Functional Principal Component Analysis

May 14, 2018

Outline

The least square optimality for functional data

- Suppose we observe functions $x_1, x_2, ..., x_n$. It is not necessary to view these functions as random, but we can think of them as the observed realizations of random functions residing in some separable Hilbert space H.
- We assume that the data have been centered, i.e. $\sum_{i=1}^{n} x_i = 0$. (The estimator of the mean function.)
- Fix an integer $p < N$. We think of p as being much smaller than N, typically a single digit number.

The least square optimality for functional data

- Suppose we observe functions $x_1, x_2, ..., x_n$. It is not necessary to view these functions as random, but we can think of them as the observed realizations of random functions residing in some separable Hilbert space H.
- We assume that the data have been centered, i.e. $\sum_{i=1}^{n} x_i = 0$. (The estimator of the mean function.)
- Fix an integer $p < N$. We think of p as being much smaller than N, typically a single digit number.
- We want to find an orthonormal basis $u_1, u_2, ..., u_p$ such that

$$\hat{S}^2 = \sum_{i=1}^{N} \|x_i - \sum_{k=1}^{p} \langle x_i, u_k \rangle u_k\|^2 \tag{1}$$

is minimized.

Reduction to the finite dimensional problem

- Once a basis minimizing $\hat{S}^2$ is found, $\sum_{k=1}^{p} \langle x_i, u_k \rangle u_k$ is an approximation to x_i.
- For the p we have chosen, this approximation is uniformly optimal, in the sense of minimizing $\hat{S}^2$. This means that instead of working with infinitely dimensional curves x_i, we can work with p-dimensional vectors

$$\mathbf{x}_i = [\langle x_i, u_1 \rangle, \langle x_i, u_2 \rangle, ..., \langle x_i, u_p \rangle,]^T \tag{2}$$

- This is the central idea of functional data analysis, as to perform any practical calculations we must reduce the dimension from infinity to a finite number.

Empirical functional principal components

- The functions u_j are called collectively the optimal empirical orthonormal basis or natural orthonormal components, the words empirical and natural emphasizing that they are computed directly from the functional data.
- The functions $u_1, u_2, ..., u_p$ minimizing $\hat{S}^2$ are equal (up to a sign) to the normalized eigenfunctions, $\hat{v}_1, \hat{v}_2, ..., \hat{v}_p$ of the sample covariance operator, i.e. $\hat{C}(u_i) = \hat{\lambda}_i u_i$ where $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq, ..., \geq \hat{\lambda}_p$.
- The eigenfunctions $\hat{v}_i$ are called the empirical functional principal components (EFPC) of the data $x_1, x_2, ..., x_N$. The $\hat{v}_i$ are thus the natural orthonormal components and form the optimal empirical orthonormal basis.

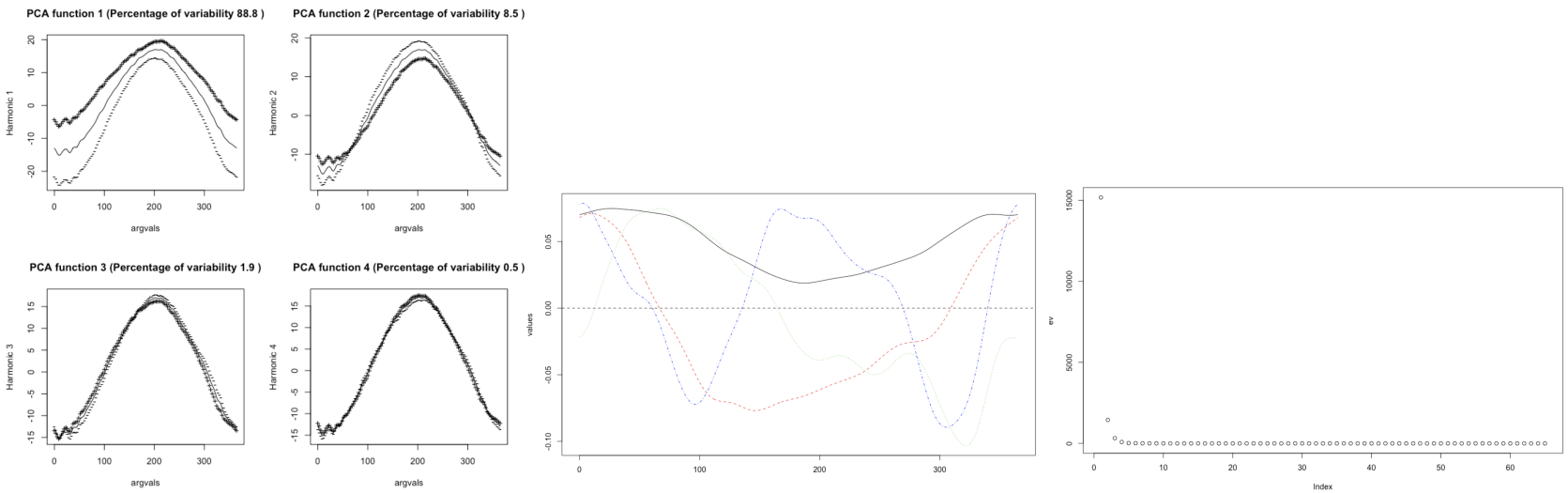
Example from the Canadian Weather Data

- The following code shows utilization of the fda for computing the EFPC for the temperature data

```
#Example of the principle component analysis

daybasis65 = create.fourier.basis(c(0, 365), nbasis=65,period=365)
harmaccelLfd = vec2Lfd(c(0,(2*pi/365)^2,0), c(0, 365))
harmfdPar = fdPar(daybasis65, harmaccelLfd, lambda=1e5)

daytempfd = smooth.basis(day.5, CanadianWeather$dailyAv[,,"Temperature.C"],
daybasis65, fdnames=list("Day", "Station", "Deg C"))$fd

daytemppcaobj = pca.fd(daytempfd, nharm=4, harmfdPar)

op = par(mfrow=c(2,2))
plot.pca.fd(daytemppcaobj, cex.main=0.9)

dev.off()
plot(daytemppcaobj$harmonics)

##Extract the eigenvalues
ev=daytemppcaobj$values

plot(ev)
```

Graphical illustration of the principle components

- The principal component functions or harmonics are shown as perturbations of the mean, which is the solid line. The +s show what happens when a small amount of a principal component is added to the mean, and the -s show the effect of subtracting the component.



Outline

FPC and Karhunen-Loeve expansion

- Suppose X are zero mean random function in H having the same distribution as X.
- Parallel to empirical optimization we can ask which orthonormal elements $v_1, ..., v_p$ in H minimize

$$E\|X - \sum_{i=1}^{p} \langle X, v_i \rangle v_i\|^2. \tag{3}$$

- The solution is given by the eigenfunctions v_i of the covariance operator C.
- They allow for the optimal representation of X.
- The functional principal components (FPC) are defined as the eigenfunctions of the covariance operator C of X.
- The representation

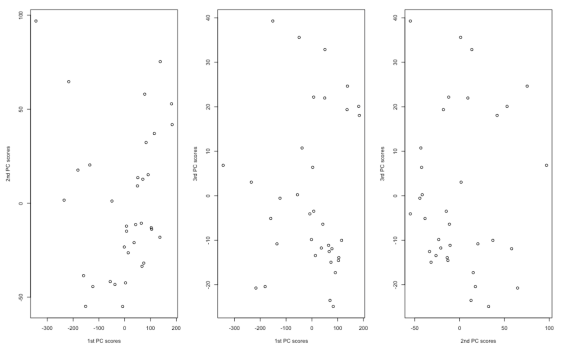
$$X = \sum_{i=1}^{\infty} \langle X, v_i \rangle v_i \tag{4}$$

is called the Karhunen-Loeve expansion

Scores

- The inner product $\langle X_i, v_j \rangle = \int X_i(t) v_j(t) dt$ is called the jth score of X_j.
- It is interpreted as the weight of the contribution of the FPC v_j to the curve X_j.

```
##plot the scores
 par(mfrow=c(1,3))
 plot(daytemppcaobj$scores[,1], daytemppcaobj$scores[,2], xlab="1st PC scores", ylab="2nd PC s
 plot(daytemppcaobj$scores[,1], daytemppcaobj$scores[,3], xlab="1st PC scores", ylab="3rd PC s
 plot(daytemppcaobj$scores[,2], daytemppcaobj$scores[,3], xlab="2nd PC scores", ylab="3rd PC s
```



Outline

Practical considerations

- We often estimate the eigenvalues and eigenfunctions of C, but the interpretation of these quantities as parameters, and their estimation, must be approached with care.
- The eigenvalues must be identifiable, so we must assume that $\lambda_1 > \lambda_2 > \dots$.
- In practice, we can estimate only the p largest eigenvalues, and assume that $\lambda_1 > \lambda_2 > \dots > \lambda_p > \lambda_{p+1}$ which implies that the first p eigenvalues are nonzero.
- The eigenfunctions v_j are defined by $C(v_j) = \lambda_j v_j$, so if v_j is an eigenfunction, then so is av_j, for any nonzero scalar a (by definition, eigenfunctions are nonzero). The v_j are typically normalized, so that $\|v_j\| = 1$, but this does not determine the sign of v_j.
- Thus if $\hat{v}_j$ is an estimate computed from the data, we can only hope that $\hat{c}_j \hat{v}_j$ is close to v_j, where

$$\hat{s}_j = sign(\langle v_j, v_j \rangle)$$

- Note that $\hat{s}_j$ cannot be computed form the data, so it must be ensured that the statistics we want to work with do not depend on the $\hat{s}_j$.
- We define the estimated eigenelements by:

$$\hat{C}_N(\hat{v}_j) = \hat{\lambda}_j \hat{v}_j \qquad j = 1, 2, \dots, N \tag{5}$$

Analysis of the Brownian Bridge case

- Since for the Brownian Bridge we have explicit representation of its eigenvalues and eigenfunction it is a convenient example to compare empirical and theoretical FPC.
- A Brownian bridge is a continuous-time stochastic process $B(t)$ whose probability distribution is the conditional probability distribution of a Wiener process $W(t)$ subject to the condition that $W(T) = 0$, so that the process is pinned at the origin at both $t = 0$ and $t = T$. More precisely:

$$B_t := (W_t \mid W_T = 0), \qquad t \in [0, T]$$

Simulation of the Brownian Bridge

The following code is pretty straight forward, we establish the grid first (line 4 to 6), generate a random noise (line 9) and pin it to 0 at time 0 (line 11 to 13). Only one sample has been generated in this case, this can be modified depending on the user's needs.

```
#Simulation an independent sample of a Brownian bridge
#over an equidistant grid

n=2000 #size of the equidistant one dimensional grid
MC=1 #Monte Carlo sample size
t=matrix(seq(0,1,by=1/n),nrow=1) #grid


ZZ=matrix(rnorm(n*MC),ncol=n)/sqrt(n) #random noise

#Simulating Brownian Bridge that starts from zero
ZeC=matrix(rep(0,MC),ncol=1)
BB=cbind(ZeC,t(apply(ZZ,1,cumsum)))-matrix(apply(ZZ,1,sum),ncol=1)%*%t

#Ploting trajectories
quartz()
plot(t,BB[1,],type='l',ylim=c(min(BB),max(BB)))
legend(0.1,max(BB)-1*0.1*max(BB),1,text.col =1)
for(i in 2:MC)
{
  lines(t,BB[i,],type='l',col=i)
  legend(0.1,max(BB)-i*0.1*max(BB),i,text.col =i)
}
```