Aula Prática 1 - Análise Multivariada e Aprendizado Não-Supervisionado

Cibele Russo

15/10/2020

Prática 1

A Análise de Componentes principais é uma ferramenta bastante útil para a redução de dimensionalidade e compressão de imagens. Uma aplicação vem do reconhecimento de dígitos em códigos de endereçamento postal (CEP, ou ZIP, em inglês). Os dados a seguir são descritos em Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. The elements of statistical learning. Vol. 1. No. 10. New York: Springer series in statistics, 2001.

O objeto zip.train disponível na library ElemStatLearn ('https://cran.r-project.org/src/contrib/Archive/ElemStatLearn/') consiste em 7291 observações e 257 variáveis, sendo primeira coluna o dígito representado pelas demais colunas, que são os elementos de uma matriz de pixels 16×16.

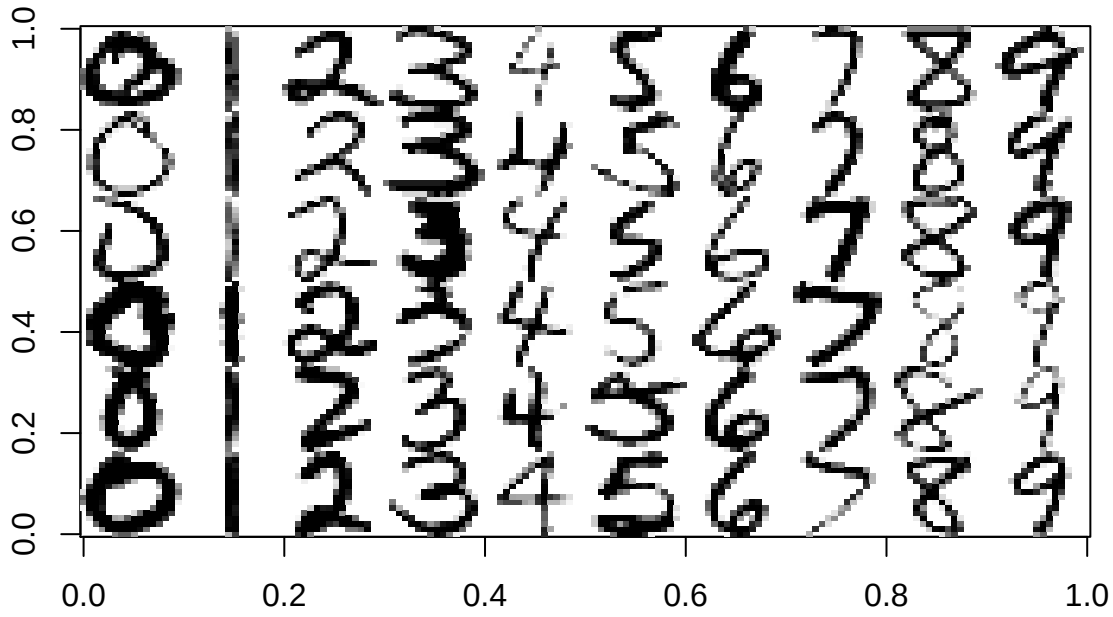
1. Desenvolva a Análise de Componentes Principais para os dados disponíveis em zip.train. Observe que a primeira coluna indica o numeral representado pelas demais 256 colunas de cada observação.
2. Escolha o número de componentes principais e justifique a sua escolha.
3. Reproduza algumas imagens do banco de dados utilizando somente as componentes que achar necessárias. Utilize alguns dos comandos sugeridos a seguir e acrescente o que achar necessário.

```
#install.packages("ElemStatLearn", 'kableExtra', 'nnet')
library(ElemStatLearn)
library(kableExtra)
library(nnet)

findRows <- function(zip, n) {
  # Find  n (random) rows with zip representing 0,1,2,...,9
  res <- vector(length=10, mode="list")
  names(res) <- 0:9
  ind <- zip[,1]
  for (j in 0:9) {
    res[[j+1]] <- sample( which(ind==j), n ) }
  return(res) }

# Making a plot like that on page 4:

digits <- vector(length=10, mode="list")
names(digits) <- 0:9
rows <- findRows(zip.train, 6)
for (j in 0:9) {
  digits[[j+1]] <- do.call("cbind", lapply(as.list(rows[[j+1]]),
                                           function(x) zip2image(zip.train, x)) )
}
```

```
## [1] "digit  0  taken"
## [1] "digit  0  taken"
## [1] "digit  0  taken"
## [1] "digit  0  taken"
## [1] "digit  0  taken"
## [1] "digit  0  taken"
## [1] "digit  1  taken"
## [1] "digit  1  taken"
## [1] "digit  1  taken"
## [1] "digit  1  taken"
## [1] "digit  1  taken"
## [1] "digit  1  taken"
## [1] "digit  2  taken"
## [1] "digit  2  taken"
## [1] "digit  2  taken"
## [1] "digit  2  taken"
## [1] "digit  2  taken"
## [1] "digit  2  taken"
## [1] "digit  3  taken"
## [1] "digit  3  taken"
## [1] "digit  3  taken"
## [1] "digit  3  taken"
## [1] "digit  3  taken"
## [1] "digit  3  taken"
## [1] "digit  4  taken"
## [1] "digit  4  taken"
## [1] "digit  4  taken"
## [1] "digit  4  taken"
## [1] "digit  4  taken"
## [1] "digit  4  taken"
## [1] "digit  5  taken"
## [1] "digit  5  taken"
## [1] "digit  5  taken"
## [1] "digit  5  taken"
## [1] "digit  5  taken"
## [1] "digit  5  taken"
## [1] "digit  6  taken"
## [1] "digit  6  taken"
## [1] "digit  6  taken"
## [1] "digit  6  taken"
## [1] "digit  6  taken"
## [1] "digit  6  taken"
## [1] "digit  7  taken"
## [1] "digit  7  taken"
## [1] "digit  7  taken"
## [1] "digit  7  taken"
## [1] "digit  7  taken"
## [1] "digit  7  taken"
## [1] "digit  8  taken"
## [1] "digit  8  taken"
## [1] "digit  8  taken"
## [1] "digit  8  taken"
## [1] "digit  8  taken"
## [1] "digit  8  taken"
```

```
## [1] "digit  9  taken"
## [1] "digit  9  taken"
## [1] "digit  9  taken"
## [1] "digit  9  taken"
## [1] "digit  9  taken"
## [1] "digit  9  taken"
```

```
im <- do.call("rbind", digits)
image(im, col=gray(256:0/256), zlim=c(0,1), xlab="", ylab="" )
```



Análise de Componentes Principais

```
pca<- prcomp(zip.train[,-1])
R<- pca$sdev
```

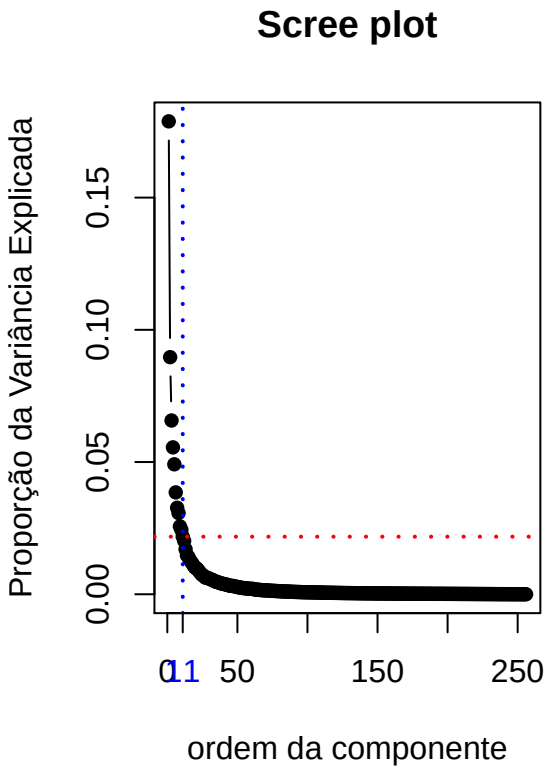
Autovalores e proporção da variância explicada dos componentes principais (CP) (Apresentamos as 15 primeiras CP).

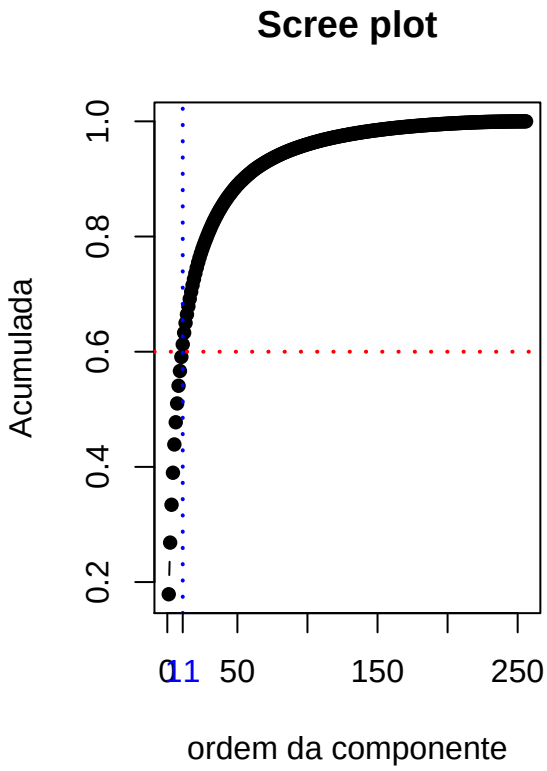
```
v1<- round(R^2 / sum(R^2), 3) #proporção da variância explicada
v2<- cumsum((R^2 / sum(R^2)))
kable(data.frame(paste("CP", 1: 15), v1[1:15], v2[1:15]),
      col.names = c("CP","Proporção Variância Explicada", "Acumulada"),
      align = "c")  %>% kable_styling(position = "center", font_size = 12)
```

CP	Proporção Variância Explicada	Acumulada
CP 1	0.179	0.1788442
CP 2	0.090	0.2685146
CP 3	0.066	0.3342319
CP 4	0.056	0.3897774
CP 5	0.049	0.4389193
CP 6	0.039	0.4774385
CP 7	0.033	0.5101490
CP 8	0.031	0.5408593
CP 9	0.026	0.5665276
CP 10	0.024	0.5908946
CP 11	0.022	0.6126697
CP 12	0.020	0.6328776
CP 13	0.017	0.6499137
CP 14	0.015	0.6646049
CP 15	0.014	0.6787671

Scree-plot para auxiliar determinar quantas CP serão selecionadas

```
par(mfrow=c(1,2))

aux<- R^2 / sum(R^2)
plot(aux, type="b", pch=16, main="Scree plot", ylab="Proporção da Variância Explicada",
     xlab="ordem da componente")
abline(h = aux[11],  v = 11, col=c("red","blue"), lwd=2, lty= 3)
axis(1, at= 11, labels=11, col.axis="blue")

plot(cumsum(R^2 / sum(R^2)), type="b", pch=16,
     main="Scree plot", ylab="Acumulada", xlab="ordem da componente")
abline(h = 0.6,  v = 11, col=c("red","blue"), lwd=2, lty= 3)
axis(1, at= 11, labels=11, col.axis="blue")
```






```
pca1<- prcomp(zip.train[,-1], rank. = 11)
#round(cor(pca1$x))  # cp são não correlacionados
```

Escores das componentes para a base de treino

```
new <- predict(pca1, zip.train[,-1])
new <- data.frame(num=zip.train[,1], new)
```

Classificação via modelo multinomial via redes neurais com matriz de confusão (Exercício)

```
modelo<- multinom(num~., data=new)
```

```
## # weights:  130 (108 variable)
## initial  value 16788.147913
## iter  10 value 3409.221249
## iter  20 value 3005.782809
## iter  30 value 2849.235523
## iter  40 value 2658.968795
## iter  50 value 2581.380623
## iter  60 value 2518.023414
## iter  70 value 2446.482745
## iter  80 value 2422.956867
## iter  90 value 2404.252720
## iter 100 value 2389.102190
## final  value 2389.102190
## stopped after 100 iterations
```

```
p<- predict(modelo, new)
```

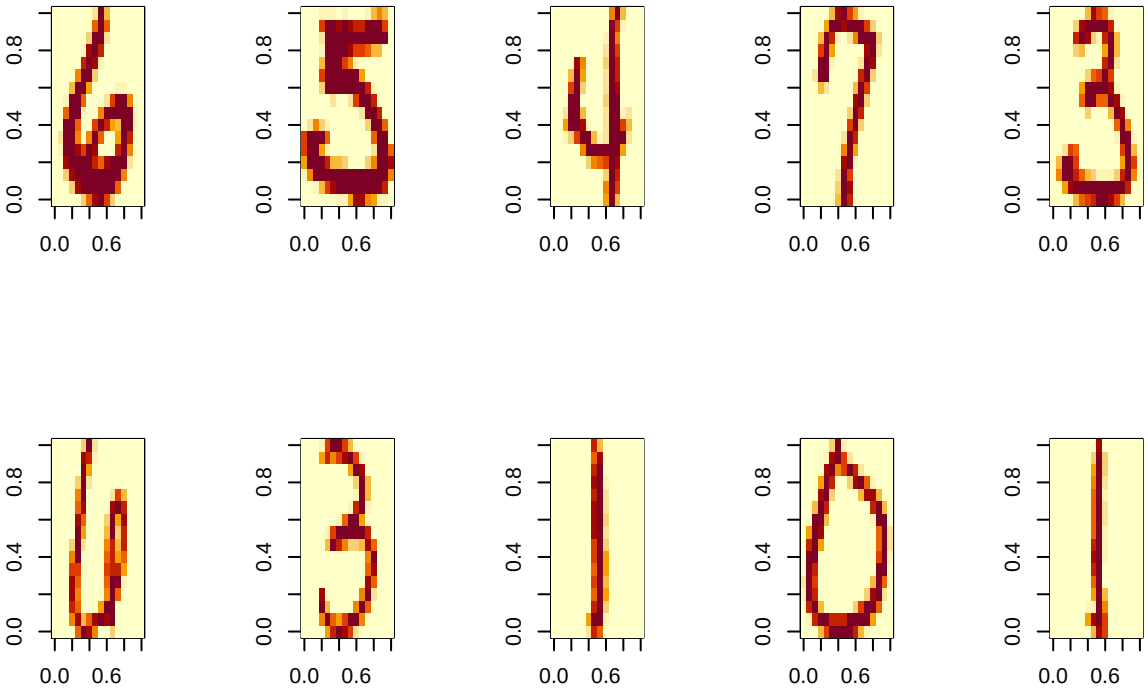
```
##
## p      0    1    2    3    4    5    6    7    8    9
##   0 1138    0   14    2    2   18    8    1    4    3
##   1    0 1002    1    0    5    0    4    0    4    5
##   2   13    0  638    9   15    8   26    5    9    0
##   3    4    0   11  584    0   30    0    0   15    2
##   4    4    0   28    1  559   15    3    3    9   25
##   5   18    1    6   34    2  440    4    4   21    8
##   6   13    0   14    0   16   15  615    0    2    0
##   7    0    0    5    2    2    0    0  594    2   40
##   8    4    2   14   23    9   19    3    5  469    5
##   9    0    0    0    3   42   11    1   33    7  556
```

Na diagonal principal são apresentados o número de acertos e fora da diagonal principal os elementos com classificasãoincorreta.

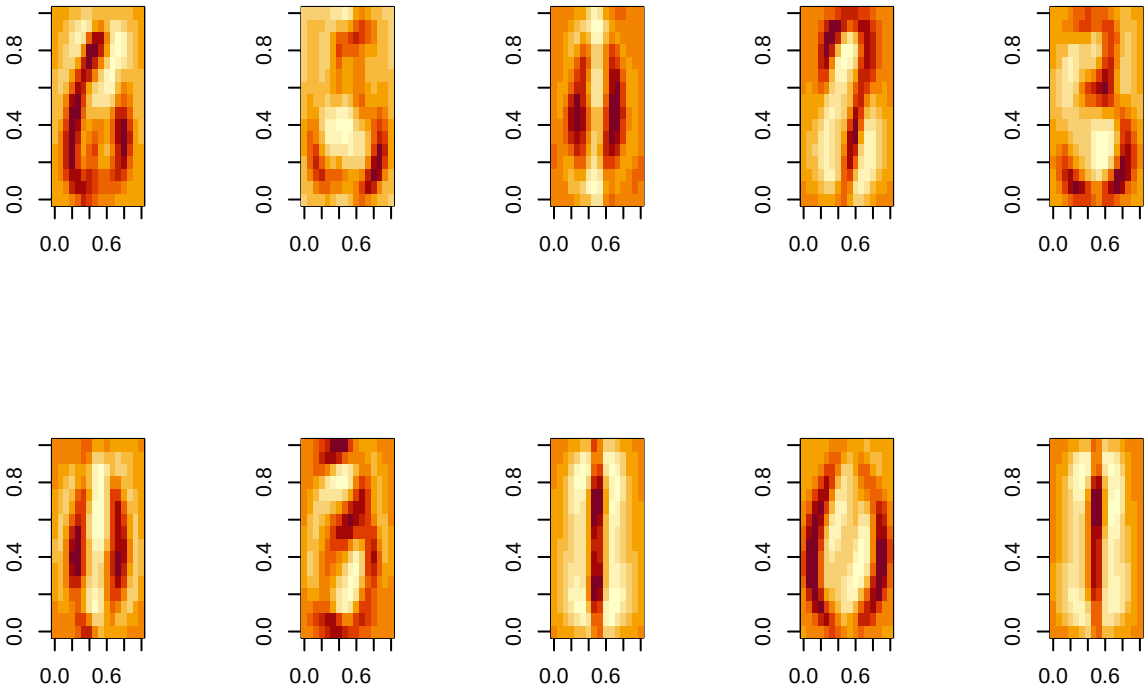
Comparação da imagem original com a imagem comprimida usando 11 componentes principais.

```
pca<- prcomp(zip.train[,-1])
pred <- pca$x[,1:11] %*% t(pca$rotation[,1:11])
```

```
par(mfrow= c(2,5))

for (i in 1:10){
  im <- zip.train[i,-1]
  im <- t(matrix(im, 16, 16, byrow = TRUE))
  im <- im[, 16:1]
  image(im)
}
```




```
for (i in 1:10){
  im <- t(matrix(pred[i,], 16, 16, byrow = TRUE))
  im <- im[, 16:1]
  image(im)
}
```



O objeto zip.test contém 2007 observações que não foram utilizadas para a ACP. Faça a previsão para essas observações utilizando a ACP.