Lista 8: Fundamentos Estatísticos para Ciência dos Dados

Ricardo Pagoto Marinho

4 de maio de 2018

- Para esta lista, um algoritmo semelhante ao apresentado na lista foi utilizado. Ele é apresentado a seguir:

```
library(imager)
im<-read.csv("zip.train",header = FALSE,sep="")
leio o arquivo de teste
mat_dig<-list()
for(i in 1:10){
    mat_dig[[i]]<-list()
}
crio uma matriz para armazenar as informacoes
sobre as imagens dos 10 digitos
for(i in 1:nrow(im)){
    dig<-im[i,1]+1
    descubro o digito da linha
    pixels<-im[i,2:ncol(im)]
    pego os pixels da imagem do digito
    ind<-length(mat_dig[[dig]])+1
    descubro quantas imagens ja estao armazenadas para aquele
    digito
    mat_dig[[dig]][[ind]]<-matrix(pixels,ncol = 16,nrow = 16)
    armazeno os pixels da imagem daquele digito em uma matriz 16x1
}
como os digitos nao aparecem na mesma quantidade nas imagens
preciso saber qual a quantidade de vezes que o digito que
aparece mais vezes aparece
maior<-0
for(i in 1:length(mat_dig[])){
    if(length(mat_dig[[i]])>maior){
        maior<-length(mat_dig[[i]])
    }
}
mat_pixels<-matrix(0,nrow = 16*16,ncol=10*maior)
crio a matriz de pixels com o tamanho da imagem (16x16)
e a quantidade de imagens (10*maior)
for(i in 1:10){
    for(j in 1:length(mat_dig[[i]])){
        mat_pixels[,j+(i-1)*length(mat_dig[[i]])] =
        stack(as.data.frame(mat_dig[[i]][[j]]))[,1]
    }
}
set.seed(123)
ind=sample(10,1:maior,replace = T)
indcol=ind+((1:10)-1)*maior
```

```
mat_test=mat_pixels[,indcol]
acho a imagem de teste
mat_pixels=mat_pixels[,-indcol]
mat_media=apply(mat_pixels, 1, mean)
mat_centrada=mat_pixels-mat_media
pca_pixels=prcomp(t(mat_centrada))
autovalores=(pca_pixels$sdev)^2
autovetores = pca_pixels$rot[ , 1:20]
exemplo utilizando 20 autovetores
auto_face=list()
for(i in 1:20){
    auto_face[[i]]=as.cimg(pca_pixels$rot[,i],x=16,y=16)
}
coef = t(autovetores) %*% (mat_test - mat_media)
coef_treino = t(autovetores) %*% (mat_pixels - mat_media)
coefmedio = matrix(0, ncol=10, nrow=20)
for(i in 1:10){
    coefmedio[,i] = apply(coef_treino[ ,(1+(i-1)*10) : (i*10)], 1, mean)
}
indproximo = numeric()
for(j in 1:10){
    indproximo[j] = which.min( apply((coefmedio - coef[,j])^2, 2, mean) )
}
```

0	1	2	3	4	5	6	7	8	9
0	3	4	2	5	6	8	1	7	0
1	1	3	5	9	1	7	2	4	4
0	3	4	2	5	6	8	1	7	0
1	1	3	5	9	1	7	2	4	4
4	0	2	4	9	5	4	6	8	6
0	3	4	2	5	6	8	1	7	0
6	0	2	4	9	5	4	6	8	6
1	1	3	5	9	1	7	2	4	4
0	3	4	2	5	6	8	1	7	0
1	1	3	5	9	1	7	2	4	4