Exercício 3: K-means Clustering

Rúbia Reis Guerra
2013031143

16 de Abril de 2017

1 K-médias

K-médias (MacQueen, 1967) é um dos algoritmos de aprendizagem não supervisionada mais simples que resolve o conhecido problema de agrupamento. O procedimento segue uma maneira simples e fácil de classificar um conjunto de dados através de um certo número de clusters (k) fixados a priori. A ideia principal é definir k centróides, um para cada cluster. O próximo passo é associar cada ponto pertence a um determinado conjunto de dados ao centróide mais próximo. Feito isso, um protótipo de cluster está concluído. Neste ponto, precisa-se recalcular k novos centroides como baricentros dos clusters resultantes do passo anterior. Depois de ter esses novos centroídes k, novos agrupamentos são criados entre os pontos do conjunto de dados e os respectivos novos centróides mais próximos. Itera-se os passos anteriores, até os centróides não se movam mais.

1.1 Funções

Foram implementadas a função k-médias (mykmeans) e a função auxiliar para calcular a distância de cada conjunto de pontos ao centróide (distance).

```
<<echo=T,fig=F>>=
rm(list=ls())
library('MASS')
distance <- function(xt, centers){
  distMatrix <- matrix(NA, nrow=dim(xt)[1], ncol=dim(centers)[1])
  for(i in 1:nrow(centers)) {
    distMatrix[,i] <- sqrt(rowSums(t(t(xt)-centers[i,])^2))
  }
  distMatrix
}

mykmeans <- function(x, k, maxIter) {
  clusterOld <- c()
  centerOld <- c()
  centers <- x[sample(nrow(x), k),]
```

```
  flag <- FALSE
  i <- 0
  while(i <= maxIter && flag==FALSE) {
    i <- i + 1
    if(i > 1) {
      clusterOld <- clusters
      centerOld <- centers
    }
    distsToCenters <- distance(x, centers)
    clusters <- apply(distsToCenters, 1, which.min)
    centers <- apply(x, 2, tapply, clusters, mean)
    flag <- identical(clusters,clusterOld)
  }

  list(clusters=clusters, centers=centers)
}
@
```

1.2 Testes

Conforme indicado nas instruções da atividade, a implementação de k-médias foi testada para $sd = \{0.3, 0.5, 0.7\}$ e $k = \{2, 4, 8\}$.

```
<<echo=T,fig=F>>=
########################################################################
k <- c(2, 4, 8)
sd2 <- (c(0.3, 0.5, 0.7)^2)
for(j in 1:length(sd))
{
  for(i in 1:length(k))
  {
    N <- 100
    maxIter <- 100
    cores <- rainbow(k[i])

    S <- matrix(c(sd2[j],0,0,sd2[j]),byrow=T,ncol=2)
    g1 <- mvrnorm(N,mu=c(2,2), Sigma=S)
    g2 <- mvrnorm(N,mu=c(2,4), Sigma=S)
    g3 <- mvrnorm(N,mu=c(4,2), Sigma=S)
    g4 <- mvrnorm(N,mu=c(4,4), Sigma=S)
    samples <- rbind(g1,g2,g3,g4)

    b <- mykmeans(samples, k[i], maxIter)
    for(l in 1:(k[i]-1))
    {
```

```
      plot(samples[b$clusters==l,1],samples[b$clusters==l,2],type='p',
      col=cores[l],xlab='',ylab='',xlim=c(0,6),ylim=c(0,6))
      par(new=T)
    }

    plot(samples[b$clusters==k[i],1],samples[b$clusters==k[i],2],type='p',
    col=cores[k[i]],xlab='',ylab='',xlim=c(0,6),ylim=c(0,6))
  }
}
@
```

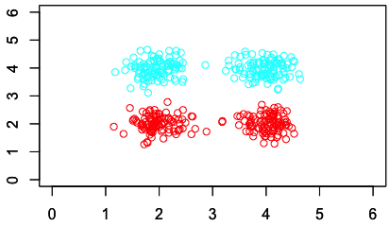

Figura 1: sd = 0.3, k = 2

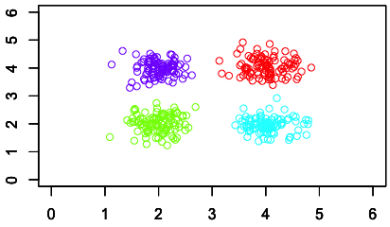

Figura 2: sd = 0.3, k = 4

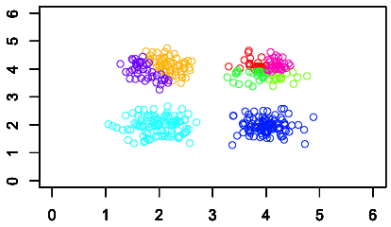

Figura 3: sd = 0.3, k = 8

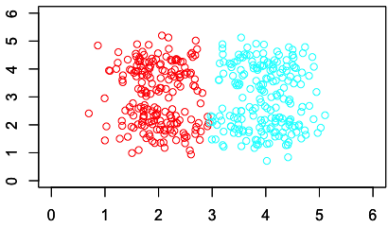

Figura 4: sd = 0.5, k = 2

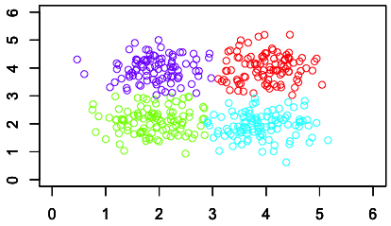

Figura 5: sd = 0.5, k = 4

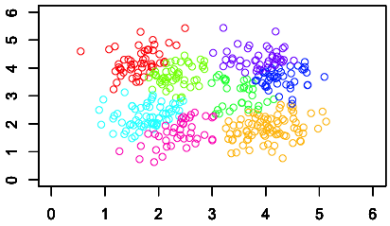

Figura 6: sd = 0.5, k = 8

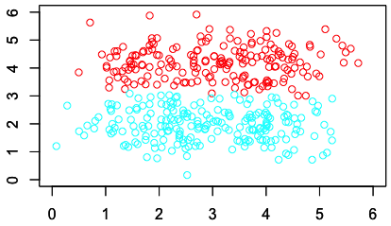

Figura 7: sd = 0.7, k = 2

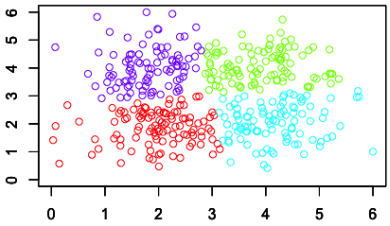

Figura 8: sd = 0.7, k = 4

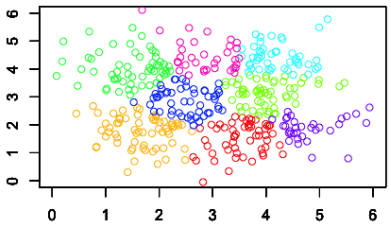

Figura 9: sd = 0.7, k = 8