15.071 Analytics Edge - Homework Assignment #7 Collaborative Filtering

Problem-1.

a.) By looking at the centering values, the three most popular songs are:

songID	songName	year	artist	genre	songCentering
54	You're The One	1990	Dwight Yoakam	Country	1.718345
26	Undo	2001	Bjork	Rock	1.693505
439	Secrets	2009	OneRepublic	Rock	1.639106

By again comparing the centering values, the three users who are most enthused are:

userID	userCentering
1540	0.5863062
838	0.5038647
1569	0.5019513

b.) The best number of archetypes to consider is 3; since for k=3, the highest R-squared value is achieved as shown in the table below:

R output:

```
> rank.info$info
   rank        r2      SSE      RMSE       MAE
1     1 0.3086000 764.3149 0.2345599 0.1759665
2     2 0.3231645 748.2145 0.2320762 0.1730743
3     3 0.3307980 739.7759 0.2307638 0.1704053 → The highest r2 value for k=3
4     4 0.3240683 747.2153 0.2319212 0.1706654
5     5 0.3233732 747.9838 0.2320405 0.1700171
6     6 0.3076614 765.3525 0.2347191 0.1702432
7     7 0.3058033 767.4066 0.2350338 0.1697900
8     8 0.2810630 794.7560 0.2391853 0.1716516
9     9 0.2599380 818.1088 0.2426739 0.1727793
10   10 0.2525903 826.2313 0.2438757 0.1729414
```

This number (k=3) represents the number of archetypes that yields the best prediction on the existing data, i.e. the prediction with the highest R-squared value of 0.3307980. An archetypal user can basically be thought of as an idealized stereotype of certain music listeners.

The calculated best number of archetypes =3 is lower than I would initially guess, considering we have 7 different genres. However apparently selecting an archetype number higher than 3 in this case causes overfitting in our predictive model according to the cross validation results. Choosing too few or too many archetypes lowers the R-squared value (i.e. predictive power) of the model.

c.) The out-of-sample performance (OSR-squared value) of our fitted model on the testing data set is **0.3263433 = ~0.326** This is close the R-squared value we have calculated in the question above which is a consistent result and also shows we did suffer from overfitting.

R output:

```
> r2_test<-1-sum((pred_correct-songs_ratings.test[,3])^2) /
+   sum((mean(songs_ratings.train[,3]) - songs_ratings.test[,3])^2
```

```
> r2_test
[1] 0.3263433
```

d.) Below is the table that summarizes each archetype's average preferences by music genre:

	genre	a1	a2	a3
1	Country	0.044	0.054	-0.001
2	Electronic	0.013	-0.004	-0.029
3	Folk	-0.016	0.003	-0.014
4	Pop	-0.005	-0.001	0.002
5	Rap	0.002	0.004	-0.024
6	RnB	0.006	0.001	-0.017
7	Rock	0.004	-0.004	-0.008

In my opinion, I can say archetype-2 (a2) is the easiest to interpret since a2 shows a strong positive preference for Country genre and relatively low preference (positive or negative) for other genres. I can say archetype-2 is a fan of Country genre.

Another outstanding observation is that archetype-3 (a3) shows positive preference only for Pop genre, and shows negative preference for (dislikes) all other genres. So a3 is also easy to interpret in this way.

e.) Colin's scores for each archetype are summarized below:

Archetype #	Colin's Score
a1	0.344
a2	10.543
a3	3.131

Based on these scores, I would describe Colin is most similar to archetype-2 which has strong preference for Country genre. Thus I would expect Colin likes the Country music most among all genres.

R output:

```
> fit$u[userNum,]*fit$d
[1]  0.3443222 10.5429473  3.1306233
```

f.) Colin's score for song-id:131 is **1.67** and his score for song-id:270 is **1.06.** Thus clearly Colin prefers song#131 over song#270.

We can look at the songs database to better interpret and verify this result. This result is consistent with our findings from the previous question, because as we observe from the Songs database, Song#131 ("She's Good Or You") is a Country genre song whereas Song#270 ("Around the World") is an Electronic genre song.
We know that Colin is more of an archetype#2 user which has strong preference for Country genre, so it is expected that Colin's score for the Country song#131 would be higher.

songID	songName	year	artist	genre
131	She's Good For You	2005	Deana Carter	Country
270	Around The World (Radio Edit)	1996	Daft Punk	Electronic

R output:

```
> print(c(manual_pred,auto_pred))
[1] 1.670886 1.670886
> print(c(manual_pred2,auto_pred2))
[1] 1.060232 1.060232
```

g.) Optional Question

The following 5 songs (among the songs she has not yet listened yet) would be at the top of Daisy's playlist:

songID	songName	year	artist	genre	pred
439	Secrets	2009	OneRepublic	Rock	1.851483
630	Marry Me	2009	Train	Pop	1.629435
368	Bulletproof	2009	La Roux	Pop	1.502435
562	Alejandro	2009	Lady GaGa	Pop	1.489427
221	Livin' On A Prayer	1986	Bon Jovi	Rock	1.471996

I would describe Daisy's music preference to be oriented on Rock and Pop genres. Pop and Rock are the genres that Daisy likes the most.

We can look deeper into this by looking at Daisy's score for each archetype (just like we did for Colin). The table below summarize Daisy's score on each archetype:

Archetype #	Daisy's Score
a1	-0.942
a2	-1.081
a3	1.154

This shows that Daisy is most similar to Arcetype#3 (a3) and somewhat opposite of Arcetype#2 (a2) and Arcetype#1(a1). This is relatively consistent with our results, since a3 has positive preference for pop genre.

R output:

```
> fit$u[userNum2,]*fit$d
[1] -0.9423239 -1.0812300  1.1542138
```

APPENDIX: R-Code

```
### HW-07 Collaborative Filtering for Recommendations Songs to Music Listeners
### PART-A: Finding the most popular songs and most active users:

library(dplyr)
install.packages("softImpute")
library(softImpute)
install.packages('Matrix')
library(Matrix)

source("functions_HW7.R")

songs_ratings<-read.csv("MusicRatings.csv")
songs<-read.csv("Songs.csv")
users<-read.csv("Users.csv")

head(songs_ratings,5)
head(songs,5)
head(users,5)

training.rows <- cf.training.set(songs_ratings$userID,
                                 songs_ratings$songID, prop=0.96)

training.rows[1:10]
# so we can use it to define the actual training and test sets:
songs_ratings.train <- songs_ratings[training.rows,]
songs_ratings.test <- songs_ratings[-training.rows,]

mat.train <- Incomplete(songs_ratings.train[,1],
                        songs_ratings.train[,2],
                        songs_ratings.train[,3])

head(mat.train,5)

set.seed(15071)

mat.centered <- biScale(mat.train, maxit=1000,
                       row.scale = FALSE, col.scale = FALSE)

# to get the most popular songs:

song_pop<-attr(mat.centered, "biScale:column")$center

songsPlus<-songs %>% mutate(songCentering=song_pop)

topsongs<-songsPlus[order(-songsPlus$songCentering),]
topsongs[c(1:5),]

# to get the most active users:
active_users<-attr(mat.centered, "biScale:row")$center

usersPlus<-users %>% mutate(userCentering=active_users)
topUsers<-usersPlus[order(-usersPlus$userCentering),]
topUsers[c(1:5),]

### PART-B: Determining the optimal number of archetypes
```

```
rank.info <- cf.evaluate.ranks(dat=songs_ratings.train, ranks=1:10,
prop.validate=0.05)

rank.info$info

### PART-C: Finding Out-of-sample performance of the model on the test set

set.seed(15071)

fit <- softImpute(x=mat.centered, rank.max=3, lambda=0, maxit=1000)

pred_test<-impute(fit, songs_ratings.test[,1], songs_ratings.test[,2])

minimum <- min(songs_ratings.train[,3])
maximum <- max(songs_ratings.train[,3])

pred_correct<-pmin(pmax(pred_test, minimum), maximum)

r2_test<-1-sum((pred_correct-songs_ratings.test[,3])^2) /
  sum((mean(songs_ratings.train[,3]) - songs_ratings.test[,3])^2)

r2_test

### PART-D: Interpreting our archetypes:

head(fit$v,5)

fit_arch<-as.data.frame(fit$v)

songsWithArch <- songs %>% mutate(Arch1=fit_arch[,1], Arch2=fit_arch[,2],
Arch3=fit_arch[,3])

arch_summary<-songsWithArch %>% group_by(genre) %>%
  summarize(a1=mean(Arch1),a2=mean(Arch2),a3=mean(Arch3))

arch_summary[-1]<-round(arch_summary[,-1],3)

View(arch_summary)


### PART-E Predicting a user's weights on the different archetypes:

# Let us first define a user number:
userNum=1251 # Set Collin

fit$d

fit$u[userNum,]
head(fit$u,5)

fit$u[userNum,]*fit$d

# Looking at the archetypes' weights on each genre, we can get a sense
# of this user's movie preferences!

# PART-F Predicting a user's preference for songs:

userNum=1251  ## User name Collin
```

```
songNum=131  ##  Song ID: 131

manual_pred<-usersPlus[userNum,]$userCentering + songsPlus[songNum,]$songCentering +
  sum(fit$u[userNum,]*fit$d*fit$v[songNum,])

auto_pred<-impute(fit,userNum,songNum)

print(c(manual_pred,auto_pred))

songNum2=270 ## Repeat for Song270

manual_pred2<-usersPlus[userNum,]$userCentering + songsPlus[songNum2,]$songCentering +
  sum(fit$u[userNum,]*fit$d*fit$v[songNum2,])

auto_pred2<-impute(fit,userNum,songNum2)

print(c(manual_pred2,auto_pred2))

### PART-G: 9. Predicting top unwatched/unlistened songs for a user

# Pick user for which we'll predict top movies
userNum2=1584 ### User name Daisy

Mat_all<-Incomplete(songs_ratings[,1], songs_ratings[,2], songs_ratings[,3])

notWatched = (Mat_all[userNum2,]==0)
songs_notWatched = songs[notWatched,]

pred_for_user<-sapply(songs_notWatched$songID, function(x) impute(fit,userNum2,x))

songs_notWatched <- songs_notWatched %>% mutate(pred=pred_for_user)

topPrefs<-songs_notWatched[order(-songs_notWatched$pred),]
head(topPrefs,5)

### Also let's find Daisy's scores for each archetype to see consistency with our
results:

userNum2=1584 # Set Daisy

fit$d

fit$u[userNum2,]
head(fit$u,5)

fit$u[userNum2,]*fit$d

### End of the Code
```