Literature Review & Data Description

Gopal Narasimhaiah

040703878

Ceni Babaoglu, Ph.D>

February 14th,2022

Ryerson University

Abstract

The pandemic has made the human population stay at home longer which has opened the number of available hours to them to use and consume. One activity that has been accessible is the internet and the usage has increased over time due to the lack of alternative activities and the choice to participate and be stimulated by the offerings of the Internet. From Statistics Canada - Almost half of Canadians (48%) streamed video content, such as Netflix, Crave, news, concerts or fitness videos, more often since the start of the pandemic. New York Times reported 15-17% increase in Netflix and YouTube usage and Forbes states there has been a increase of 12%.

With this trend, there is a place where people will want access to items, products, videos faster and tailored to them. The system that takes this space is called the Recommender systems which has been prevalent these days and avoidable in our daily journey. My project will focus on developing a recommender system which will help individuals retrieve content based on their interests through algorithms that will create choices and customized lists based on that user

Dataset: From GroupsLens (University of Minnesota)

- Available at MovieLens site
    - Movies.csv
    - Ratings.csv
    - Links.csv
    - Tags.csv

Objective: To attain a greater understanding of Collaborative Filters and Contest Based Filter Models for a hybrid model to answer the following question

What performance can collaborative filtering produce movie recommendations based on the related dataset.

DataSet
https://grouplens.org/datasets/movielens/

Github
https://github.com/gnarasim311/cind820capstone

References

https://www150.statcan.gc.ca/n1/pub/45-28-0001/2021001/article/00027-eng.htm

https://www.nytimes.com/interactive/2020/04/07/technology/coronavirus-internet-use.html

https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada

Introduction

In todays evolved Internet Age of Business, having a system in place to predict what item/content/story a user wants is a paramount to organization's success. The Online Giants of Youtube, Netflix, Amazon, Instagram all have highly manicured recommendation systems in which they predict new content that is relevant to their customer base. Netflix which has become a normal web service in a media consumer lives opened up their recommendation system strategy in a form of competition. On 6 October 2006, Netflix, Inc., launched the Netflix Prize, a contest offering US$1m to the first individual or team to develop a recommendation system capable of predicting movie ratings with at least 10% greater accuracy than Cinematch, the company's existing system [1]. Comparable recommendation systems belonging to Amazon, Facebook, Google,Match, Microsoft, Twitter, and other technology-driven companies tend to operate similarly, their inner workings "wired shut" with patent and trade secret laws, non-disclosure agreements, non-compete clauses, and other legal instruments. [1].

Recommendation Systems – use the opinion of members of the community to help the individual within the community identify the information of products most likely to be interesting to them or relevant to their needs [2]

Recommendation Systems Techniques

Collaborative Filtering (CF) - produces recommendations according to the similarities between the active user and other users, or between the target items and similar items that have been rated by the active users [3]

Content Based Filtering (CBF) - , produces recommendations aimed at discovering product attributes and relations between products and between customers [3]

Knowledge Based Approach – produces recommendations based on specific queries that are made by the user, It might prompt the user to give a series of rules or guidelines on what the results should look like, or an example of an item. The system then searches through its database of items and returns similar results. [4]

Hybrid Approach – a blend of all approaches above to generate recommendations. [3]

Literature Review

In this project, the goal is developing a hybrid recommendation system that is based on User Based Collaborative Filter (UBCF) and Item Based Collaborative Filter (IBCF). Recommendations Systems are what make many Online Businesses function.

The recommendation techniques were listed in the introduction and separated into 4 categories. Collaborative filtering (CF) permitting the exploitation of information about user interaction and transactions, such as product ratings and orders. [5]Content Based Filtering is aimed at discovering product attributes and relations between products and between customers. Knowledge Based Filtering is enabling the generation of advice based on explicit human knowledge about the item assortment, user preferences, and recommendation criteria [5] Hybrid Recommendation is based on everything above[5]. In the context of this project of developing a system, we need to understand UBCF and IBCF. User-based collaborative filtering (UBCF) engine follows the “people like you” logic which recommends to a user an item that similar users liked before[6][7][8]. Item-based collaborative filtering (IBCF) follows the logic “if you like this, you might also like that also” and it recommends items that are similar to the ones you previously liked. [6][7][8].

The algorithms that are used but not limited to are based on clustering, the probability of collaborative filtering algorithm, collaborative filtering based on neural networks ,collaborative filtering matrix decomposition based on a variety of models such as the probability model, Bayes model abstraction, maximum entropy model, Gibbs abstract and linear regression.[9][10][11] Even with these algorithms here still exists problems of sparsity, early rater and cold start.[9]

In recommendation engines, cold start refers to the condition when the recommendation system is not yet optimal to generate the best results because of data sparsity i.e. problems in finding an ample number of similar users since in general the active users only scored a small fraction of items.[9][10][11]

To solve the problems of scalability and sparsity in the collaborative filtering, this paper proposed a personalized recommendation approach joins the user clustering technology and item clustering technology[12][13]. Users are clustered based on users’ ratings on items, and each users cluster has a cluster center. Based on the similarity between target user and cluster centers, the nearest neighbors of target user can be found and smooth the prediction where necessary. Then, the proposed approach utilizes the item clustering collaborative filtering to produce the recommendations. The recommendation joining user clustering and item clustering collaborative filtering is more scalable and more accurate than the traditional one. [13][14][15]

To potentially solve the cold start problem using a hybrid method which first cluster items using the rating matrix and then uses the clustering results to build a decision tree to combine items with existing ones. [13][14][15]

DataSet

The dataset that will be used will be MovieLens and is available at MovieLens website which is publicly available.

Dataset – For the interest of this project – used the smallest set available and at the time of the download 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. Movies and Ratings datasets will be used in building a recommender system.

Load the datasets

```
movies <- read.csv("movies.csv")
ratings <- read.csv("ratings.csv")
links <- read.csv("links.csv")
tags <- read.csv("tags.csv")
```

Movie summary

```
dim(movies)

## [1] 9742    3
```

There are 9742 observations and 3 attributes in the movies data-set.

```
head(movies)

##   movieId                               title
## 1       1                    Toy Story (1995)
## 2       2                      Jumanji (1995)
## 3       3             Grumpier Old Men (1995)
## 4       4            Waiting to Exhale (1995)
## 5       5 Father of the Bride Part II (1995)
## 6       6                           Heat (1995)
##                                         genres
## 1 Adventure|Animation|Children|Comedy|Fantasy
## 2                  Adventure|Children|Fantasy
## 3                               Comedy|Romance
## 4                        Comedy|Drama|Romance
## 5                                      Comedy
## 6                       Action|Crime|Thriller

str(movies)

## 'data.frame':    9742 obs. of  3 variables:
##  $ movieId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ title  : chr  "Toy Story (1995)" "Jumanji (1995)" "Grumpier Old Men (19
95)" "Waiting to Exhale (1995)" ...
```

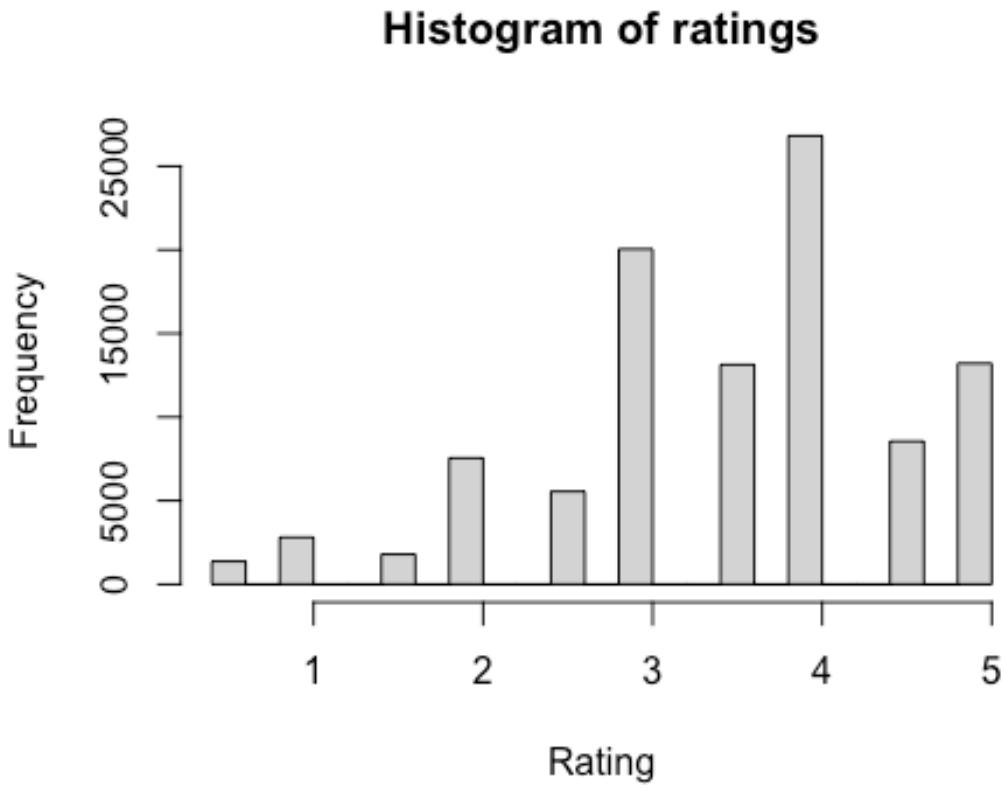
```
##  $ genres : chr  "Adventure|Animation|Children|Comedy|Fantasy" "Adventure|
Children|Fantasy" "Comedy|Romance" "Comedy|Drama|Romance" ...

summary(movies)

##     movieId          title              genres
##  Min.   :     1   Length:9742        Length:9742
##  1st Qu.:  3248   Class :character   Class :character
##  Median :  7300   Mode  :character   Mode  :character
##  Mean   : 42200
##  3rd Qu.: 76232
##  Max.   :193609
```

Rating summary

```
dim(ratings)

## [1] 100836      4
```

There are 100836 observations and 4 attributes in the ratings data-set.

```
head(ratings)

##   userId movieId rating timestamp
## 1      1       1      4 964982703
## 2      1       3      4 964981247
## 3      1       6      4 964982224
## 4      1      47      5 964983815
## 5      1      50      5 964982931
## 6      1      70      3 964982400

hist(ratings$rating, main = "Histogram of ratings", xlab = "Rating")
```




```
str(ratings)

## 'data.frame':    100836 obs. of  4 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : int  1 3 6 47 50 70 101 110 151 157 ...
##  $ rating   : num  4 4 4 5 5 3 5 4 5 5 ...
##  $ timestamp: int  964982703 964981247 964982224 964983815 964982931 96498
2400 964980868 964982176 964984041 964984100 ...

summary(ratings)

##      userId         movieId           rating        timestamp
##  Min.   :  1.0   Min.   :     1   Min.   :0.500   Min.   :8.281e+08
##  1st Qu.:177.0   1st Qu.:  1199   1st Qu.:3.000   1st Qu.:1.019e+09
##  Median :325.0   Median :  2991   Median :3.500   Median :1.186e+09
##  Mean   :326.1   Mean   : 19435   Mean   :3.502   Mean   :1.206e+09
##  3rd Qu.:477.0   3rd Qu.:  8122   3rd Qu.:4.000   3rd Qu.:1.436e+09
##  Max.   :610.0   Max.   :193609   Max.   :5.000   Max.   :1.538e+09
```

Links summary

```
dim(links)

## [1] 9742    3
```

There are 9742 observations and 3 attributes in the link data-set.

```
head(links)
```

```
##   movieId imdbId tmdbId
## 1       1 114709    862
## 2       2 113497   8844
## 3       3 113228  15602
## 4       4 114885  31357
## 5       5 113041  11862
## 6       6 113277    949
```

```
str(links)
```

```
## 'data.frame':    9742 obs. of  3 variables:
##  $ movieId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ imdbId : int  114709 113497 113228 114885 113041 113277 114319 112302 1
14576 113189 ...
##  $ tmdbId : int  862 8844 15602 31357 11862 949 11860 45325 9091 710 ...
```

```
summary(links)
```

```
##     movieId           imdbId            tmdbId
##  Min.   :     1   Min.   :    417   Min.   :     2
##  1st Qu.:  3248   1st Qu.:  95181   1st Qu.:  9666
##  Median :  7300   Median : 167260   Median : 16529
##  Mean   : 42200   Mean   : 677184   Mean   : 55162
##  3rd Qu.: 76232   3rd Qu.: 805568   3rd Qu.: 44206
##  Max.   :193609   Max.   :8391976   Max.   :525662
##                                     NA's   :8
```

Tags summary

```
dim(tags)
```

```
## [1] 3683    4
```

There are 3683 observations and 4 attributes in the tag data-set.

```
head(tags)
```

```
##   userId movieId             tag  timestamp
## 1      2   60756           funny 1445714994
## 2      2   60756 Highly quotable 1445714996
## 3      2   60756    will ferrell 1445714992
## 4      2   89774    Boxing story 1445715207
## 5      2   89774             MMA 1445715200
## 6      2   89774       Tom Hardy 1445715205
```

```
str(tags)
```

```
## 'data.frame':    3683 obs. of  4 variables:
##  $ userId   : int  2 2 2 2 2 2 2 2 2 7 ...
```

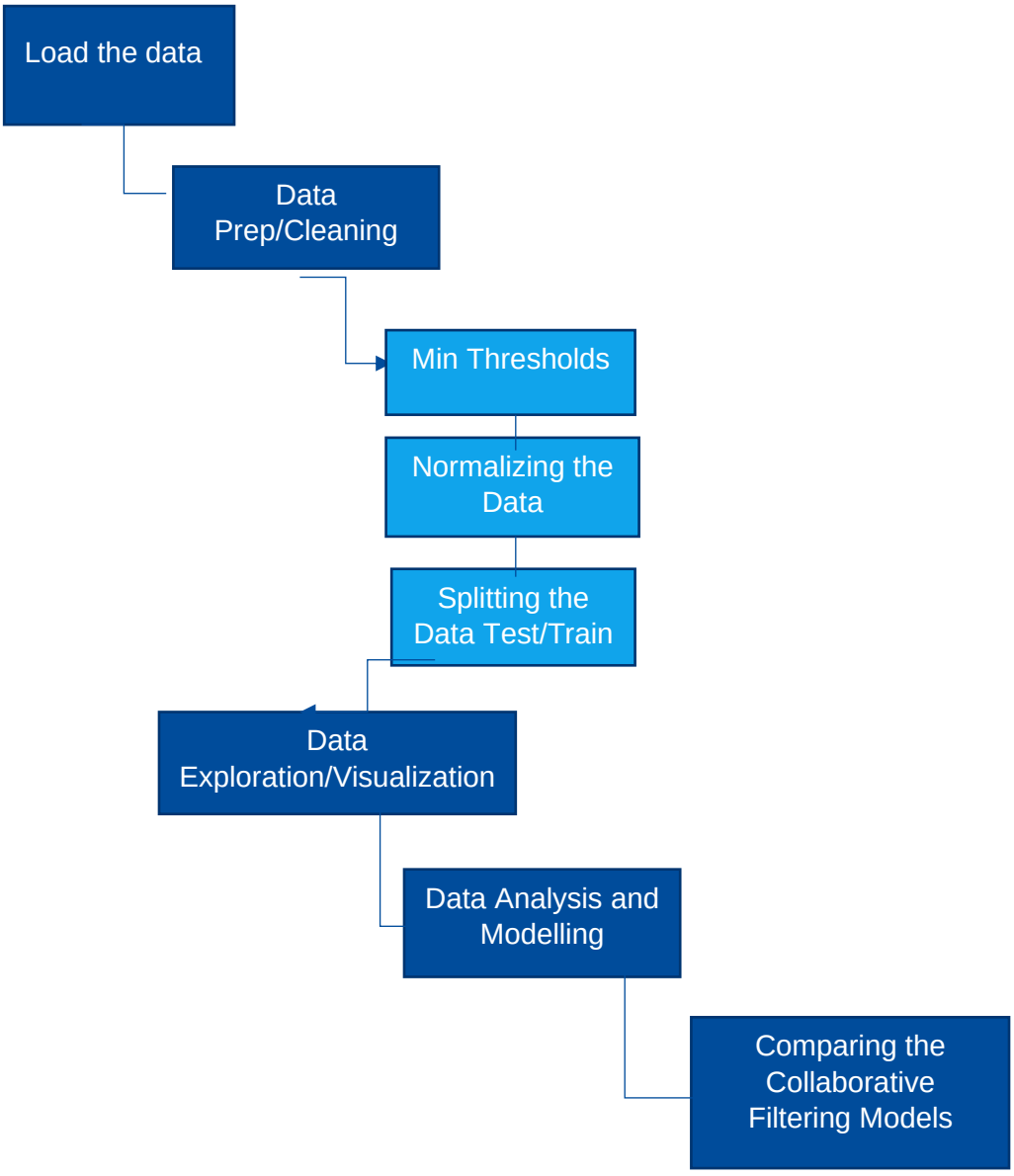
```
##  $ movieId  : int  60756 60756 60756 89774 89774 89774 106782 106782 10678
2 48516 ...
##  $ tag      : chr  "funny" "Highly quotable" "will ferrell" "Boxing story"
 ...
##  $ timestamp: int  1445714994 1445714996 1445714992 1445715207 1445715200
1445715205 1445715054 1445715051 1445715056 1169687325 ...
```

```
summary(tags)
```

```
##      userId         movieId          tag               timestamp
##  Min.   :  2.0   Min.   :     1   Length:3683        Min.   :1.137e+09
##  1st Qu.:424.0   1st Qu.:  1262   Class :character   1st Qu.:1.138e+09
##  Median :474.0   Median :  4454   Mode  :character   Median :1.270e+09
##  Mean   :431.1   Mean   : 27252                      Mean   :1.320e+09
##  3rd Qu.:477.0   3rd Qu.: 39263                      3rd Qu.:1.498e+09
##  Max.   :610.0   Max.   :193565                      Max.   :1.537e+09
```

Building the System/Approach

Below is the framework to build a recommendation system shown graphically



Step 1: Load the Dataset
Loading dataset is split into four files- movies.csv, ratings.csv, links.csv and tags.csv and will be using R language to load and be saved in rmd format.

Step 2: Data Preparation and Cleaning
Minimum Thresholds - limit the input data based on minimum thresholds:

For this project we restrict the model training to those users who have rated at least 50 movies, and those movies that have been rated by at least 100 users

Normalizing the Data – to eliminate bias in data will assign the average rating given by each user to 0 to remove user who give high and low ratings consistently.

Split the Data for Test/Train – Split data in test and train sets so we can test models and compare

Step 3 : Data Exploration/Visualization
Exploring the dataset and deciphering the movie business from this dataset

Step 4: Data Analysis and Modelling
Analysis of UBCF and IBCF models

Step 5: Comparison of Collaborative Filtering Methods
Evaluating collaborative filtering methods based on parameters to determine their effectiveness

References

[1] Hallinan, B., & Striphas, T. (2016). Recommended for you: The Netflix Prize and the production of algorithmic culture. New Media & Society, 18(1), 117–137. https://doi.org/10.1177/1461444814538646

[2] Shardanand, U., Maes, P. 1995. Social information filtering: Algorithms for automating "Word of Mouth". In Proceedings of the Conference on Human Factors in Computing Systems (CHI'95). ACM, New York, 210--217.

[3] Ali A. Amer, Hassan I. Abdalla, Loc Nguyen,Enhancing recommendation systems performance using highly-effective similarity measures,Knowledge-Based Systems,Volume 217,2021, https://doi.org/10.1016/j.knosys

[4] Burke, R. (2000). Knowledge-based recommender systems. *Encyclopedia of library and information systems*, *69*(Supplement 32), 175-186

[5] Min Dong, Xianyi Zeng, Ludovic Koehl, Junjie Zhang,An interactive knowledge-based recommender system for fashion product design in the big data environment,Information Sciences,Volume 540,2020,Pages 469-488

[6] Deng, Z., & Wang, J. (2013). Collaborative Filtering Algorithm Based on User Clustering. Applied Mechanics and Materials, 411-414, 1044. http://dx.doi.org/10.4028/www.scientific.net/AMM.411-414.1044

[7]Ahmad A. Kardan, Mahnaz Ebrahimi,A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups,Information Sciences,Volume 219,2013,Pages 93-110,

[8]Xiaofeng Zhang, Huijie Liu, Xiaoyun Chen, Jingbin Zhong, Di Wang,A novel hybrid deep recommendation system to differentiate user's preference and item's attractiveness,Information Sciences,Volume 519,2020,Pages 306-316

[9] Ajaegbu, C. An optimized item-based collaborative filtering algorithm. J Ambient Intell Human Comput 12, 10629–10636 (2021). https://doi-org.ezproxy.lib.ryerson.ca/10.1007/s12652-020-02876-1

[10] Chen, W., Niu, Z., Zhao, X. et al. A hybrid recommendation algorithm adapted in e-learning environments. World Wide Web 17, 271–284 (2014). https://doi-org.ezproxy.lib.ryerson.ca/10.1007/s11280-012-0187-z

[11] Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J. García Clemente, Gregorio Martínez Pérez,Design of a recommender system based on users' behavior and collaborative location and tracking,Journal of Computational Science,Volume 12,2016,Pages 83-94,

[12]Tahmasebi, F., Meghdadi, M., Ahmadian, S. et al. A hybrid recommendation system based on profile expansion technique to alleviate cold start problem. Multimed Tools Appl 80, 2339–2354 (2021). https://doi-org.ezproxy.lib.ryerson.ca/10.1007/s11042-020-09768-8

[13] Schafer J.B., Frankowski D., Herlocker J., Sen S. (2007) Collaborative Filtering Recommender Systems. In: Brusilovsky P., Kobsa A., Nejdl W. (eds) The Adaptive Web. Lecture Notes in Computer Science, vol 4321. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-72079-9_9

[14] Cite: SongJie Gong, "A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering," Journal of Software vol. 5, no. 7, pp. 745-752, 2010.

[15] Sun, D., Luo, Z., & Zhang, F. (2011, October). A novel approach for collaborative filtering to alleviate the new item cold-start problem. In Communications and Information Technologies (ISCIT), 2011 11th International Symposium on (pp. 402-406). IEEE.