Analysis of Recommender Systems

Vibhav Agarwal (*Author*)

Abstract— Recommender Systems, as the word suggests, are software tools that are aimed at providing suggestions to a user based on the activities of the user itself. The recommendations made are necessary because it helps the users support their decision, such as what items to purchase, what movies to watch, or what news clippings to read. Recommender systems have become an important utility for both the companies and the user in the e-commerce market such that it helps the user sort the vast data available and for the company in the sense such that maximum of their data is sold/ used.
Development of Recommender Systems involves a multitude of skill sets such as Machine Learning, Data Science, Statistics, Adaptive User Interface and/or Consumer Behavioral Psychology.

Index Terms— Recommender, Systems, Items, Content, Algorithm,User, Filtering.

1 INTRODUCTION

Recommender Systems are tools and algorithms to provide the user with a recommendation that it is most likely to view.
These days, Recommender Systems or RS, have started to grow with the advent of machine learning and the application of artificial intelligence along the lines. These systems are used by almost every major company for delivering the best result. Google search uses RS to filter results based on the location, time, and the end user's browser history; Netflix uses it to give personal favorites without really getting a lot of ratings; AdSense uses it to target specific ads; Amazon uses it to display product based on the user's preferences- the essence of all being that the machine has *learnt about you and your preferences*. We will be discussing the various algorithms with examples to illustrate our point.
In the RS, there are two utilities involved: Items and User.
Items refer to any sort of content that has to be recommended and has to be sorted or aligned. Items are characterized on basis of their value, their complexity or their utility. The value of an item maybe true (1) if the item is preferred by the user or false (0) if the item is disliked by the user.
User refers to the people using the RS. The users may have diverse choices and likings. It is therefore dependent on a number of factors. For this motive and to provide maximum efficiency, the RS takes in a large number of factors before returning a result. The sorting and representation can be done in a number of ways and it primarily depends upon the technique used.
The working of Recommender Systems can be as simple as recommending related products in the case of Amazon and the complexity can go much higher in cases such as news articles where the textual representation, the structure of the article preferred, the region involved and the time dependencies of the items play a role, to name a few.
The only two dynamics involved are the user and the items. The objective is to create a list of items that would most likely be preferred by the user. The various steps involved are discussed in the paper and methods to evaluate a likely outcome are discussed.

2 THEORETICAL BACKGROUND

Recommender Systems (RS) are used to help users find new items , such as related news items, and related books on *"assuming" what the user likes* based on information about the user, or the recommended item[5]. These RS play a very important role for the companies in decision making, or maximizing their profits in terms of the content that they offer.
F. Ricci et al. (2011), defines the recommendation problem as-"estimating the response of a user for new items, based on historical information stored in the system, and suggesting to this user novel and original items for which the predicted response is high."
These systems also play an important role in decision-making, helping users to maximize profits or minimize risks [4].
Most companies prefer using a mix of the algorithms like Bayesian algorithm (being the most popular) [5], decision tree, matrix factorization based, and ensemble learning

2.1 FUNCTIONS OF THE RECOMMENDER SYSTEM

- Maximizing the sold items
- Increase diversity of their markets
- Increase the user satisfaction
- Build fidelity with the user
- Better understanding of the needs of the audience

2.2 WORKING

The Recommender Systems or RS after collecting the required data, work on a 3 fold synergistic model:

i. Content Based filtering
ii. Collaborative filtering

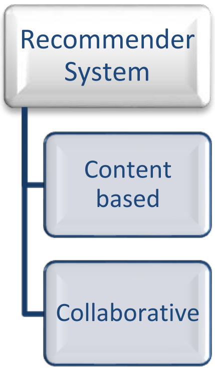

Fig 1. Types of Recommender Systems

2.3 CONTENT BASED FILTERING

A Content-based recommendation system recommends various items to users based on a profile that is unique to each user. The user's profile is centered around that user's ratings, it's choices, and a wide spectrum of factors including the number of times that user has clicked on different items or the amount of time that the user spends on the item or whether or not complete streaming/watching the entire item.

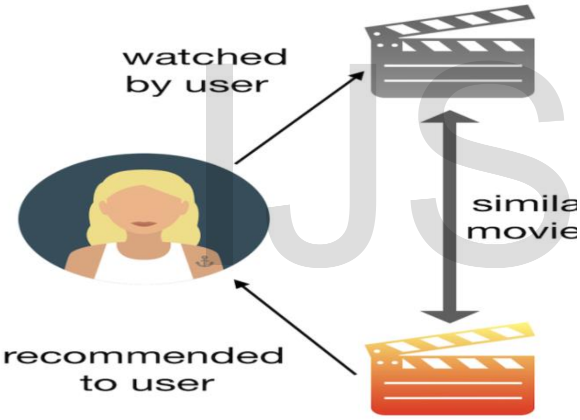

Fig 2. Content based filtering

Source:towardsdatascience.com

2.4 COLLABORATIVE BASED FILTERING

Rocco, 2019,[3]defines this type of filter is “based on users’ rates, and it will recommend the user movies that we haven’t watched yet, but users similar to us have, and *like*. To determine whether two users are similar or not, this filter considers the movies both of them watched and how they rated them. By looking at the items in common, this type of algorithm will basically predict the rate of a movie for a user who hasn’t watched it yet, based on the similar users’ rates.”

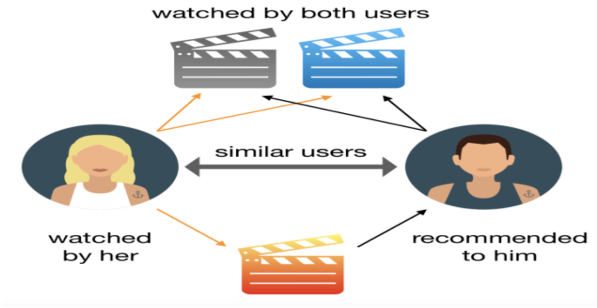

Fig 3. Collaborative filtering

Source:towardsdatascience.com

2.5 HYBRID BASED FILTERING

This third type of hybrid model is a synergistic alliance of the collaborative and the content based filtering model. In this model both the filtering processes are executed and the results which are common to both are given at the top with a maximum chance to be selected and the remaining results are returned normally. For example, in the search of people with similar names on a social networking site, it may first run the content based filter and then run the collaborative filter and then return the common result.

2.6 Machine Learning and its applications in RS

Two definitions of Machine Learning should be considered. Arthur Samuel described it as: "the field of study that gives computers the ability to learn without being explicitly programmed." This is a primitive definition, and a building block on which Tom Mitchell provides a more modern definition: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

“Take the task of playing chess.
E=The number of games of chess the computer plays
T=The task of playing chess
P= The probability that the computer would win the next game.”

Andrew Ng, Stanford University,[2] (2011)states that there are two main types of Machine Learning algorithms :

i. Supervised Learning: In this type of learning, the dataset is given and the relating factors are also given and the results of the task are also given. These type of problems are also known as ‘classification’ or ‘regression’ problems.

ii. Unsupervised learning: In this type of learning, the dataset is given but the relating factors or the result is not given. The classification of the dataset has to be done by the machine itself.

3 Gathering and Representing Data

3.1 Item profile/ User profile

For the purpose of the paper, the term 'data' refers to the item profile or details about a particular item, which would help to sort the data out.
The User profile refers to the preference or the rating of a user to a particular genre of a movie, or a particular movie which would be highly taken into account in the collaborative filtering.
For example: Consider a movie streaming service; the factors to be taken into account when making the profile of the item are:

i. Popularity (Users tend to watch movies which they have heard more about)
ii. Actor/Actress (Users tend to watch all the movies of their favorite actor irrespective of the plot of the movie)
iii. Release year(If a 90s fan is using Netflix, he will have release year as his preference)
iv. Genre(A particular user may have a taste for thriller or horror movies)
v. Director(A user may like a certain director like Martin Scorcese)
vi. Time of the day(A person might prefer a horror movie during daytime and a drama at night or vice-versa)-User dependent-different for every user

3.2 Gathering data/ preferences from the user

User Modeling is the process of identifying a user's choices and preference, and make recommendations on the basis of the actions of the user. These actions do not need to necessarily be in the form of ratings. There are two ways in which user modeling is done:

I. **Explicit**- Explicit user data collection happens when the user is aware that it is giving information to the computer which would eventually be used to develop recommendations.
The explicit data collection can further be categorized into two arenas:
   a. Boolean values: The traditional truth or false(0 or 1) values that are collected by the machine.
   Example: A like on Facebook.
   b. Fuzzy logic: The data collection based on the *degree of truth*. The values of this type are 1 to 5. A basic example is a rating of a product on Amazon. This would mean that the person like or dislikes the product and a rating of the same would be evaluated using the fuzzy logic. This fuzzy logic would then help recommend products that are of similar types.

II. **Implicit**- This type of data collection takes place when the user is not aware that it is directly providing information to the Machine.
Some companies monitor activities like key clicks and other activities like exiting a media content in the middle of the streaming, not watching the media content even after multiple advertisements, or taking a lot of time to continue the next episode or where they left from denoting the dislike factor and activities like binge watching, and increased number of hours on the show, to denote liking. The analytics team at Netflix concludes that this type of implicit data collection helps them decide the type of content that the users like and purchase content accordingly. [5]

For the purpose of the paper, we would not be taking in implicit data collection into account and will only use the explicit data gathered.
The purpose of the explicit data is to understand the preferences of the user. This data would be further used as the training set of the machine.
A key hurdle is to condense the data collected by the Boolean values and the data collected by the Fuzzy Logic. The key advantage of this process is that it gives us the functionality to work independently on the data.

To put it on a scale, any value ranging from 3, 4, or 5 can be taken a "like" or a 1 and any rating of 1, or 2, can be ignored and be taken as an item of non-interest or 0.
Some websites may even have a rating of say 1-10 or maybe the scales might extend to 100,but the end data is scaled by the standard formula on discreet values from 0-1.

3.3 Gathering training set about an item

The training set will not be available for the machine to apply its sort on. The training set in this case refers to the genre of a song, plot of a book, or section of news. For this case, the entire dataset or the document containing information about the data has to be sorted. In case of news articles, and research paper recommender systems like *Scienstein*, the entire document can be analyzed, whereas in case of media content, information about the content from a set reliable source can be taken into account. For this purpose, an algorithm called **TF-IDF** or **Term Frequency-Inverse Document Frequency** has to be applied. The algorithm is defined as "the number of times a word appears in a document, divided by the total number of words in that document(TF)" ; the second term is the "Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus

divided by the number of documents where the specific term appears.”
By using the TF-IDF algorithm, any information contained in a document can be sorted out by taking out the key words in the description. The reason behind taking the key words out is to have information about the genre, the plot, the words used maximum times in the review which is not easily available. Even if the plot is available, we have to shorten it down to a few words. These words are further arranged in frequency of their appearance by any standard sorting method like bubble sort or insert sort of the order log(n).

A Beautiful Mind	2001 Biography RonHowardRussellCroweJenniferConnelyJoshLucas
The Shawshank Redemption	1994 Drama Mystery CrimeFictionFrankDarabontTimRobbinsMorganFreemanBobGunton
Promised Land	2012 Drama GusVanSantTimGuineeMattDamonJohnKrasinski
A Few Good Men	1992 LegalDrama Courtroom MysteryRobRienerTomCruiseJackNicholsonAaronSorkinDemiMoore

Table 1. Example

3.4 Example

The idea is to first filter the content on a given factors. Although the system has been constructed for movies, for books or music, similar variables can be taken into account. For books, the author, the plot, the year of release, and the genre can be taken into account.
For music, the music composer, the genre, and the instrument used can be taken into account. For the purpose of this model, take in an example of a movie streaming service like Netflix and take the example of 5 movies.
The model will work in 3 steps:

i. Gathering the data
ii. Data Cleaning
In the ***ntlk*** package of the Python language, there is a function which allows us to separate the key words from the text. The ***Rake*** function is used to extract the main words of the plot and separate out the common keywords.

Everything needs to be lower case except the first word. The space between the names have to be removed otherwise Tom Brady and Tom Cruise will catch a similarity. Then the entire
data with the director, actor, plot(condensed), year of release and genre is grouped together in a group called **bag of words**.
This data set is ready for the user and can now be used in the algorithm.

4. Content based Recommender Systems

4.1 Principle

Now that the data to be collected is arranged, the data has to be processed and the essence of a content based RS is to understand the similarity between two items. Once the similarity has been established, the items with the highest similarity value will be recommended. This type of RS is independent of the psychology of the user and has no application of AI in it. As opposed to the Collaborative filter, where the actions of a similar user predict the choice, in this type of a system, only the similarities between the item have to be taken, which, again, can be on a diverse level.

4.2 Developing similarity using cosine function

In application, the cosine similarity function is used to determine how similar two vectors are. In mathematical equations, it is used to calculate the angle between two vectors in a 2 dimensional space.
Here, we will be plotting the properties of the item in a multi dimensional vector. After the vector has been formed, it would be treated as treated mathematically by putting it in the equation below.[1]

$$Cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\,\|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\sqrt{\sum_1^n b_i^2}}$$

where, $\vec{a}\cdot\vec{b} = \sum_1^n a_i b_i = a_1b_1 + a_2b_2 + \cdots + a_nb_n$ is the dot product of the two vectors.

Fig 4. Cosine similarity

With the help of this equation, for two given vectors, a value between 0 and 1 will be given where 0 will signify no similarity, and 1 will stand for complete similarity, both being the ideal cases. Here the cosine function will give the similarity between two items based on the properties listed in section 3.1.

4.3 Result

On running the algorithm discussed above, the cosine between various vectors will be formulated and on plotting the value of any two given vectors on a two-dimensional matrix, a similar output would be obtained.

$$\begin{array}{c|ccccc} & Movie_1 & Movie_2 & Movie_3 & \cdots & Movie_n \\ \hline Movie_1 & 1 & 0.158 & 0.138 & \ldots & 0.056 \\ Movie_2 & 0.158 & 1 & 0.367 & \ldots & 0.056 \\ Movie_3 & 0.138 & 0.367 & 1 & \ldots & 0.049 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ Movie_n & 0.056 & 0.056 & 0.049 & \ldots & 1 \end{array}$$

Fig 5. Plotting of similarity on a matrix

In any such matrix, diagonal values would always be one since any item would be absolutely similar to itself. On getting the similarity between two items, the item with the maximum similarity would be recommended.

5. Collaborative Filtering

5.1 Introduction

Collaborative filtering is based on recommendations taken by another user who has given similar ratings. To illustrate collaborative filtering, let two users be u and v. The key principle behind this concept is that if users *u* and *v* have given similar ratings to items *i , j, k, l and m* then they are most likely to give similar ratings to another item *n.* The model is based on the two users having a 'similar taste', to put it in simpler words.

5.2 Principle

The principle behind the collaborative filtering is the neighbor-hood approach. What this means is that for a given user *u,* the machine would try to find a user *v,* which would be called a neighbor of user u if the user v has given similar ratings.
Furthermore, this methodology may not recommend the best recommendation for the user, but the key aim of this approach is to let the user discover a complete new genre of items which the user may not have generally discovered.

For example, if a particular user likes movies of the genre romance and comedy, any model based RS would be able to recommend romantic comedies. However, according a trend, users who have liked romantic comedy tend o like horror comedy as well. A model based recommender system would not have been able to find this genre out. But a RS based on collaborative approach, can understand the psychology of all the users and the type of audience and customers present.

5.3 The Neighbor Approach

The first step in starting with a dynamic collaborative based recommender system is to understand and chart users that are similar to the target user.

Let us consider a user I who has to be given a recommendation. The first step is charting out users i1,i2,i3 which are 'similar' to the user i. This approach is called the neighbor approach and the similar users are called neighbors.

The second step is to predict the rating of an item I for a user u which user *U* hasn't rated but one of the users *U1,U2* or *U3* has.

In order to find similarity between the users, we need to compare the users based on the ratings they have previously given to items.

	Twilight	Star Wars	Good Hunting	Shawshank	Godfather
Mary	2	3		2	3
John		3	1		5
Hannah	4			5	2
Amit	2	5	3	1	

Table 2. Example training set

In order to predict the rating Amit would give for Godfather, we need to understand and find out the similarity between the users first.

5.4 Finding Similairty

Let the number of items *i* rated by the users *U1* and *U2* be *CI***(denoting common items rated)** and the rating given be *r(U1)* by User *U1* for the common items and rating *r(U2)* given by User *U2* for the same common items.
Let the similarity rating be given by *w(U).*
Then the Mean Squared Distance would be given by:

$$MSD(U1, U2) = \frac{|CI|}{\sum_{CI}(rU1 - rU2)^2}$$

Fig 6. Calculation mean squared distance

Example- To take out the similarity Amit has with each user, the above formula would be put to take out the ratings of similarity for the other users.

	Amit
Mary	0.6
John	0.25
Hannah	0.1
Amit	1

Table 3. Similarity of Amit with other users

As it is obvious, the similarity for Amit with himself would be 1, so it can be ignored in each case. As for the users available, Mary looks like she has a 60% similar taste as Amit.

5.5 Predicting Rating for a User

Now that the similarity has been taken out and for a User *U1*, a set of neighbors has been established, the set of neighbors for any user UX is represented by U such that *r(U)* is the rating given by each neighbor and *N(u)* is the number of neighbors found by the previous algorithm.

$$r(UX) = \frac{1}{N(u)} \sum r(U)$$

Fig 7. Calculating rating for a user

However, there is one factor that the above formula misses out. In the above given example the rating of similarity for Amit and Mary are the highest and to take into account the ratings of John and Hannah with the same weightage as Mary would be foolish and imprudent.
A logical solution to this problem would be that while calculating the predicted ratings, the existing similarity ratings calculated previously be taken into account.

Let, for each user corresponding U to user UX the similarity rating be *wU* and once again, rating be *r(U)*. Now, the resultant formula would appear like this:

$$r(UX) = \frac{\sum_{N(u)} wUrU}{\sum_{N(u)} wU}$$

Fig 8.Accurate predicted rating

Example- For this example, we will use the above formula to predict a rating that Amit would give for Godfather.
0.6*3 + 0.25*5 +0.1*2 / (0.6+0.25+0.1) = 3.42
This rating is most similar to Mary's and thus can be predicted as a logical prediction for the user.

6 Limitations

- Both the models do not take in implicit responses.
- The hybrid model of the two has not been discussed in the paper.

7 Conclusion

As you might understand after reading the paper, that Recommender Systems are a great tool to help model one's user database and enhance the user experience. What is also important is that RS also help the companies understand the behavioral psychology about their users- it helps them to modify their platforms according to the type of content their users enjoy. This helps companies cut down a large part of the expenditure they might make without having a dataset regarding the users.
With the future advent of technology and the upcoming role of Artificial Intelligence, we are yet to rediscover what technology might hold for us.

8 Reference

[1] Ricci, Francesco &Rokach, Lior&Shapira, Bracha. (2010). Recommender Systems Handbook. 10.1007/978-0-387-85820-3_1.

[2] Andre Ng, Stanford University(2011). Machine Learning by Stanford University

[3] Rocca,2019. Overview of some major recommendation systems.

[4] Bouneffouf, D., Bouzeghoub, A., &Ganarski, A. L. (2013, January). Risk-aware recommender systems.In Neural Information Processing (pp. 57-65).Springer Berlin Heidelberg.

[5] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, 17(6), 734-749.