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))
```

```r
library(seriation)
```

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

```r
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
```

```r
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"
```

```r
## 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
```