International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2020): 7.803

Multi - Category & Multi - Criteria Recommendation System using Collaborative Based Filtering

Shubham Gaur, Rishabh Naulakha, Khushal Hanswal, Meenakshi Sharma, Abhishek Mohanty

[1, 2]Department of Information Technology, SRM Institute of Science & Technology, Chennai - 603203, India

Abstract: ***Recommendations Systems have become one of the most popular application of data science today. It predicts or offers products to customers based on their past browsing history or purchases. Although a lot of effort, research and time has been spent on recommendation engines, we are yet to truly unlock their potential. At the core, a recommender system employs a machine learning algorithm whose job is to predict user's ratings for a particular entity. Through this project, we are employing a multi category recommendation system which will give the user recommendations across different categories based on the user data of multiple categories consisting of different attributes. The concept of K - Nearest Neighbor Algorithm is implemented to derive the similarity of unknown entities or users based on past ratings of a particular entity. The implementation is carried out using JavaScript in Node, thereby extending the capabilities of Collaborative based filtering Algorithm to multiple categories.***

Keywords: Multi Criteria Recommender System (MCRS), K - Nearest Neighbor Algorithm (KNN), Similarity Score, Collaborative - based filtering algorithm

1. Introduction

The increasing amount of data and its complexity drives today's world to make use of it in a very respectable and efficient way possible. The volume of data is increasing at very fast pace and now you can measure the data in three manners, by 3 Vs i.e., Volume, Velocity and Variety. The data can be used for various other associations, it may be visualizing, measuring, or analyzing. This modern era is dedicated to time, efficiency and data, utilizing the same to your best use.

Big data era is going on while the World Wide Web also keeps growing exponentially, the size and complexity of many websites such as Google, YouTube, Netflix, Amazon and others grow along with it, making it increasingly difficult and time - consuming to find the item i. e., movie, music, restaurant, book or any product you might be looking for.

Traditionally, majority of recommender systems have a vision for providing recommendations by modelling a user's utility or preference for an item as a single preference rating.

The idea of exploiting the information from user's rating can be useful to solve one of the problems recommender systems suffer from, predicting user's preferences about a particular item using a single rating. This is surely a clear limitation since the user who is willing to make a choice might consider more than one aspect of the item. For example, consider a movie recommendation system, some users like a movie based on plot, direction, and conflict while other user may show interest in the same movie but for its acting, characters or any other attribute of that movie. However, reaching to an extent wherever possible, capturing not only the user's preferences for a given movie but also her preferences for a specific movie aspect such as story, acting and visual effects, can provide immense opportunities for further improvements in the quality of recommendation

Multi - Criteria Recommendation system (MCRS) which utilizes multi criteria ratings to evaluate different attributes of the item can improve the accuracy of recommendations.

As a result, various number of recommendation techniques that attempt to take advantage of such multi - criteria user rating have been developed in recent years. Conducting a review of current algorithms that use multi - criteria ratings for generating recommendations and calculating predictions are provided.

This project involves building a "Multi Category Multi Criteria Recommendation System using Collaborative Filtering (CF) " i.e., the category of recommender systems that use *multi - criteria preference rating for multiple categories.* Recommendation for X category will be based on the user which has similar ratings of that category selected by the user.

Please note that Ratings for individual category is derived from the different aspects or attributes of that item. For example, the data set already contains the different categories such as books and movies, and books consisting of username, plot, genre of book, novel ratings and movies consisting of username, genre, plot and IMDB rating of that movie as multiple criteria.

Then the user enters the input data based upon his/her interests and gives the rating. The user is asked in which category he wants recommendation (books/movies). Post this the recommendation engine recommends the movies/ books with their attributes.

2. Litreature Review

In the recommendation space lot of work has been done in the past. To drill down into multi criteria recommendation system, some of the following research, that have been conducted in the space of multi - criteria recommendation system are:

1) Recommender system using Dynamic Normalized Tree of concepts model for user modelling
2) Recommender System: A random walk approach
3) Social Media Recommender Systems: Review and Open Research Issues
4) A fuzzy approach for multi - criteria decision making in web recommendation for e - commerce

Some of the technologies that are used in the above research work are:

1) Fuzzy logic
2) Dynamic Normalized Tree Concept Model
3) Collaborative based Filtering Algorithm
4) Multi - Criteria Pseudo Rating and User Profiling
5) Multi - decision criteria for multiple domains
6) Arithmetic Research Driven Recommender System Framework
7) Content Based Filtering Algorithm
8) Random Walk Approach

There has been research work done in the space of multi category or cross domain recommendation system. Cross domain recommender systems (CDRS) assist the recommendations in a target domain based on knowledge learned from source domain.

The Figure 1 shows the notions of domain according to attributes and types of recommended items.
(a) **Attribute level:** same type of items (movies) with different values of certain attribute (genre).
(b) **Type level**: similar types of items (movies and TV shows), sharing some of their attributes.
(c) **Item level**: different types of items (books and movies).
(d) **System level:** same type of items (movies) on different systems (theatre and TV).

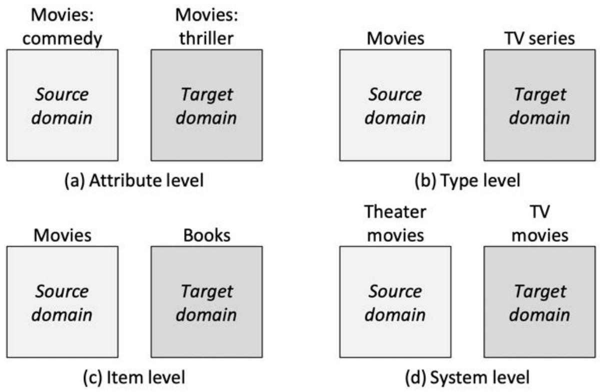

Figure 1: Cross Domain Recommender System

Post reading the research paper in the multi category as well in the multi criteria recommendation space, we have drawn various inferences from it.

2.1 Comprehensive Recommender System

Based on our observation and research there has been recommender systems that gives recommendations based on:

1) **Single Criteria (Attributes/Entity):** Consider movie recommendation system in which the recommendation is given based on IMDB rating as the only criteria.
2) **Multi Criteria**: In the above example itself, consider more aspects of deciding the recommending along with IMDB rating such as director, genre, tickets sold.
3) **Multi Category Single Criteria**: Consider two domain books and movies each consisting of User Rating as an attribute. User will be asked which category he/she wants recommendation for and accordingly he will be served with it.
4) **Multi Category Multi Criteria:** Consider two domain books and movies each consisting of User Rating, genre, tickets/books sold as attributes. Based on these multi aspects comprehensive rating will be calculated. User will be asked which category he/she wants recommendation for and will be served with its movies/books with all attributes merged in it.

Currently there is no recommender system that does the Multi Category Multi Criteria Recommendation. In the light of the above facts and inferences, our motive behind this project is to build this model using Collaborative based filtering algorithm. It filters the rating according to user's interest and preferences.

3. Proposed System

The recommender system is divided into two parts – frontend and the backend.

3.1 Frontend

1) The recommender system gets the input from the user on the website. The user is provided with the form in which he/she can provide the ratings of movies and books according to multiple aspects provided to him.
2) The user is asked which category he wants recommendation for.
3) Finally, at the end the user is provided with the recommendations in the frontend itself.

3.2 Backend

1) The input user ratings are used to calculate the Similarity Score against the already collated dataset according to user profiles.
2) The algorithm uses K - Nearest Neighbor Algorithm.
3) Send recommendation to the frontend.

Figure 2 illustrates how the flow of the Multi - Category Multi Criteria Recommendation System (2MC - RS) works.

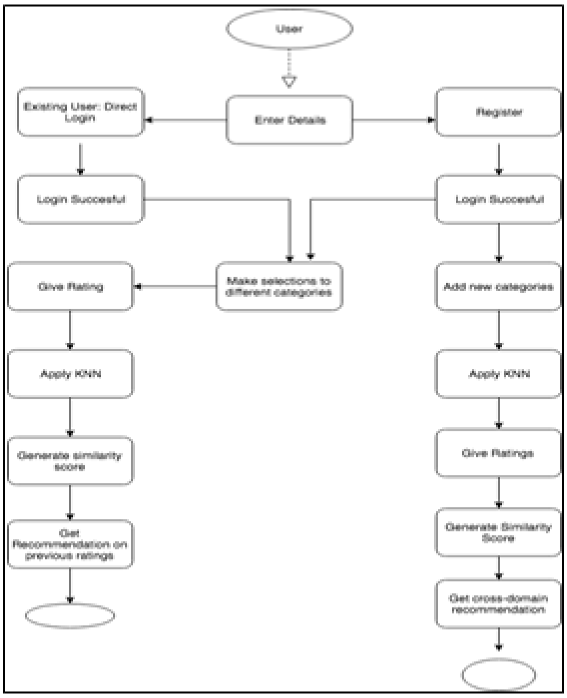

Figure 2: Activity Diagram – flow of the recommender system

4. Implementation Methodology

4.1 Dataset

An example of how the dataset will appear is given in the figure 3. Dataset contains the ratings of movies of users taken into our data store, which will be used to compare against the new input user data.

```
var dataset={
    'Lisa Rose': {
    'Lady in the Water': 2.5,
    'Snakes on a Plane': 3.5,
    'Just My Luck': 3.0,
    'Superman Returns': 3.5,
    'You, Me and Dupree': 2.5,
    'The Night Listener': 3.0},


'Gene Seymour': {'Lady in the Water': 3.0,
    'Snakes on a Plane': 3.5,
    'Just My Luck': 1.5,
    'Superman Returns': 5.0,
    'The Night Listener': 3.0,
    'You, Me and Dupree': 3.5},
```

Figure 3: JSON data Snippet

4.2 Multi Criteria Rating

Consider looking at the multi - criteria aspect of the category movies. In this scenario, we have assigned the user profile or names into numbers for easy calculation. The multiple aspects taken into consideration are title, genres, rating, timestamp, movieId.

Table 1: Example of Multi - Criteria Rating

	userId	movieId	rating	timestamp	title	genres
0	1	16	4.0	1217897793	Casino (1995)	Crime|Drama
1	9	16	4.0	842686699	Casino (1995)	Crime|Drama
2	12	16	1.5	1144396284	Casino (1995)	Crime|Drama
3	24	16	4.0	963468757	Casino (1995)	Crime|Drama
4	29	16	3.0	836820223	Casino (1995)	Crime|Drama

4.3 Euclidean Distance

The Similarity Score calculation takes place using the Euclidean Distance method. User1 and User2 acts as the dataset and the input user rating respectively. It is simply calculating the distance between the two points.

```
function euclideanDistance (user1, user2) {
  const n = _.size(user1.reviews)
  let coefficient = 0

  if (n === 0) {
    return n
  }

  for (let i = 0; i < n; i++) {
    coefficient += Math.pow(user1.reviews[i].rating - user2.reviews[i].rating, 2)
  }

  return 1 / (1 + Math.sqrt(coefficient))
}
```

Figure 4: JavaScript Snippet of Euclidean Distance

4.3.1 Formulating Euclidean Distance

$$\sqrt{(p_1-q_1)^2+(p_2-q_2)^2+\ldots+(p_n-q_n)^2} = \sqrt{\sum_{i=1}^{n}(p_i-q_i)^2}$$

Equation 1: Generalizing Euclidean Distance Formula

Exploratory Data Analysis of Movies Category

We can see that the integer values have taller bars than the floating values since most of the users assign rating as integer value i. e.1, 2, 3, 4 or 5. Furthermore, it is evident that the data has a weak normal distribution with the mean of around 3.5. There are a few outliers in the data as well.

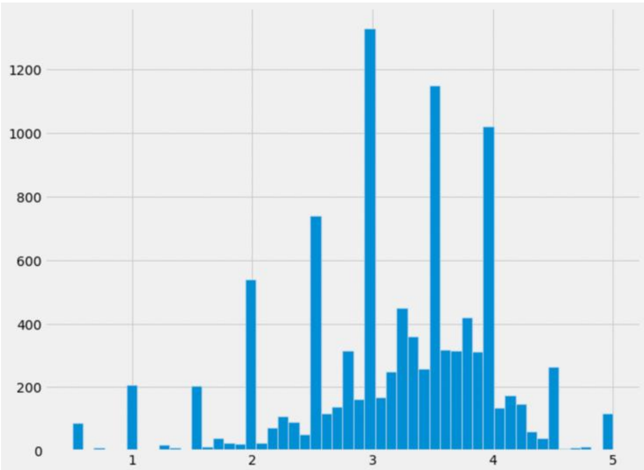

Figure 5: Plotting histogram for average ratings

Movies with a higher number of ratings usually have a high average rating as well since a good movie is normally well -

known and a well - known movie is watched by a large number of people, and thus usually has a higher rating.

```
plt.figure(figsize=(10,8))
plt.rcParams['patch.force_edgecolor'] = True
sns.jointplot(x='rating', y='rating_counts', data=ratings_mean_count, alpha=0.4)
```

The graph shows that, in general, movies with higher average ratings actually have more number of ratings, compared with movies that have lower average ratings.

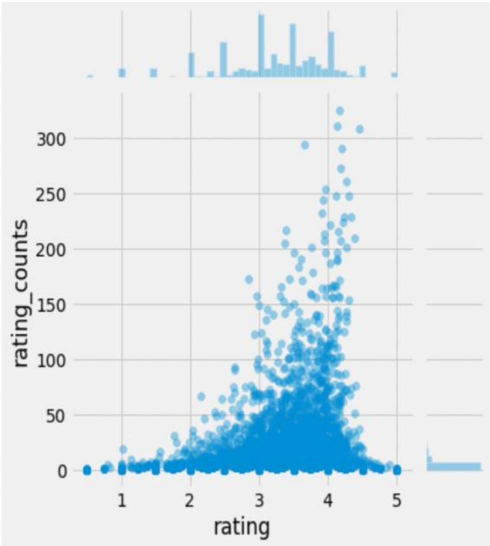

Figure 6: Plotting average ratings against no. of ratings

Collaborative Based Filtering

Collaborative based filtering approach builds a model from the past behavior of the user such as items previously purchased or chosen or the numerical ratings given to those items, as well as similar decisions made by other users. This approach is then used to predict items (or ratings for other items) that the user may have interest in.

There are two types of Collaborative filter recommender systems:

1) **User - based collaborative filtering**: Items are recommended to a user based on the items that have been liked by user similar to other users.
2) **Item - based collaborative filtering**: Predict similar items based on user's previous ratings.

In this project we are using User based collaborative filtering approach as the similarity score is calculated between the two users who have given ratings and it's facilitated by the Euclidean distance. The item which has the least deviation from the distance mean becomes the first item that the recommender system will recommend.

Less the deviation from mean decides the order of ranking of recommended items to the user.

Ranking of Recommended Items

We have seen how the Similarity Score is calculated using Euclidean distance. We create a hash in JavaScript comprising of ranking, total, and Similarity Sum. For each category and each item, we are calculating the total value by multiplying the Similarity Score with by multiplying the Similarity Score with the ratings of similar users for that category. The Similarity Sum (SimSum) is calculated by adding the Similarity Score of each item for that category.

```
let recommCategory = Object.keys(recommendations);
console.log(recommCategory)
for(let i=0; i< recommCategory.length; i++) {
    let eachCategoryValue = Object.keys(recommendations[recommCategory[i]]);
    console.log(eachCategoryValue)
    for(let j= 0; j< eachCategoryValue.length; j++) {
        console.log(recommendations[recommCategory[i]][eachCategoryValue[j]])
        recommendations[recommCategory[i]][eachCategoryValue[j]].ranking =
        recommendations[recommCategory[i]][eachCategoryValue[j]].total /
        recommendations[recommCategory[i]][eachCategoryValue[j]].simSum
    }
 }
let finalAnswer = Object.keys(recommendations[selectedCategory])
// Sort movies by ranking
finalAnswer.sort(byRanking);
function byRanking(a, b) {
    return recommendations[selectedCategory][b].ranking -
    recommendations[selectedCategory][a].ranking;
}
```

Figure 7: Ranking of Recommended Items Snippet

The ranking is decided by the dividing the total value with the SimSum. Now among those decimal numbers given to each item, sorting is performed on it in ascending order.

The top two are returned to the user on the frontend.

5. Conclusions

Through the medium of this project, we have explored the various types of recommender system particularly the Multi Criteria Recommendation System and Cross Domain Recommendation System.

We have implemented a comprehensive recommendation system that involves both multi criteria and multi category.

We have also done exploratory analysis on one of the categories i.e., movies. We have drawn following inferences from it:
1) Movies with higher average ratings actually have more number of ratings, compared with movies that have lower average ratings.
2) Data has a weak normal distribution with the mean of around 3.5.

We have used various pandas and NumPy libraries for doing exploratory data analysis and implemented the Recommendation and Similarity Score logic in JavaScript. Considering the 2MC - RS system is very comprehensive and has not been implemented in the past, we have opened the opportunities for exploration and analysis as well as prediction in this space.

6. Acknowledgement

The authors would wish to thank Dr. A. Shanthini, Assistant Professor, Department of Information Technology, SRMIST, for her sincere efforts in helping and guiding him through the research paper. Her comments have helped in greatly improving the manuscript.

References

[1] Abel, F. Herder, E. Houben, G. J., Henze, N., Krause D.: Cross - system user modeling and personalization on the social web. User Modeling and User - Adapted Interaction 1 - 42 (2011)
[2] Azak, M.: Crossing: A Framework to Develop Knowledge - based Recommenders in Cross Domains. MSc thesis, Middle East Technical University (2010)
[3] Berkovsky, S., Kuflik, T., Ricci, F.: Entertainment Personalization Mechanism through Cross - domain User Modelling.1st International Conference on Intelligent Technologies for Interactive Entertainment, pp.215–219 (2005)
[4] Braunhofer, M., Kaminskas, M., Ricci, F.: Location - awareMusicRecommendation. International Journal of Multimedia Information Retrieval 2 (1), pp.31–44 (2013)
[5] Chung, R., Sundaram, D., Srinivasan, A.: 2007. IntegratedPersonal Recommender Systems.9th International Journal