**## R CODE FOR THE SIMULATIONS IN CHAPTER 7: CLUSTERING**

**## 3,000 words (nwords)**

**## 1,000 documents (ndoc). Document lengths: Average = 500; Standard Deviation = 100**

**## 3 clusters**

**## Cluster 1: Words 1 to 10 with probability 10 times larger than the probability of remaining words**

**## Cluster 2: Words 11 to 20 with probability 10 times larger than the probability of remaining words**

**## Cluster 3: Words 21 to 30 with probability 10 times larger than the probability of remaining words**

**library(slam)**

**library(tm)**

**library(cluster)**

**library(skmeans)**

**library(movMF)**

**nwords=3000**

**word=dim(3000)**

**for (i in 1:3000) {**

**word[i]=paste("word",i, sep = "")**

**}**

**word**

**ndoc=1000**

**ncluster=3**

**nobs=dim(ndoc)**

**set.seed(1234)**

**nobs=as.integer(rnorm(1000,500,100))**

**p=matrix(nrow=3,ncol=3000)**

**f=matrix(nrow=1000,ncol=3000)**

**k=nwords-10**

**r=10 ## factor to increase the probabilities**

**prob1=r**

**prob2=1**

**sum=10\*prob1+k\*prob2**

**prob1=r/sum**

**prob2=1/sum**

**for (i in 1:10) {**

**p[1,i]=prob1**

**}**

**for (i in 11:nwords) {**

**p[1,i]=prob2**

**}**

**for (i in 1:10) {**

**p[2,i]=prob2**

**}**

**for (i in 11:20) {**

**p[2,i]=prob1**

**}**

**for (i in 21:nwords) {**

**p[2,i]=prob2**

**}**

**for (i in 1:20) {**

**p[3,i]=prob2**

**}**

**for (i in 21:30) {**

**p[3,i]=prob1**

**}**

**for (i in 31:nwords) {**

**p[3,i]=prob2**

**}**

**plot(p[1,1:50])**

**plot(p[2,1:50])**

**plot(p[3,1:50])**

**set.seed(1234)**

**group=dim(3)**

**txt=dim(ndoc)**

**for (i in 1:200) {**

**group[i]=1**

**txt[i]=toString(sample(word,nobs[i],replace=TRUE,p[1,]))**

**}**

**for (i in 201:500) {**

**group[i]=2**

**txt[i]=toString(sample(word,nobs[i],replace=TRUE,p[2,]))**

**}**

**for (i in 501:1000) {**

**group[i]=3**

**txt[i]=toString(sample(word,nobs[i],replace=TRUE,p[3,]))**

**}**

**## create corpus**

**corpus <- VCorpus(VectorSource(txt),readerControl = list(reader = readPlain)) ## this is how to create corpus**

**corpus <- tm\_map(corpus, content\_transformer(tolower))**

**corpus <- tm\_map(corpus, removePunctuation)**

**corp.dtm <- DocumentTermMatrix(corpus,control=list(stemming=FALSE))**

**corp.dtm**

**##########################################################**

**####################### agglomorative ######################**

**##########################################################**

**## we use the program hclust and agnes (in R-package cluster)**

**## argument diss=FALSE: we use the dissimilarity matrix that is being calculated from raw data.**

**## argument metric="euclidian" indicates that we use Euclidian distance**

**## default is "average" linkage. We use “ward”**

**xx=as.matrix(corp.dtm)**

**for (i in 1:ndoc) {**

**xx[i,]=xx[i,]/sum(xx[i,])**

**}**

**xx ## relative (document) frequencies**

**## hclust**

**set.seed(1234)**

**hc <- hclust(dist(xx),"ward")**

**plot(hc) ## dendrogram**

**hcd <- as.dendrogram(hc)**

**## Customized plot; omit document labels**

**plot(hcd, ylab = "Distance", xlab="Agglomerative Clustering: Euclidean Distance and Ward Linkage",leaflab = "none")**

**## Customized plot; focus on 50 documents**

**plot(hcd,ylab = "Distance", xlab="Agglomerative Clustering: Euclidean Distance and Ward Linkage",xlim = c(1, 50))**

**clustering=cutree(hc,k=3) ## results for three clusters**

**table(clustering)**

**colnames(xx)[order(colMeans(xx[clustering==1,]),decreasing = TRUE)[1:10]]**

**colnames(xx)[order(colMeans(xx[clustering==2,]),decreasing = TRUE)[1:10]]**

**colnames(xx)[order(colMeans(xx[clustering==3,]),decreasing = TRUE)[1:10]]**

**tt=table(group,clustering)**

**tt**

**max(tt[1,])**

**max(tt[2,])**

**max(tt[3,])**

**misclass=100\*(ndoc-(max(tt[1,])+max(tt[2,])+max(tt[3,])))/ndoc**

**misclass**

**## agnes (R-library cluster)**

**set.seed(1234)**

**clus=agnes(xx,diss=FALSE,metric="euclidean",method="ward")**

**plot(clus) ## dendrogram**

**clus$merge ## describes the sequential merge steps**

**clustering=cutree(clus,k=3) ## results for 3 clusters**

**table(clustering)**

**colnames(xx)[order(colMeans(xx[clustering==1,]),decreasing = TRUE)[1:10]]**

**colnames(xx)[order(colMeans(xx[clustering==2,]),decreasing = TRUE)[1:10]]**

**colnames(xx)[order(colMeans(xx[clustering==3,]),decreasing = TRUE)[1:10]]**

**tt=table(group,clustering)**

**tt**

**max(tt[1,])**

**max(tt[2,])**

**max(tt[3,])**

**misclass=100\*(ndoc-(max(tt[1,])+max(tt[2,])+max(tt[3,])))/ndoc**

**misclass**

**##########################################################**

**####################### k-means ###########################**

**##########################################################**

**xx=as.matrix(corp.dtm)**

**for (i in 1:ndoc) {**

**xx[i,]=xx[i,]/sum(xx[i,])**

**}**

**xx ## relative (document) frequencies**

**set.seed(1234)**

**clus=kmeans(xx,centers=3, nstart=10)**

**clus**

**uu=as.matrix(clus$centers)**

**colnames(uu)[order(uu[1,], decreasing = TRUE)[1:10]]**

**colnames(uu)[order(uu[2,], decreasing = TRUE)[1:10]]**

**colnames(uu)[order(uu[3,], decreasing = TRUE)[1:10]]**

**clustering=clus$cluster**

**table(clustering)**

**tt=table(group,clustering)**

**tt**

**max(tt[1,])**

**max(tt[2,])**

**max(tt[3,])**

**misclass=100\*(ndoc-(max(tt[1,])+max(tt[2,])+max(tt[3,])))/ndoc**

**misclass**

**##########################################################**

**####################### SPHERICAL k-means CLUSTERING #######**

**##########################################################**

**xx=as.matrix(corp.dtm)**

**xx ## count frequencies**

**set.seed(1234)**

**party <- skmeans(xx, k =3, control = list(nruns = 20)) ## hard partition**

**party**

**party$cluster**

**table(party$cluster)**

**party$prototypes**

**uu=as.matrix(party$prototypes)**

**colnames(uu)[order(uu[1,], decreasing = TRUE)[1:10]]**

**colnames(uu)[order(uu[2,], decreasing = TRUE)[1:10]]**

**colnames(uu)[order(uu[3,], decreasing = TRUE)[1:10]]**

**tt=table(group,party$cluster)**

**tt**

**max(tt[1,])**

**max(tt[2,])**

**max(tt[3,])**

**misclass=100\*(ndoc-(max(tt[1,])+max(tt[2,])+max(tt[3,])))/ndoc**

**misclass**

**##########################################################**

**####################### CLUSTERING WITH MOVMF ############**

**##########################################################**

**xx=as.matrix(corp.dtm)**

**xx ## count frequencies**

**set.seed(1234)**

**best\_model <- movMF(xx, k = 3, nruns = 20,kappa = list(common = TRUE))**

**best\_model**

**apply(coef(best\_model)$theta, 1, function(x)**

**colnames(coef(best\_model)$theta)[order(x, decreasing = TRUE)[1:10]])**

**clustering <- predict(best\_model)**

**clustering**

**table(clustering)**

**tt=table(group,clustering)**

**tt**

**max(tt[1,])**

**max(tt[2,])**

**max(tt[3,])**

**misclass=100\*(ndoc-(max(tt[1,])+max(tt[2,])+max(tt[3,])))/ndoc**

**misclass**