An Overview of Recommender Systems

Himanshu Rawat

SP Jain School of Global Management

Abstract
This study examines the scope of Recommender Systems being used in solving different real world problems. The study explores the areas of application, the industries using these tools & the impact of the system in user experience. The study further examines what these systems are composed of, how to develop these systems & how to evaluate these systems & the reproducibility of the systems. By the end of the study it is expected that the reader will have all the essential information regarding the recommender systems.

Introduction

When a user interacts with a very large catalogue of items, it can be movies on Netflix or products on Amazon what really matters is there are millions of items in a really large catalogue and there are two ways in which user can interact with a large catalogue of items. First is search - User knows what they are looking for and they go search for the item they are looking for. When the user doesn't know what they are looking for that is where the recommendations comes in. The system recommends the user certain items they think user will be interested in based on what they know about the user. The reason why there was so much of development in past 10-20 years in the field of recommender system is because we moved from ***an era of scarcity to an era of abundance*.** With increasing availability of large volumes of data, it has become difficult to go through all the available options for finding what one is looking for. Even with developments in searching techniques, the user may still come across several options that do not fit their preferences. A recommender system has become one of the most interesting research platforms for investigating information overload. The recommender system's turns users' current preference data into predictions of future likes and interests. Recommender systems provide users with a list of personalized recommendations on catalogue of items. A recommender system creates a user profile by extracting the user's relevant characteristics and based on these characteristics recommends the set of items that may be of interest to the user.

If we went shopping 20 years ago at a local retailer's place, we would find certain number of products on the shelf of the retailer. Even for a large retailer, shelf space is a scarce commodity, it limits the number of products a retailer can carry. Shelf space is expensive because it involves real estate cost, and therefore retailer can carry on a limited number of products. A similar situation applies in case of TVnetworks, a TV networks can carry only so many number of shows because there are only so many number of hours in a day and there are only so many movie theatres so they can only screen a certain number of movies. With the development of the internet things changed for good. The web enables zero cost dissemination of information about products, this means we can have many more products than ever before, there is no shelf space limitation for the number of products, that's why number of products on amazon are much more available than any other physical retailer, number of movies available on netflix are much more than available at any blockbuster, this near zero cost dissemination gives rise to long tail phenomenon, on x axis we take items on the catalogue and rank them on the basis of popularity, so the most popular items are on the left and as we move on towards right items become less and less popular, by popular we mean number of times a item is bought or number of times a movie is seen. On y axis we have actual popularity, we get a curve with very steep fall initially i.e we have a few very popular items and as we move towards right as the item rank becomes greater and the popularity falls less and less steeply, however there is a cut off. Items less popular than this cut-off point maybe purchased once a week or maybe once a month, if you are a physical retailer we don't stock this item and therefore retailer doesn't stock any item which are less popular, this is also applicable to books, music, videos, news articles. Items which are less popular can be find only online but there are so many of them, how do we introduce a user to all such items that they may not otherwise find, we need a better way for the user to find all these items, the user doesn't even know where to start looking, this is where recommender system comes in. So recommendation system works in the case of many items like books, movies, interestingly even for people, for example Facebook , Twitter, Linkdin there are so many people that we don't know who to follow or who to friend and so Facebook or linkdin or twitter makes recommendation on the people that you could follow or friend.

Several years ago a book was published called "Touching the wild" it was a book on mountaineering, the book came out it did not make much popular. Few people bought the book, it got decent reviews but it never became the best seller few years after a new book was published on mountaineering called "Into the thin air" it picked a lot of attraction, and lots of people started buying "Into thin air", Amazon noticed that a few of the people who bought "Into the thin air" also bought " Touching the wild". This made "Touching the wild" even a bigger best seller compared to "Into the thin air". This explains the power of a good recommender system.

A good recommendation system exposes the people to hidden gems. Oldest recommender system and the simplest recommender system is editorial and hand curated. We can find such system on the home page of a website, we can see the topics hand-picked by the editorial staff on the home page of the website. One of the drawback of such recommender system is that these are done entirely by the staff of the website without taking any inputs from the users. Another simple recommender system is "Simple Aggregates" on many websites we can find list of most popular products or most viewed items. These are simple aggregates that take into account the user preferences. It does not take into account the views of all users but only considers the activities of the few users. There are also recommender systems which are tailored to individual users. For example movie recommendation on the basis of the movies that you have watched recently.

<u>Recommendation System Techniques</u>

Problems for a Recommender System

1) It needs to gather ratings already present in the system

2) Extrapolating data i.e. use current trends/situations to handle unknown variables

3) Evaluating extrapolation methods to measure the effectiveness of system

Mainly there are 3 types of recommender system:

- Collaborative Filtering Techniques
- Content Based Filtering Techniques
- Hybrid Filtering Techniques

1) <u>Collaborative Filtering Technique:</u>

This technique identifies data of users/items across the world who are using same platform & makes recommendations based on the similar tastes/preferences. Like on e-commerce sites the users post their ratings. These ratings can be used in two ways:

i) **Explicit:** A user continuously rates items and services genuinely. This gives an idea about the user behavior & allows to picturize the user profile. However, it becmoes difficult to get this picture when the user hasn't rated all the items he/she has purchased.

2) **Implicit:** The system uses ratings given by other users to form a rating pattern of first user for the same item & recommend products/services accordingly.

a) User-based Collaborative filtering

Ratings are given by different users for the products/services they have purchased based on their user experience. A mean value based on the weightage of the ratings is calculated & the recommendations are made to the user. The similarity values are calculated for the user.

- Neighbourhood Formation
  Most similar users become part of a neighbourhood. This is done by the ratings they have given to products they have purchased & rated.

- Rating Prediction
  All the items that first user(u1) has bought & rated well, are recommended to the Neighbour u2, which he is yet to purchase. The recommendations to u2 are made in descending order, giving highest predictions first.

- Jaccard Similarity
  Similarity between two set of users is calculated. The more similar two users are in their preference, more of their preferences can be shared among themselves. It is expressed by: Sim (A,B)= |Ra Intersection Rb|/ |Ra Union Rb| (Where Sim(A,B)=Similarity between User A & B; Ra & Rb are the ratings given by User A & B)

  This technique is used less as this technique Ignores weightage attached with rating.

- Cosine Similarity
  This technique tells us about how much two users are more closer in preference as compared to others. E.g. Cosine similarity of Ua is compared with respect to Ub & Uc.

  Its drawback is that it is unable to give appropriate weightage to rating relationship between Ua-Ub & Ua-Uc. This happens because it considers non rated items as ‘zero rated’ which over calculates the result.

- Central Cosine
  Also known as Pearson’s cosine normalizes the ratings by placing users & items into different rows & coloumns Row mean is subtracted from ratings value to normalize the rating & Cosine(Theta) for preferences of two users are compared. Higher the value of Cosine (Theta) indicates that the two users are closely matched.

b) Item based Collaborative filtering

Instead of users the items are analyzed & recommended to buyers. Two similar items are compared based on the ratings given by the users. These items are identified based on the users ratings two similar items & giving them similar ratings. Then for each item in the category a list of top rated items is defined, again in descending order.

Above mentioned techniques are used for Item based collaborative filtering only. However, item based filtering gives **much** better results compared to user based methods. This is primarily because the items have well defined attributes which are always fixed. Whereas, users are human beings who can have completely random attributes. E.g. A user who likes classic rock might like heavy metal as well, these are completely opposite genres of music & at times are still liked by the same person.

c) Topic Diversification Algorithms:

Many of the systems are biased towards something a user has already tried or very similar to his purchase style (consumer behavior). This impacts a user's satisfaction as at the end of the day he keeps receiving the similar recommendations & doesn't receive anything new. A different item list is generated which has recommendations ranked on the basis of similar ratings to the product and of similar class. These recommendations can further be evaluated into the results they are producing whether they are closer to a user's preference or is it something different.

Pros:

i) It can work for any kind of item. No feature selection is needed.

Cons:

i) Cold start is difficult. Needs a lot of users in the system

ii) Sparsity of user/ratings matrix

iii) Popularity bias like Harry Potter effect which may or may not be a bad thing

iv) Difficulty in recommending unrated items.

2) <u>Content Based Filtering Techniques:</u>

Idea behind content based filtering techniques is to recommend items to customer which are similar to previous items highly rated by him or words searched by user in the search engines.

E.g. Movies-> Same actors, Directos, Genres etc.

People-> Recommend people with many common friends.

a) Term Frequency-Inverse Doc Frequency (TF-IDF)

This function is calculated in two parts Term Frequency & Inverse Doc Frequency.

Term Frequency(ij)= Frequency of i with respect to j (Fij) / (Max-Appearence of a word * Max word appearing maximum in how many documents)

Inverse Doc Frequency: Many a times in articles words like The or A would appear a lot more than useful words. That doesn't mean these words are more important as being calculated by Term Frequency. To balance it Inverse Doc Frequency is calculated. It is given by expression:

IDFi= Log N/ni; Where ni= Total no. of docs that mention term 'i' & N= Total no. of docs

TF-IDF score for a word is obtained by multiplying the above two equations together.

b) User Profile

A user profile is generated based on his response to the ratings, where he has explicitly reviewed certains features. E.g. A rates 2 movies 3, 5 out of 5 star. We can subtract the average rating from individual rating & divide it by total no. of ratings. The expression is given by:

Dot product of vectors of normalized user rating & item profile / magnitude of user rating & item ratings

Plotting this value in cosine(Theta) against the average movie ratings received by other movies we can find out the similarity between user's rating and similarly rated item in same item class. The higher the value of Cos(Theta) higher the chances the user will give higher rating to the item recommended to him.

Pros:

i) There is no need for data on other users

ii) It is able to give good recommendations to users with unique tastes

iii) It is able to recommend new & even unpopular items

iv) A logical explanation to user can be given. There on his customized needs can be explored.

Cons:

i) Finding the appropriate features can be difficult.

ii) Over speacialization of recommended items. Might end up giving similar recommendations again n again.

Iii) Difficult to have a cold start. How to build a profile without ratings for a new user.

3) <u>Hybrid Filtering Techniques:</u>

Even though the above techniques have been very successfull there has been identified scope for improvement. There are certain challanges with these two techniques like issue of scalability, limitation of content analysis, sparsity of data among others. By combining these two techniques one can overcome the shortcomings of other technique & prepare a better model.

i) To reach new users synthetic user profiles can be made depending on not just the ratings but the items the user has purchased.

ii) Combine 2 recommender systems using Linear Method

E.g. Global Baseline approach

Lets say Movies of Genre 'G' have an average rating of 3.5 stars in the system

A Movie 'M' of same genre is rated .5 star higher

User 'U' is a good rater & rates movies .5 stars higher as per his profile.

Therefore, baseline = 3.5+.5+.5= 4.5stars

So we can assume that user U might rate movie M as 4.5 & thus build a profile of recommendations.

Evaluation of recommender system:

In the past decade, there has been a vast amount of research in the field of recommender systems which were focussed on developing new algorithms for recommendations. An application designer designing a recommendation system has a large variety of algorithms but has to make a decision about the most appropriate algorithm to use. Such decisions are based on experiments which compares the performance of a number of recommender systems. Such experiments are typically performed by using some evaluation metrics that provides a ranking to the candidate algorithms. In this paper, we discuss the process of evaluating a recommendation system using three different types of experiment:

i) Offline experiments
ii) User studies
iii) Online experiments

i) Offline experiments: In this experiments, data set of users choosing or rating items is pre collected. The data set is then used to simulate the behaviour of users who interacts with the recommendation system. The assumption taken in such experiments is that the behaviour of the user when the data was collected will be similar to the user behaviour when recommender system is deployed.

ii) User studies: It is conducted by recruiting a set of users, who are given to perform several tasks which requires interaction with the recommendation system. While the users perform the task, their behaviour are observed, recorded and collected for both quantitative and qualitative measurements.

iii) Online evaluation: In this experiments, the recommender system is used by real time users to perform real tasks. Such experiments redirect a sample of users to interact with different alternative recommendation systems, and record the user's interaction with the different systems. It is more trustworthy to compare the recommendation systems online thereby obtaining a rank of alternatives. The real effect of recommendation system depends on a number of factors which includes the user's intent, user's context and the interface through which interaction takes place with the recommendation system.

Prediction Accuracy:

At the base of the majority of the recommendation system lies a prediction engine which predicts user opinions over items (for example – rating of movies) or the probability of usage (for example – purchase). A basic assumption in recommender system is that a user will prefer a system that provides more accurate predictions.

a) Measuring Ratings Prediction Accuracy

In some applications, such as in the new releases page of Netflix DVD service where users rate an item from 1 star to 5 star, one wishes to measure the accuracy of the systems predicted ratings.

i) Root mean squared error (RMSE) is the popular metric used in the evaluating accuracy of predicted ratings. The system generates predicted ratings r_{ui} for a test set N of user item pairs (u,i) for which true ratings p_{ui} are known through an offline experiment or a user study experiment. The RMSE between predicted and actual rating is given by:

RMSE = $\sqrt{1/N\sum_{(u,i)\in N} (r_{ui} - p_{ui})^2}$

Mean absolute error (MAE) is a popular alternative

MAE = $1/N \sum_{(u,i)\in N} \| r_{ui} - p_{ui} \|$

b) Measuring usage prediction

In many applications, the recommender system doesn't predict the user's preference of items such as movie ratings but recommend to user's item that they may use. For example – when we watch a movie in Netflix, it also suggests or recommends a set of movies on the basis of movie that has been watched. In this case we are interested whether the system properly predicts that the user will use the item. Through an offline evaluation of usage prediction, a data set consisting of items each user has used is collected. A test user, whose selection are hidden and the recommender system is asked to predict a set of items that the user will use. Four possible outcomes for the recommended and hidden items can be found.

	Recommended	**Not recommended**
Used	**True positive (tp)**	**False Negative (fn)**
Not used	**False Positive (fp)**	**True Negative (tn)**

Precision= tp / (tp + fp) = correctly recommended items / total recommended items.

Recall = tp / (tp + fn) = correctly recommended items / total useful recommended items.

F measure helps to simplify precision and recall into a single matrix

F measure = 2PR/P+R

A longer recommendation list will typically improve recall while it is likely to reduce precision. In applications where the number of recommendations present4d to the user is fixed precision is the better matrix to use. Whereas in application where the number of recommendations presented ot the user is not fixed it is preferable to evaluate algorithms over a range of recommendation list lengths rather than using a fixed length. In such cases, curves comparing precision to recall (Precision recall curves) or true positive rate to false positive rate {(ROC) receiver operating characteristic curves} are preferable used. Both curves measure the proportion of the preferred items that are actually recommended, precision recall curves emphasizes the proportion of recommended items that are preferred while ROC curves emphasizes the proportion of recommended items that are not preferred.

Conclusion:

Getting Information off the internet is like taking a drink from a fire hydrant" - Mitchell Kapor

Recommender system helps to reduce the problem of information overload, a common phenomenon with information retrieval systems & it enables user access to products & services which are not readily available to users on the system. This paper discussed different recommendation techniques & highlighted their pros & cons with diverse kind of hybridization strategies used to improve their performance. Various learning algorithms used in building recommendation models & evaluation matrix for measuring the quality & performance of the recommendation systems were discussed. This knowledge will help researchers & serve as a road map to improve the state of the art recommendation techniques.

Furthermore, there are some new & upcoming trends in Recommendation systems which are as follows:

a) Matrix decomposition for recommendations: This is a new technique to where vectors are built by the known scores & use them to predict unknown grades.

b) Clustering: This techniques allows us to move slowly from supervised machine learning to unsupervised machine learning.

c) Deep Learning Application: Deep Neural Networks are the next thing coming in Recommendation systems

References:

a) J.A. Konstan, J. Riedl, Recommender systems: From algorithms to user experience

b) Leskovec, Rajaraman & Ullman, Mining of massive datasets- Recommender Systems, Stanford University

c) Guy Shani & Asela Gunawardana, Evaluating recommendation systems

d) F.O. Isinkaye & B.A. Ojokah, Recommendation Systems: Principles, Methods & Evaluations

e) Neelesh Bansal, Tellmate Helper Private Ltd.

f) Paul Covington, Jay Adams, Emre Sargin- Deep Neural Networks for YouTube Recommendations