estimator for f with basis functions B_1, \dots, B_m based on X_1, \dots, X_n with X_i removed. Let

$$LikCV(m) = \sum_{i=1}^n \log \hat{f}_{-i,m}(X_i),$$

then m is selected so that $LikCV(m)$ is maximized.

- Exercise 1. Consider the data in Example 1. Perform density estimation by modelling $\ln f$ as the $\ln f_a$ in (2) using the B_j s in Example 1 and estimating a using maximum likelihood estimation. Find the ISE.