seriation2000.r

Sun Apr 23 01:39:57 2017

vecs<-read.csv( "C:\\Users\\Tony\\Dropbox\\Rowan\\DM2\\Lecture12\\vecs2c.txt",header=FALSE, stringsAsFactors=TRUE)  
  
dim(vecs)

## [1] 1292 2001

## transpose to column vectors  
vecs2 <-t(vecs)  
vecs2[1:5,1:5]

## [,1] [,2] [,3] [,4] [,5]   
## V1 "Mihir\_0" "John\_1" "Eric\_2" "Eric\_3" "Eric\_4"  
## V2 " 0" " 0" " 0" " 0" " 0"   
## V3 " 0" " 0" " 5" " 30" " 25"   
## V4 " 0" " 0" " 0" " 1" " 0"   
## V5 " 0" " 0" " 0" " 0" " 0"

vecs3<-as.data.frame(vecs2[2:2001,])  
vecs3[1:5,1:5]

## V1 V2 V3 V4 V5  
## V2 0 0 0 0 0  
## V3 0 0 5 30 25  
## V4 0 0 0 1 0  
## V5 0 0 0 0 0  
## V6 0 0 0 0 0

colnames(vecs3)<-vecs2[1,]  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.3.3

vecs3[1:5,1:5]

## Mihir\_0 John\_1 Eric\_2 Eric\_3 Eric\_4  
## V2 0 0 0 0 0  
## V3 0 0 5 30 25  
## V4 0 0 0 1 0  
## V5 0 0 0 0 0  
## V6 0 0 0 0 0

dim(vecs3)

## [1] 2000 1292

mymatrix <- matrix(nrow=2000,ncol=1292)  
  
mymatrix[1:5,1:5]

## [,1] [,2] [,3] [,4] [,5]  
## [1,] NA NA NA NA NA  
## [2,] NA NA NA NA NA  
## [3,] NA NA NA NA NA  
## [4,] NA NA NA NA NA  
## [5,] NA NA NA NA NA

for (i in 1:2000){  
 for (j in 1:1292){  
 mymatrix[i,j]<-as.integer(as.character(vecs3[i,j]))  
 }  
}  
mymatrix[1:5,1:5]

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 0 0 0 0 0  
## [2,] 0 0 5 30 25  
## [3,] 0 0 0 1 0  
## [4,] 0 0 0 0 0  
## [5,] 0 0 0 0 0

rownames(mymatrix)<-rownames(vecs3)  
colnames(mymatrix)<-colnames(vecs3)  
dim(mymatrix)

## [1] 2000 1292

## can’t get heatmap to work  
##myheatmap <- heatmap(mymatrix[1:50,1:50], Rowv=NA, Colv=NA, col = cm.colors(256), scale="column", margins=c(5,10))

library(seriation)

## Warning: package 'seriation' was built under R version 3.3.3

mat2<-seriate(mymatrix, method = "PCA\_angle", control = NULL, margin = 2)  
mat2

## object of class 'ser\_permutation', 'list'  
## contains permutation vectors for 1-mode data  
##   
## vector length seriation method  
## 1 1292 PCA\_angle

colnames(mymatrix)[1:100]

## [1] "Mihir\_0" "John\_1" "Eric\_2" "Eric\_3" "Eric\_4"   
## [6] "Eric\_5" "Stephen\_6" "Tim\_7" "Tim\_8" "Chris\_9"   
## [11] "Yousuf\_10" "Stephen\_11" "John\_12" "Tony\_13" "Parvati\_14"  
## [16] "Eric\_15" "Stephen\_16" "Yousuf\_17" "John\_18" "Matt\_19"   
## [21] "Matt\_20" "Chris\_21" "Tony\_22" "John\_23" "Chris\_24"   
## [26] "Chris\_25" "Parvati\_26" "Matt\_27" "Matt\_28" "Tim\_29"   
## [31] "Chris\_30" "Tony\_31" "Matt\_32" "Eric\_33" "Mihir\_34"   
## [36] "Stephen\_35" "Tim\_36" "Stephen\_37" "Stephen\_38" "Yousuf\_39"   
## [41] "John\_40" "Chris\_41" "Chris\_42" "Yousuf\_43" "John\_44"   
## [46] "Yousuf\_45" "Parvati\_46" "Mihir\_47" "Eric\_48" "Tim\_49"   
## [51] "Tony\_50" "Tim\_51" "Chris\_52" "Eric\_53" "Tony\_54"   
## [56] "Yousuf\_55" "Chris\_56" "John\_57" "Eric\_58" "Tim\_59"   
## [61] "Yousuf\_60" "Mihir\_61" "Stephen\_62" "Mihir\_63" "Mihir\_64"   
## [66] "Tony\_65" "Stephen\_66" "Parvati\_67" "Chris\_68" "Eric\_69"   
## [71] "Tim\_70" "Eric\_71" "Tim\_72" "Tony\_73" "Eric\_74"   
## [76] "Stephen\_75" "Tony\_76" "John\_77" "Chris\_78" "Eric\_79"   
## [81] "Yousuf\_80" "Chris\_81" "Yousuf\_82" "Yousuf\_83" "John\_84"   
## [86] "Parvati\_85" "Stephen\_86" "Yousuf\_87" "John\_88" "Chris\_89"   
## [91] "Chris\_90" "Stephen\_91" "Yousuf\_92" "Chris\_93" "Tim\_94"   
## [96] "Eric\_95" "Chris\_96" "John\_97" "Eric\_98" "Parvati\_99"

## vectors are originally random. After seriation we would expect to have mostly one name in the first 100

colnames(mymatrix[,get\_order(mat2)])[1:100]

## [1] "Eric\_330" "Eric\_854" "Eric\_503" "Eric\_387" "Eric\_892"   
## [6] "Eric\_704" "Eric\_33" "Eric\_191" "Eric\_570" "Eric\_332"   
## [11] "Eric\_599" "Eric\_448" "Eric\_1221" "Eric\_1203" "Eric\_393"   
## [16] "Eric\_542" "Eric\_1116" "Eric\_1022" "Eric\_603" "Eric\_79"   
## [21] "Eric\_1161" "Eric\_58" "Eric\_282" "Eric\_296" "Eric\_71"   
## [26] "Eric\_1064" "Eric\_639" "Eric\_1017" "Eric\_879" "Eric\_53"   
## [31] "Eric\_182" "Eric\_479" "Eric\_670" "Eric\_386" "Eric\_827"   
## [36] "Eric\_1201" "Stephen\_478" "Eric\_1098" "Eric\_3" "Eric\_887"   
## [41] "Eric\_147" "Eric\_1081" "Eric\_1195" "Eric\_1235" "Eric\_335"   
## [46] "Eric\_213" "Eric\_1281" "Eric\_1164" "Chris\_990" "Eric\_1043"   
## [51] "Eric\_454" "Eric\_1190" "Eric\_902" "Eric\_265" "Eric\_866"   
## [56] "Eric\_263" "Eric\_1169" "Eric\_845" "Eric\_456" "Eric\_694"   
## [61] "Eric\_198" "Eric\_1285" "Eric\_654" "Eric\_1132" "Eric\_1090"   
## [66] "Eric\_673" "Eric\_4" "Eric\_923" "Eric\_600" "Eric\_552"   
## [71] "Eric\_430" "Eric\_1240" "Eric\_590" "Eric\_1253" "Eric\_705"   
## [76] "Eric\_48" "Eric\_972" "Eric\_1241" "Eric\_932" "Eric\_193"   
## [81] "Eric\_392" "Eric\_724" "Eric\_452" "Eric\_560" "Eric\_681"   
## [86] "Eric\_996" "Eric\_170" "Eric\_146" "Eric\_471" "Chris\_579"   
## [91] "Eric\_463" "Eric\_617" "Eric\_470" "Eric\_573" "Eric\_403"   
## [96] "Eric\_1128" "Eric\_571" "Eric\_928" "Eric\_567" "Eric\_893"

library(plyr)

## Warning: package 'plyr' was built under R version 3.3.3

## we can do better testing with count  
## Note first 100 vectors roughly equally distributed

count(substr(colnames(mymatrix)[1:100],1,4))

## x freq  
## 1 Chri 16  
## 2 Eric 15  
## 3 John 11  
## 4 Matt 5  
## 5 Mihi 6  
## 6 Parv 6  
## 7 Step 11  
## 8 Tim\_ 10  
## 9 Tony 8  
## 10 Yous 12

count(substr(colnames(mymatrix[,get\_order(mat2)])[1:100],1,4))

## x freq  
## 1 Chri 2  
## 2 Eric 97  
## 3 Step 1

## Now we see 97 of the first 100 documents were clustered into a mostly eric cluster

## let’s look at next documents after Eric’s stop

count(substr(colnames(mymatrix)[121:220],1,4))

## x freq  
## 1 Chri 16  
## 2 Eric 15  
## 3 John 11  
## 4 Matt 5  
## 5 Mihi 6  
## 6 Parv 6  
## 7 Step 11  
## 8 Tim\_ 10  
## 9 Tony 8  
## 10 Yous 12

count(substr(colnames(mymatrix[,get\_order(mat2)])[121:220],1,4))

## x freq  
## 1 Chri 90  
## 2 Eric 2  
## 3 Mihi 4  
## 4 Step 1  
## 5 Tim\_ 2  
## 6 Yous 1

## this is chris’ cluster

count(substr(colnames(mymatrix)[300:400],1,4))

## x freq  
## 1 Chri 20  
## 2 Eric 11  
## 3 John 12  
## 4 Matt 8  
## 5 Mihi 10  
## 6 Parv 4  
## 7 Step 10  
## 8 Tim\_ 9  
## 9 Tony 11  
## 10 Yous 6

count(substr(colnames(mymatrix[,get\_order(mat2)])[300:400],1,4))

## x freq  
## 1 Chri 1  
## 2 John 18  
## 3 Matt 6  
## 4 Mihi 4  
## 5 Step 28  
## 6 Tim\_ 22  
## 7 Tony 10  
## 8 Yous 12

## things go off the rails after chris  
## it’s probably because eric and chris have the most documents  
## so the top 2000 words are dominated by video game and political  
## words. The method has promise, but we need to do a better job of  
## building the word list

list\_seriation\_methods("matrix")

## [1] "BEA" "BEA\_TSP" "Identity" "PCA" "PCA\_angle" "Random"

## previous method did not use a cosine distance measurement. That may have   
## hurt us. Let’s look at a second seriation where we order the distance  
## matrix

d<-dist(t(mymatrix))  
  
s <- seriate(d,"TSP")  
s

## object of class 'ser\_permutation', 'list'  
## contains permutation vectors for 1-mode data  
##   
## vector length seriation method  
## 1 1292 TSP

get\_order(s)[1:10]

## [1] 1058 270 568 634 317 627 418 322 850 556

t(mymatrix)[get\_order(s)[1:10],1:10]

## V2 V3 V4 V5 V6 V7 V8 V9 V10 V11  
## Parvati\_1057 0 0 0 0 0 0 0 0 0 0  
## Parvati\_269 0 0 0 0 0 0 0 0 0 0  
## Eric\_567 0 7 1 0 0 3 0 0 0 0  
## Matt\_633 0 0 0 0 0 0 3 0 0 0  
## Tim\_316 0 0 0 0 0 7 0 0 0 0  
## Tim\_626 0 0 0 0 0 7 0 0 0 0  
## Tim\_417 0 0 0 0 0 3 0 145 0 0  
## Tim\_321 0 0 0 0 0 3 0 145 0 0  
## Tim\_849 0 0 0 0 0 2 0 31 0 0  
## Tim\_555 0 0 0 0 0 2 0 31 0 0

## this is only 10 out of 1292 rows and 10 out of 2000 columns  
## but it seems to be doing the right thing by clustering all  
## of the documents that mention word v7 and v9