Shahin Ahmed
Student # 500995601

CIND820 Big Data Analytics Project
Ryerson University
Fall 2020

A study of developing a Movie Recommendation System using different Machine Learning Algorithms

Abstract

COVID-19 changes the people lifestyle, due to the lockdown at home, the internet usages are increased, and it impacts on the social life. According to Forbes the initial internet hits increased by between 50% and 70%. Entertainment sites like Netflix, Facebook usages also jumped by 12% approximately.

Considering these changes, Recommendation systems are becoming particularly important in media industry as well as online consumer world. People always prefer to see the items/movies based on their previous search into the system. My project will focus on develop the recommendation system which will help individual to search for content that would be interesting to him/her based on algorithms that sort out all possible choices and create a customize lists of items for that individual.

The dataset I choose for this project is from GroupLens research lab in the University of Minnesota and available in the MovieLens website which contains four files such as movies.csv, ratings.csv, links.csv and tags.csv. The goal of this project is to give a better understanding of User Based Collaborative Filter (**UBCF**) and Item Based Collaborative Filter (**IBCF**) models for hybrid recommender systems by answering the following question:

- With what level of performance can collaborative filtering using UBCF and IBCF models produce movie recommendations based on movies and user's ratings data?

The project will be developed using the R tools.

Dataset link:

MovieLens Latest Datasets

Github Link:

Github Repositories Link

References:

https://www.forbes.com/sites/markbeech/2020/03/25/covid-19-pushes-up-internet-use-70-streaming-more-than-12-first-figures-reveal/#56235ead3104

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

Shahin Ahmed
Student # 500995601

Introduction

In today's competitive online business world, it is important to predicting what item/content a user wants. Netflix, Amazon, YouTube, and Instagram all corporate online business and service provider developed a high standard recommendation system for predicting users new and relevant content. These systems are one of the most valued assets of these companies as demonstrated by the Netflix sponsored competition with a prize of one million dollars to improve on their system [1]. These types of systems are collectively known as recommender systems [2].

Recommender systems usually generate recommendations to users, in one of the following ways:

- **Collaborative filtering algorithms** [3] predict items/products for an active user based on the past data about the other users for same items/products.

- **Content-based algorithms** [4] creates recommendations based on items/products descriptions which could be automatically extracted or manually created, or (and) from user profile which reflects user's interests on that items/products.

- **Knowledge-based algorithms** [5] predicts a recommender system based on specific queries 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.

- **Hybrid approaches** [6] generate recommendations by combining several algorithms or recommendation components, which are based on the above three approaches: collaborative filtering and content-based and knowledge-based algorithms.

Table 1 shows some popular sites which are currently using recommendation system for different purpose [7].

Table 1: Popular sites using recommender systems

Site	**What is recommended**
Amazon	Books/other products
Facebook	Friends/Business/Media
Netflix	Movies
Instagram	Media content

Shahin Ahmed
Student # 500995601

Literature Review

In my project I am going to develop a hybrid recommendation system based on User Based Collaborative Filter (UBCF) and Item Based Collaborative Filter (IBCF). Analysis of Recommendation System is a vast topic, so I studied other's research work closely related to my project work.

In introduction, we classified recommendation system into four types which include memory-based CF, model-based CF, and Hybrid CF ([8],[9],[10]). The memory-based CF techniques explore the entire dataset to find the set of users, which are like the active user ([11],[12]). Thereafter recommendations are made based on the observation of likes and dislikes of the similar users. For similarity computation process, memory-based technique employs various similarity measures which includes PR measure ([13]), Spearman rank correlation (similar to Pearson except rating is rank) ([11]), Kendall's correlation (similar to Spearman rank but instead of rank, relative ranks are used) ([14]). User based CF and Item based CF are classified as Memory-based CF. In users-based CF method ([15],[16]), similar users are found by analyzing other users' preferences against the active user. Other way item-based CF ([17],[18]) approach focuses on finding similarity between items rather than users.

Various machine learning techniques, data mining algorithm are used to develop a user rating model in model-based CF for making predictions. Bayesian model ([19],[20]), clustering models ([21]), rule-based approach ([22]) are some popular algorithms which are used for model-based CF techniques.

The hybrid CF recommendation technique solved the issues find in memory-based and model-based techniques such as cold start, scalability, sparsity and many more. The real-life examples of the recommendation system include Netflix, Amazon, Google, Instagram and many more.

CF is most efficient and widely used recommendation system, still it has some certain limitations, like sparsity, scalability, cold start ([23],[8],[9]). Normally, a user rates only certain limited items and it becomes difficult to find similar users due to sparsity in dataset ([24], [25],[26]). Also, when a new user or new item is added to the system for the first time, the sparsity grows in the dataset. It impacts the quality of recommendations and introduces a problem known as cold-start ([26]). Besides, over time rating grows in millions and computation becomes slow which introduces a serious scalability issue.

To solve the problems of scalability and sparsity in the collaborative filtering, an approach is given in [27] in which personalized recommendation methods joins the user cluster and item cluster. To improve the prediction quality of item-based collaborative filtering, some algorithms take the attributes of items into consideration while predicting the preference of a user ([28]).

There is an attempt to cope with Item cold start using a hybrid method which first clusters items using the rating matrix and then uses the clustering results to build a decision tree to combine novel items with existing ones [29].

Shahin Ahmed
Student # 500995601

Dataset

The dataset used was from MovieLens, and is publicly available at MovieLens Latest Datasets. In order to keep the recommender simple, I used the smallest dataset available (ml-latest-small.zip), which at the time of download contained 105339 ratings and 6138 tag applications across 10329 movies. These data were created by 668 users between April 03, 1996 and January09, 2016. This dataset was generated on January 11, 2016.(http://grouplens.org/datasets/movielens/latest)

The data are contained in four files: links.csv, movies.csv, ratings.csv and tags.csv. I am going to use the files movies.csv and ratings.csv to build a recommendation system.

glimpse(movies)

```
## Rows: 9,742
## Columns: 3
## $ movieId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,...
## $ title <fct> "Toy Story (1995)", "Jumanji (1995)", "Grumpier Old Me...
## $ genres <fct> Adventure|Animation|Children|Comedy|Fantasy, Adventure...
```

summary(movies)

movieId	title	genres
Min. : 1 1st Qu. : 3248 Median : 7300 Mean : 42200 3rd Qu. : 76232 Max. : 193609	Confessions of a Dangerous Mind (2002): 2 Emma (1996) : 2 Eros (2004) : 2 Saturn 3 (1980) : 2 War of the Worlds (2005) : 2 '71 (2014) : 1 (Other) : 9731	Drama : 1053 Comedy : 946 Comedy|Drama : 435 Comedy|Romance: 363 Drama|Romance : 349 Documentary : 339 (Other) :6257

From this initial exploration, we discover that movies have 9,742 observations and 3 attributes:

movieid : integer, Unique ID for the movie
title: factor, movie title (not unique)
genres: factor, genres associated with the movie

glimpse(ratings)

```
## Rows: 100,836
## Columns: 4
## $ userId <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ movieId <int> 1, 3, 6, 47, 50, 70, 101, 110, 151, 157, 163, 216, 2...
## $ rating <dbl> 4, 4, 4, 5, 5, 3, 5, 4, 5, 5, 5, 5, 3, 5, 4, 5, 3, 3...
## $ timestamp <int> 964982703, 964981247, 964982224, 964983815, 96498293...
```

Shahin Ahmed
Student # 500995601

summary(ratings)

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

From this initial exploration, we discover that ratings have 100,836 observations and 4 attributes:

userid: integer, Unique ID for the user
movieid : integer, Unique ID for the movie
rating: double, a rating between 0 and 5 for the movie
timestamp: discrete, Date and time the rating was given

glimpse(tags)

```
## Rows: 3,683
## Columns: 4
## $ userId <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 7, 18, 18, 18, 18, 18, 18...
## $ movieId <int> 60756, 60756, 60756, 89774, 89774, 89774, 106782, 10...
## $ tag <fct> funny, Highly quotable, will ferrell, Boxing story, ...
## $ timestamp <int> 1445714994, 1445714996, 1445714992, 1445715207, 1445...
```

userId	movieId	tag	timestamp
Min. : 2.0 1st Qu. : 424.0 Median : 474.0 Mean : 431.1 3rd Qu. : 477.0 Max. : 610.0	Min. : 1 1st Qu. : 1262 Median : 4454 Mean : 27252 3rd Qu. : 39263 Max. : 193565	In Netflix queue : 131 atmospheric : 36 superhero : 24 thought-provoking : 24 Disney : 23 funny : 23 (Other) : 3422	Min. : 1.137e+09 1st Qu. : 1.138e+09 Median : 1.270e+09 Mean : 1.320e+09 3rd Qu. : 1.498e+09 Max. : 1.537e+09

From this initial exploration, we discover that tags have 3,683 observations and 4 attributes:

userid: integer, Unique ID for the user
movieid : integer, Unique ID for the movie
tag: factor, Tag associated with the movie
timestamp: intger, Date and time the rating was given

Shahin Ahmed
Student # 500995601

glimpse(links)

```
## Rows: 9,742
## Columns: 3
## $ movieId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,...
## $ imdbId <int> 114709, 113497, 113228, 114885, 113041, 113277, 114319...
## $ tmdbId <int> 862, 8844, 15602, 31357, 11862, 949, 11860, 45325, 909...
```

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

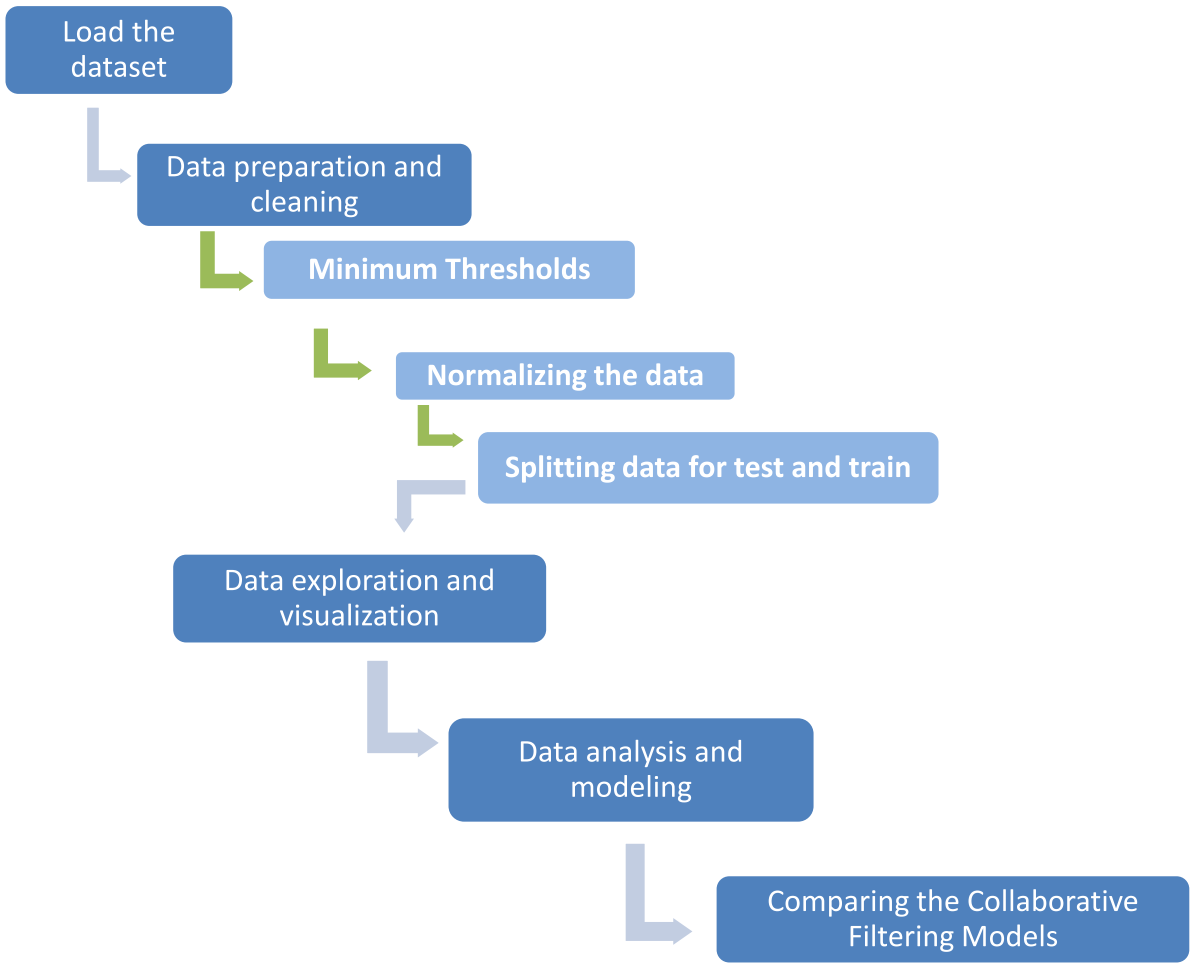
From this initial exploration, we discover that links have 9,742 observations and 3 attributes:

movieid : integer, Unique ID for the movie
imdbid: integer, IMDB ID for particular movie
tmdbid: integer, TMDB ID for particular movie

Shahin Ahmed
Student # 500995601

Approach

The workflow to build a Movie Recommendation System using the UBCF and IBCF models can be summarized graphically as follows:


Load the dataset
Data preparation and cleaning
Minimum Thresholds
Normalizing the data
Splitting data for test and train
Data exploration and visualization
Data analysis and modeling
Comparing the Collaborative Filtering Models

Step 1: Load the dataset

The dataset is split into four files- movies.csv, ratings.csv, links.csv and tags.csv. We will iteratively load the files into the workspace using download.file() function then unzip the required data file using read.csv() function and finally saving all data files (movies, rating, links and tags) as a RData file using save() function.

Step 2: Data Preparation and cleaning

In this section we will take the first look at the loaded data frames. We will also perform necessary cleaning and some transformations so that the data better suits our needs.

Shahin Ahmed
Student # 500995601

2.1 Minimum Thresholds:

For building a collaborative filtering model we can limit the input data based on minimum thresholds: for example, we may ignore users that have provided too few ratings, and also ignore those movies that have received too few ratings from users.

Here 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.

2.2 Normalizing the data:

We normalize the data so that the average rating given by each user is 0. This handles cases where a user consistently assigns higher or lower ratings to all movies compared to the average for all users. In other words, normalizing of data is done to remove the bias in each user's ratings.

2.3 Splitting data for test and train:

In this step I'm going to split the data into test and train sets, so that we could test the models on test data and later could compare different data models

Step 3: Data exploration and visualization

In this part we will try to explore the dataset and reveal some interesting facts about the movie business.

Step 4: Data analysis and modeling

In this part I am going to develop and analysis UBCF and IBCF models.

Step 5: Comparing the Collaborative Filtering Models

In this section I am going to evaluate the models created in previous part using different similarity parameters.

Shahin Ahmed
Student # 500995601

References:

[1] Netflix prize: Home. http://www.netflixprize.com/. (Accessed on03/07/2016).

[2] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. Springer, 2011.

[3] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, ``Collaborative filtering recommender systems," in The Adaptive Web. Berlin, Germany: Springer, 2007, pp. 291_324.

[4] P. Lops, M. de Gemmis, and G. Semeraro, ``Content-based recommender systems: State of the art and trends," in Recommender Systems Handbook. Boston, MA, USA: Springer, 2011, pp. 73_105.

[5] S. Trewin, ``Knowledge-based recommender systems," Encyclopedia Library Inf. Sci., vol. 69, no. 32, p. 180, 2000.

[6] R. Burke, ``Hybrid recommender systems: Survey and experiments," User Model. User-Adapted Interaction, vol. 12, no. 4, pp. 331_370, 2002.

[7] Linyuan Lua, Matus Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang,Tao Zhou, et al. "Recommender systems" Physics Reports 519.1 (2012): 1-49.

[8] Bobadilla, J., Ortega, F., Hernando, A. and Gutierrez, A. (2013) ‘Recommender systems survey’, *Knowledge-Based Systems*, July, Vol. 46, pp.109–132.

[9] Su, X. and Khoshgoftaar, T.M. (2009) ‘A survey of collaborative filtering techniques’, *Advances in Artificial Intelligence*, Vol. 3, pp.1–20.

[10] Pennock, D.M., Horvitz, E., Lawrence, S. and Giles, C.L. (2000) ‘Collaborative filtering by personality diagnosis: a hybrid memory and model-based approach’, *Proc. 16th Conf. Uncertainty in Artificial Intelligence*, pp.473–480.

[11] Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J.T. (2004) ‘Evaluating collaborative filtering recommender systems’, *ACM Transactions on Information Systems*, Vol. 22, No. 1, pp.5–53.

[12] Candillier, L., Meyer, F. and Boull, M. (2007) ‘Comparing state-of-the-art collaborative filtering systems’, *LNAI*, Vol. 4571, pp.548–562.

[13] Resnick, P., Iakovou, N., Sushak, M., Bergstrom, P. and Riedl, J. (1994) ‘GroupLens: an open architecture for collaborative filtering of netnews’, *Proc. 1994 Computer Supported Cooperative Work Conf.*, pp.175–186.

[14] Kendall, M. and Gibbons, J.D. (1990) *Rank Correlation Methods,* 5th ed., Charles Griffin, Edward Arnold, London.

[15] Zhao, Z.D. and Shang, M.S. (2010) ‘User-based collaborative-filtering recommendation algorithms on Hadoop’, in *International Workshop on Knowledge Discovery and Data Mining*, pp.478–481.

[16] Wang, J., De Vries, A.P. and Reinders, M.J.T. (2006) ‘Unifying user-based and item-based collaborative filtering approaches by similarity fusion’, *Proc. 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR’06)*, ACM, New York, NY, pp.501–508.

[17] Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. (2001) ‘Item-based collaborative filtering recommendation algorithms’, *Proc. 10th ACM International World Wide Web Conference*, pp.285–295.

[18] Treerattnapitak, K. and Juruskulchai, C. (2009) ‘Items based fuzzy c-mean clustering for collaborative filtering’, *Information Tech. J.*, Vol. 5, No. 10, pp.30–34.

[19] Pennock, D.M., Horvitz, E., Lawrence, S. and Giles, C.L. (2000) ‘Collaborative filtering by personality diagnosis: a hybrid memory and model-based approach’, *Proc. 16th Conf. Uncertainty in Artificial Intelligence*, pp.473–480.

[20] Chen, Y.H. and George, E.I. (1993) ‘A Bayesian model for collaborative filtering’, *Proc. 7th International Workshop on Artificial*.

[21] Kim, T.H., Park, S.I. and Yang, S.B. (2008) ‘Improving prediction quality in collaborative filtering based on clustering’, *Proc. IEEE/WIC/ACM International Conference*, pp.704–710.

[22] Tyagi, S. and Bharadwaj, K.K. (2012) ‘A hybrid recommender system using rule-based and case-based reasoning’, *International Journal of Information and Electronics Engineering*, Vol. 2, No. 4, pp.586–590.

[23] Adomavicius, G., Tuzhilin, A. and Zheng, R. (2011) ‘REQUEST: a query language for customizing recommendations’, *Information System Research*, Vol. 22, No. 1, pp.99–117.

[24] Anand, D. and Bharadwaj, K.K. (2011) ‘Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities’, *Expert Syst. Appl.*, Vol. 38, No. 5, pp.5101–5109.

[25] Bergholz, A. (2003) ‘Coping with sparsity in a recommender system’, *Lecture Notes Comput. Sci*., Vol. 2703, pp.86–99.

[26] Hyung, J.A. (2008) ‘A new similarity measure for collaborative filtering to alleviate the new user coldstarting problem’, *Information Sciences: an International Journal*, January, Vol. 178, No. 1, pp.37–51.

[27] Gong, S. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. Journal of Software, 5(7), 745-752.

[28] Puntheeranurak, S., & Chaiwitooanukool, T. (2011, July). An Item-based collaborative filtering method using Item-based hybrid similarity. In Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference on (pp. 469-472). IEEE.

[29] 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.