Project3_Data-612- Matrix Factorization

Samriti Malhotra, Vishal Arora

June 27, 2019

Contents

Objective :- **1**
Solution:- 1
Libraries used 1
Data loading , preperation of relevant dataset 1
Data Preperation 2
Building the Item-based Collaborative Filtering Model (IBCF) and RMSE for IBCF model. . 2
Building the User-based Collaborative Filtering Model (UBCF) and then evluate the RMSE for UBCF model 2
Building SVD model 3

Summary **3**

Objective :-

The goal of this assignment is give you practice working with Matrix Factorization techniques. The task is implement a matrix factorization method-such as singular value decomposition (SVD) or Alternating Least Squares (ALS)-in the context of a recommender system.

Solution:-

We took this dataset ml-latest-small.zip from Movie Lens site which describes 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service. It contains 100836 ratings and 3683 tag applications across 9742 movies. These data were created by 610 users between March 29, 1996 and September 24, 2018. This dataset was generated on September 26, 2018.

> Citation :- F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1-19:19. https://doi.org/10.1145/2827872

Libraries used

recommenderlab
dplyr
reshape2

Data loading , preperation of relevant dataset

Data is loaded from the github, and then selecting the columns to create a matrix which is a class of realRatingMatrix. As our matrix doesn't have any NA that means every user has seen every movie and provided ratings but all of them may not be relevant.

```
ratings <- read.csv("https://raw.githubusercontent.com/Vishal0229/DATA612_RecommenderSystem/master/Proj
titles <- read.csv("https://raw.githubusercontent.com/Vishal0229/DATA612_RecommenderSystem/master/Proje

ratings <- ratings %>% select(userId, movieId, rating)
```

```
#converting the ratings data frame into userId-movieId matrix
ratingDT <- acast(ratings, userId~movieId, value.var="rating")

#convert matrix into realRatingMatrix using recommenderLab package
ratingDT <- as(as.matrix(ratingDT), "realRatingMatrix")
dim(ratingDT)
```

```
## [1]  610 9724
```

Data Preperation

1) Select the relevant data

2) Normalize the data

As rule of thumb for beginning user who rating more than 100 movies and movies which have been watched more than 100 time. those are the ones we going to take initially.

```
ratings_movies <- ratingDT[rowCounts(ratingDT)>100, colCounts(ratingDT)>100]

dim(ratings_movies)
```

```
## [1] 245 134
```

Now the dataset has reduced but still it is a large dataset may be we might have to take a smaller dataset for SVD evluation.Lets first do the evaulation using IBCF & UBCF algorithms and compare it with the SVD to see which one has the least RMSE.

Building the Item-based Collaborative Filtering Model (IBCF) and RMSE for IBCF model.

Taking a subset of the relevant dataset ,as the memory imprint was too high and iyt was taking time to build the recommender model.

```
rating_movies <- as(ratings_movies, "realRatingMatrix")
rm()
set.seed(88)
eval_sets <- evaluationScheme(data = rating_movies, method = "split", train = 0.8, given = -1, goodRati

#IBCF
eval_recommender_ibcf <- Recommender(data = getData(eval_sets, "train"), method = "IBCF", parameter = NU
eval_prediction_ibcf <- predict(object = eval_recommender_ibcf, newdata = getData(eval_sets, "known"), n
calcPredictionAccuracy(x = eval_prediction_ibcf, data = getData(eval_sets, "unknown"), byUser = FALSE)
```

```
##      RMSE       MSE       MAE
## 0.8860376 0.7850626 0.6901797
```

Building the User-based Collaborative Filtering Model (UBCF) and then evluate the RMSE for UBCF model

```
#IBCF
eval_recommender_ubcf <- Recommender(data = getData(eval_sets, "train"), method = "UBCF", parameter = NU
eval_prediction_ubcf <- predict(object = eval_recommender_ubcf, newdata = getData(eval_sets, "known"), n
calcPredictionAccuracy(x = eval_prediction_ubcf, data = getData(eval_sets, "unknown"), byUser = FALSE)
```

```
##      RMSE       MSE       MAE
## 0.8047560 0.6476322 0.6497898
```

Building SVD model

```
svdModel <- Recommender(getData(eval_sets, "train"), method = "SVD", parameter = list(k = 50))
svdPredModel <- predict(svdModel, newdata = getData(eval_sets, "known"), type = "ratings")


calcPredictionAccuracy(x=svdPredModel, getData(eval_sets, "unknown"), byUser = FALSE)
```

```
##      RMSE       MSE       MAE
## 0.7996445 0.6394314 0.6375262
```

Summary

From the above RMSE and other values for various models algorithms we can clearly that SVD is slightly better than UBCF and which in turn is better than IBCF. We can evaluate the svd model by manually calculating SVD(using Base R package) and also SVD can be performed step-by-step with R by calculating ATA and AAT then finding the eigenvalues and eigenvectors of the matrices. However, results can be slightly different than the output of the svd()/recommenderLab . There is a nice article on this (SVD Article Aaron)[https://rpubs.com/aaronsc32/singular-value-decomposition-r].