Assigment: PCA and principal curves.

Pedro Delicado

16 de novembre de 2021

1. PCA and Principal curves for ZIP numbers

Reading the data

Consider the ZIP number data set, from the book of Hastie et al. (2009). Read the training data set (in the file `zip.train`) and select only the *zeros*.

Questions

a. Do a hierarchical clustering of these data using the `ward.D` method, plot the resulting dendogram and cut it into $k = 4$ clusters.

b. Plot the average digit at each cluster.

c. Compute the principal components for this data set. Plot the scatterplot of the scores in the first two PCs, using a different color for points in different clusters.

d. For each one of the k clusters obtained above, do the following tasks: *(A unique plot should be done, at which the k densities are represented simultaneously)*

- Consider the bivariate data set of the scores in PC1 and PC2 of the points in this cluster.
- Estimate non-parametrically the joint density of *(PC1,PC2)*, conditional to this cluster. Use the default bandwith values.
- Represent the estimated bivariate density using the level curve that covers the 75% of the points in this cluster.

e. Over the prvious plot, represent the principal curve obtained from the 256-dimensional set of zeros using the package `princurve`.

f. For each one of the k clusters obtained above, do the following tasks: *(A unique scatter plot of the scores in PC1 and PC2 should be done, over which the k densities are represented simultaneously)*

- Consider the univariate data set of the `lambda` scores over the principal curve of the points in this cluster.
- Estimate non-parametrically the density function of *lambda*, conditional to this cluster. Use the default bandwith value.
- Plot the estimated density function.

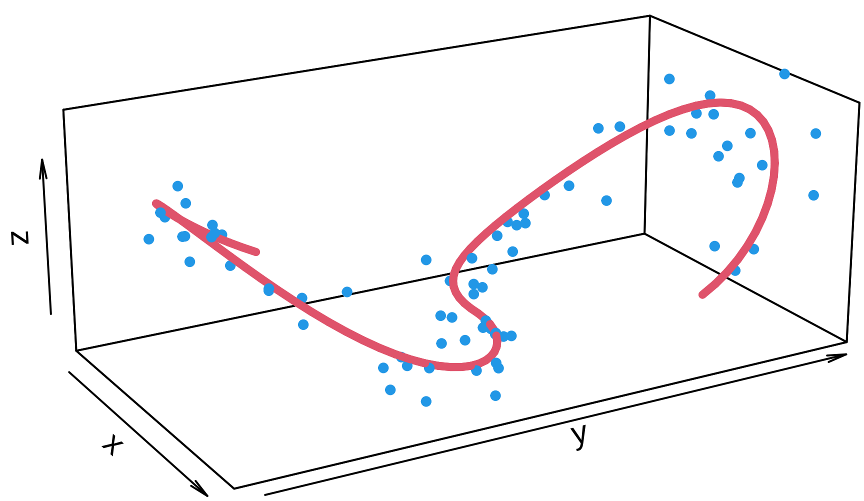
2. Choosing the smoothing parameter in Principal Curves (Hastie and Stuetzle 1989)

Consider the 3-dimensional data set generated by the following code.

```
t <- seq(-1.5*pi,1.5*pi,l=100)
R<- 1
n<-75
sd.eps <- .15
```

```r
set.seed(1)
y <- R*sign(t) - R*sign(t)*cos(t/R)
x <- -R*sin(t/R)
z <- (y/(2*R))^2
rt <- sort(runif(n)*3*pi - 1.5*pi)
eps <- rnorm(n)*sd.eps
ry <- R*sign(rt) - (R+eps)*sign(rt)*cos(rt/R)
rx <- -(R+eps)*sin(rt/R)
rz <- (ry/(2*R))^2 + runif(n,min=-2*sd.eps,max=2*sd.eps)
XYZ <- cbind(rx,ry,rz)


require(plot3D)
lines3D(x,y,z,colvar = NULL,
        phi = 20, theta = 60, r =sqrt(3), d =3, scale=FALSE,
        col=2,lwd=4,as=1,
        xlim=range(rx),ylim=range(ry),zlim=range(rz))
points3D(rx,ry,rz,col=4,pch=19,cex=.6,add=TRUE)
```



When fitting principal curves to these data, use the function `princurve::principal_curve` with the following options:

- `smoother="smooth_spline"`. This is the default, so you do not need to use it explicitely.
- The only additional argument that you will pass to `smooth_spline` will be the *degrees of freedom* `df` (see `help(smooth.spline)` if you want)

For instance, the following sentence

```
principal_curve(XYZ, df=6)
```

fits the required principal curve with degrees of freedom `df` equal to 6.

Questions

a. Choose the value of the degrees of freedom `df` by leave-one-out cross-validation.

Restrict the search of `df` to `seq(2,8,by=1)`.

(Hint: The function `project_to_curve` should be used and, specifically the element `dist` of the object it returns).

b. Give a graphical representation of the principal curve output for the optimal `df` and comment on the obtained results.

c. Compute the leave-one-out cross-validation for `df=50` and compare it with the result corresponding to the optimal `df` value you found before.

- Before fitting the principal curve with `df=50` and based only on the leave-one-out cross-validation values, what value for `df` do you think that is better, the previous optimal one or `df=50`?
- Fit now the principal curve with `df=50` and plot the fitted curve in the 3D scatterplot of the original points.
- Now, what value of `df` do you prefer?
- The overfitting with `df=50` is clear. Nevertheless leave-one-out cross-validation has not been able to detect this fact. Why do you think that `df=50` is given a so good value of leave-one-out cross-validation?