Music Recommendation System

Yiwen Mei

2024-03-18

```
# Recommendation System
# a) Data Prepocessing
library(arules)
```

```
## Loading required package: Matrix
```

```
## 
## Attaching package: ’arules’
```

```
## The following objects are masked from ’package:base’:
## 
##     abbreviate, write
```

```
df = read.csv("music.csv", stringsAsFactors=T)
str(df)
```

```
## ’data.frame’:    289955 obs. of  4 variables:
##  $ customer: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ artist  : Factor w/ 1004 levels "...and you will know us by the trail of dead",..: 715 851 374 28
##  $ sex     : Factor w/ 2 levels "f","m": 1 1 1 1 1 1 1 1 1 1 ...
##  $ country : Factor w/ 159 levels "Afghanistan",..: 58 58 58 58 58 58 58 58 58 58 ...
```

```
df$customer <- factor(df$customer)
head(df, 19)
```

```
##    customer                  artist sex       country
## 1         1   red hot chili peppers   f       Germany
## 2         1 the black dahlia murder   f       Germany
## 3         1               goldfrapp   f       Germany
## 4         1        dropkick murphys   f       Germany
## 5         1                le tigre   f       Germany
## 6         1              schandmaul   f       Germany
## 7         1                   edguy   f       Germany
## 8         1            jack johnson   f       Germany
## 9         1               eluveitie   f       Germany
## 10        1             the killers   f       Germany
## 11        1            judas priest   f       Germany
## 12        1              rob zombie   f       Germany
## 13        1              john mayer   f       Germany
## 14        1                 the who   f       Germany
## 15        1              guano apes   f       Germany
## 16        1      the rolling stones   f       Germany
## 17        3        devendra banhart   m United States
## 18        3        boards of canada   m United States
## 19        3               cocorosie   m United States
```

```
df = df[, -c(3, 4)]
original = dim(df)[1]
# remove all rows with unknown artists name from df
df = df[df$artist != "[unknown]",]
new = dim(df)[1]
original - new
```

```
## [1] 553
```

```
# There are 553 songs from unknown artists are in the dataset.

# b) Convert the dataframe into a transactions dataset
# create a list of users (each with the songs of their preference)
playlist = split(df$artist,f=df$customer)
# see first user's songs
playlist[[1]]
```

```
##  [1] red hot chili peppers   the black dahlia murder goldfrapp
##  [4] dropkick murphys        le tigre                schandmaul
##  [7] edguy                   jack johnson            eluveitie
## [10] the killers             judas priest            rob zombie
## [13] john mayer              the who                 guano apes
## [16] the rolling stones
## 1004 Levels: ...and you will know us by the trail of dead [unknown] ... zero 7
```

```
# see first two users' songs
playlist[1:2]
```

```
## $`1`
##  [1] red hot chili peppers   the black dahlia murder goldfrapp
##  [4] dropkick murphys        le tigre                schandmaul
##  [7] edguy                   jack johnson            eluveitie
## [10] the killers             judas priest            rob zombie
## [13] john mayer              the who                 guano apes
## [16] the rolling stones
## 1004 Levels: ...and you will know us by the trail of dead [unknown] ... zero 7
##
## $`3`
##  [1] devendra banhart    boards of canada    cocorosie
##  [4] aphex twin          animal collective   atmosphere
##  [7] joanna newsom       air                 portishead
## [10] massive attack      broken social scene arcade fire
## [13] plaid               prefuse 73          m83
## [16] the flashbulb       pavement            goldfrapp
## [19] amon tobin          sage francis        four tet
## [22] max richter         autechre            radiohead
## [25] neutral milk hotel  beastie boys        aesop rock
## [28] mf doom             the books
## 1004 Levels: ...and you will know us by the trail of dead [unknown] ... zero 7
```

```
# remove song duplicates, if any
playlist = lapply(playlist,unique)
# convert the list of users to transactions object
df.trans <- as(playlist,"transactions")
inspect(head(df.trans,2))
```

```
##     items                      transactionID
## [1] {dropkick murphys,
##      edguy,
##      eluveitie,
##      goldfrapp,
##      guano apes,
##      jack johnson,
##      john mayer,
##      judas priest,
##      le tigre,
##      red hot chili peppers,
##      rob zombie,
##      schandmaul,
##      the black dahlia murder,
##      the killers,
##      the rolling stones,
##      the who}                              1
## [2] {aesop rock,
##      air,
##      amon tobin,
##      animal collective,
##      aphex twin,
##      arcade fire,
##      atmosphere,
##      autechre,
##      beastie boys,
##      boards of canada,
##      broken social scene,
##      cocorosie,
##      devendra banhart,
##      four tet,
##      goldfrapp,
##      joanna newsom,
##      m83,
##      massive attack,
##      max richter,
##      mf doom,
##      neutral milk hotel,
##      pavement,
##      plaid,
##      portishead,
##      prefuse 73,
##      radiohead,
##      sage francis,
##      the books,
##      the flashbulb}                        3
```

```
length(playlist[[1]])
```

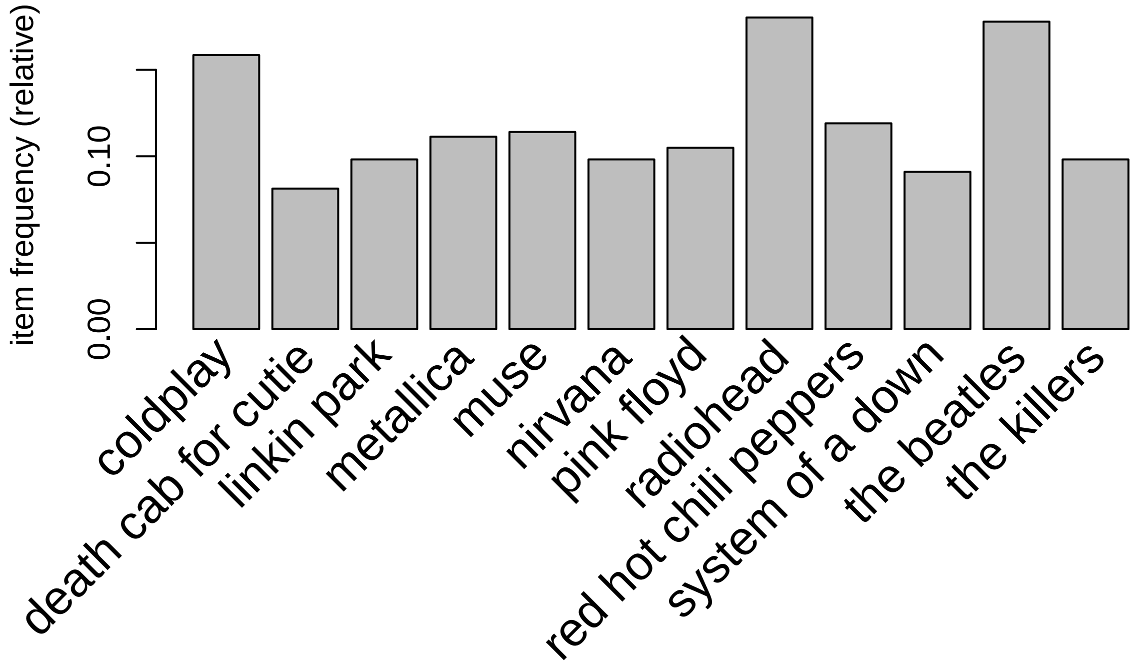
```
## [1] 16
```

```
length(playlist[[2]])
```

```
## [1] 29
```

```
# The first transaction includes 16 artists.
# The second transaction includes 29 artists.

# c)
# display barplot showing all artists downloaded by more than 8% of the customers (support at least 0.0
itemFrequencyPlot(df.trans,support=.08,cex.names=1.5)
```



```
# radiohead is the most popular artist.

# d)
list1 = list(supp=0.01, conf=0.5,target="rules")
rules1 <- apriori(playlist, parameter=list1)
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 150
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[1003 item(s), 15000 transaction(s)] done [0.04s].
## sorting and recoding items ... [654 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.01s].
## writing ... [50 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
rules_sorted = sort(rules1, by = "lift")
inspect(rules_sorted)
```

```
##      lhs                                     rhs            support
## [1]  {the pussycat dolls}                 => {rihanna}      0.01040000
## [2]  {t.i.}                               => {kanye west}   0.01040000
## [3]  {judas priest}                       => {iron maiden}  0.01353333
## [4]  {sonata arctica}                     => {nightwish}    0.01346667
## [5]  {pink floyd, the doors}              => {led zeppelin} 0.01066667
## [6]  {led zeppelin, the doors}            => {pink floyd}   0.01066667
## [7]  {megadeth}                           => {metallica}    0.01626667
## [8]  {placebo, radiohead}                 => {muse}         0.01366667
## [9]  {oasis, the killers}                 => {coldplay}     0.01113333
## [10] {keane}                              => {coldplay}     0.02226667
## [11] {radiohead, snow patrol}             => {coldplay}     0.01006667
## [12] {snow patrol, the killers}           => {coldplay}     0.01040000
## [13] {death cab for cutie, the killers}   => {coldplay}     0.01086667
## [14] {oasis, radiohead}                   => {coldplay}     0.01273333
## [15] {sigur rós, the beatles}             => {radiohead}    0.01046667
## [16] {travis}                             => {coldplay}     0.01373333
## [17] {coldplay, the smashing pumpkins}    => {radiohead}    0.01093333
## [18] {bob dylan, pink floyd}              => {the beatles}  0.01033333
## [19] {the beatles, the smashing pumpkins} => {radiohead}    0.01146667
## [20] {the beatles, the killers}           => {coldplay}     0.01253333
## [21] {bob dylan, the rolling stones}      => {the beatles}  0.01146667
## [22] {snow patrol}                        => {coldplay}     0.02646667
## [23] {radiohead, the killers}             => {coldplay}     0.01506667
## [24] {bloc party, the killers}            => {coldplay}     0.01106667
## [25] {radiohead, u2}                      => {coldplay}     0.01140000
## [26] {beck, the beatles}                  => {radiohead}    0.01300000
## [27] {oasis, the beatles}                 => {coldplay}     0.01060000
## [28] {the fray}                           => {coldplay}     0.01126667
## [29] {led zeppelin, the rolling stones}   => {the beatles}  0.01066667
## [30] {david bowie, pink floyd}            => {the beatles}  0.01006667
## [31] {bob dylan, radiohead}               => {the beatles}  0.01386667
## [32] {coldplay, sigur rós}                => {radiohead}    0.01206667
## [33] {red hot chili peppers, the killers} => {coldplay}     0.01086667
## [34] {muse, the killers}                  => {coldplay}     0.01513333
## [35] {david bowie, the rolling stones}    => {the beatles}  0.01000000
## [36] {radiohead, the rolling stones}      => {the beatles}  0.01060000
## [37] {the beatles, the shins}             => {radiohead}    0.01066667
## [38] {the beatles, the strokes}           => {radiohead}    0.01046667
## [39] {broken social scene}                => {radiohead}    0.01506667
## [40] {the kinks}                          => {the beatles}  0.01360000
## [41] {led zeppelin, radiohead}            => {the beatles}  0.01306667
## [42] {simon & garfunkel}                  => {the beatles}  0.01540000
## [43] {the flaming lips}                   => {radiohead}    0.01306667
## [44] {david bowie, radiohead}             => {the beatles}  0.01393333
## [45] {blur}                               => {radiohead}    0.01753333
## [46] {pink floyd, the doors}              => {the beatles}  0.01000000
## [47] {beck}                               => {radiohead}    0.02926667
## [48] {muse, the beatles}                  => {radiohead}    0.01380000
## [49] {death cab for cutie, the shins}     => {radiohead}    0.01006667
## [50] {death cab for cutie, the beatles}   => {radiohead}    0.01246667
##      confidence coverage   lift      count
## [1]  0.5777778  0.01800000 13.415893 156
## [2]  0.5672727  0.01833333  8.854413 156
## [3]  0.5075000  0.02666667  8.562992 203
## [4]  0.5101010  0.02640000  8.236292 202
## [5]  0.5387205  0.01980000  6.802027 160
## [6]  0.5970149  0.01786667  5.689469 160
## [7]  0.5281385  0.03080000  4.743759 244
## [8]  0.5137845  0.02660000  4.504247 205
## [9]  0.6626984  0.01680000  4.180183 167
## [10] 0.6374046  0.03493333  4.020634 334
## [11] 0.6344538  0.01586667  4.002021 151
## [12] 0.5954198  0.01746667  3.755802 156
## [13] 0.5884477  0.01846667  3.711823 163
## [14] 0.5876923  0.02166667  3.707058 191
## [15] 0.6434426  0.01626667  3.569393 157
## [16] 0.5628415  0.02440000  3.550304 206
## [17] 0.6283525  0.01740000  3.485683 164
## [18] 0.6150794  0.01680000  3.458092 155
## [19] 0.6209386  0.01846667  3.444556 172
## [20] 0.5340909  0.02346667  3.368950 188
## [21] 0.5910653  0.01940000  3.323081 172
## [22] 0.5251323  0.05040000  3.312441 397
## [23] 0.5243619  0.02873333  3.307582 226
## [24] 0.5236593  0.02113333  3.303150 166
## [25] 0.5213415  0.02186667  3.288529 171
## [26] 0.5909091  0.02200000  3.277972 195
## [27] 0.5196078  0.02040000  3.277594 159
## [28] 0.5168196  0.02180000  3.260006 169
## [29] 0.5776173  0.01846667  3.247474 160
## [30] 0.5741445  0.01753333  3.227949 151
## [31] 0.5730028  0.02420000  3.221530 208
## [32] 0.5801282  0.02080000  3.218167 181
## [33] 0.5093750  0.02133333  3.213047 163
## [34] 0.5089686  0.02973333  3.210483 227
## [35] 0.5703422  0.01753333  3.206572 150
## [36] 0.5638298  0.01880000  3.169958 159
## [37] 0.5673759  0.01880000  3.147425 160
## [38] 0.5607143  0.01866667  3.110471 157
## [39] 0.5472155  0.02753333  3.035589 226
## [40] 0.5298701  0.02566667  2.979030 204
## [41] 0.5283019  0.02473333  2.970213 196
## [42] 0.5238095  0.02940000  2.944956 231
## [43] 0.5297297  0.02466667  2.938589 196
## [44] 0.5225000  0.02666667  2.937594 209
## [45] 0.5228628  0.03353333  2.900496 263
## [46] 0.5050505  0.01980000  2.839489 150
## [47] 0.5092807  0.05746667  2.825152 439
## [48] 0.5073529  0.02720000  2.814458 207
## [49] 0.5033333  0.02000000  2.792160 151
## [50] 0.5013405  0.02486667  2.781105 187
```

```
# I would suggest radiohead to a customer who listened to the beattles and the smashing punmpkins.

# e)
rules_sorted_new = sort(subset(rules1, lift > 5), by = "confidence")
inspect(rules_sorted_new)
```

```
##     lhs                            rhs             support    confidence
## [1] {led zeppelin, the doors} => {pink floyd}    0.01066667 0.5970149
## [2] {the pussycat dolls}      => {rihanna}       0.01040000 0.5777778
## [3] {t.i.}                    => {kanye west}    0.01040000 0.5672727
## [4] {pink floyd, the doors}   => {led zeppelin} 0.01066667 0.5387205
## [5] {sonata arctica}          => {nightwish}     0.01346667 0.5101010
## [6] {judas priest}            => {iron maiden}   0.01353333 0.5075000
##     coverage   lift      count
## [1] 0.01786667  5.689469 160
## [2] 0.01800000 13.415893 156
## [3] 0.01833333  8.854413 156
## [4] 0.01980000  6.802027 160
## [5] 0.02640000  8.236292 202
## [6] 0.02666667  8.562992 203
```

```
# I would suggest iron maiden to a customer who listened to judas priest.
```