

Homework 7

Aaron Politsky

November 15, 2015

1.

Find the user who has rated the most amount of video games.

```
which.max(rowCounts(ratingData))
```

```
## U584295664
##      1593
```

2.

Which video game has been rated by the most amount of users?

```
which.max(colCounts(ratingData))
```

```
## I760611623
##      489
```

3.

Find the user who is most similar to “U141954350”

```
user <- ratingData[rownames(ratingData)=="U141954350",]
getData.frame(user)
```

```
##      user      item rating
## 1 U141954350 I236414343      1
## 2 U141954350 I279487841      1
## 3 U141954350 I465121818      1
## 4 U141954350 I500530161      1
## 5 U141954350 I640618913      1
## 6 U141954350 I761586720      1
## 7 U141954350 I791430645      1
## 8 U141954350 I907103993      1
## 9 U141954350 I923148766      1
## 10 U141954350 I981874545      1
```

```
rowSds(user)
```

```
## U141954350
##      0
```

This user is unique in that he has rated 10 games, but has rated them all the same. His de-meaned data would be 0. His pearson correlation would be undefined.

```
rdn = normalize(ratingData)
user <- rdn['U141954350',]

similarity.matrix <- similarity(x=user,
                               y=rdn[rownames(rdn)!='U141954350',],
                               which="users",
                               method = "cosine")
```

The user with the most similarity to U141954350 is U007042545

This isn't saying a whole lot, since their common support is only over one item:

```
getData.frame(rdn['U007042545',,]) [
  getData.frame(rdn['U007042545'])$item %in% getData.frame(rdn['U141954350'])$item,]
```

```
##          user      item rating
## 1 U007042545 I791430645      0
```

4.

Recommend a game for our user.

Let's try user- and item-based collaborative filtering.

```
ubcf.rec.model <-
  Recommender(ratingData[rownames(ratingData)!='U141954350',],
              method="UBCF",
              param=list(normalize="Z-score", method="cosine",nn=10))
ubcf.recommendation <-
  predict(ubcf.rec.model, ratingData['U141954350',], type="topNList", n=10)
as(ubcf.recommendation, "matrix")[1,1:11]
```

```
## I000034290 I004580124 I006362038 I008874402 I009101756 I009224134
##          1          1          1          1          1          1
## I011188977 I013267561 I014540332 I015123815 I015265292
##          1          1          1          1          0
```

```
ibcf.rec.model <-
  Recommender(ratingData, method="IBCF",
              param=list(normalize="Z-score", method="Pearson"))
ibcf.recommendation <-
  predict(ibcf.rec.model, ratingData['U141954350',], type="topNList", n=5)
as(ibcf.recommendation, "list")
```

```
## [[1]]
## character(0)
```

I wonder if the fact that the item-based collaboration returns an empty list may have to do with how our user has rated everything the same and our use of Pearson similarity. Or, since not many users have rated games that our user has, perhaps there isn't much coming out of the similarity-weighted ratings average.

Let's try latent popularity and latent-factor recommendation models:

```
pop.rec.model <-  
  Recommender(ratingData, method = "POPULAR")  
pop.recommendation <-  
  predict(pop.rec.model, ratingData['U141954350',], type="topNList", n=5)  
as(pop.recommendation, "list")  
  
## [[1]]  
## [1] "I760611623" "I936489406" "I962238428" "I232939121" "I357170215"  
  
lf.rec.model <-  
  Recommender(ratingData,  
    method='SVD',           # Item-Based Collaborative Filtering  
    parameter=list(  
      categories=25,        # number of latent factors  
      normalize = "Z-score",  
      treat_na = "median"  
    ))  
lf.recommendation <-  
  predict(lf.rec.model, ratingData, type="ratings")  
lf.list <- as(lf.recommendation, "list")  
which.max(lf.list[['U141954350']])  
  
## I000034290  
##          1
```

Latent-factor modeling agrees with user-based collaborative filtering, but not with popularity. (The latter makes sense, because it seems that our user doesn't rate commonly-rated games.)