International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 1 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT19511

Movie Recommended System by Using Collaborative Filtering

Bheema Shireesha[1], Navuluri Madhavilatha[2], Chunduru Anilkumar[3]

[1,3]Assistant Professor, Department of Computer Science and Engineering, Dr. APJ Abdul Kalam IIIT Ongole, Andhra Pradesh, India

[2]Guest Faculty, Department of Computer Science and Engineering, Dr. APJ Abdul Kalam IIIT Ongole, Andhra Pradesh, India

ABSTRACT

Recommendation system helps people in decision making an item/person. Recommender systems are now pervasive and seek to make profit out of customers or successfully meet their needs. Companies like Amazon use their huge amounts of data to give recommendations for users. Based on similarities among items, systems can give predictions for a new item's rating. Recommender systems use the user, item, and ratings information to predict how other users will like a particular item. In this project, we attempt to under- stand the different kinds of recommendation systems and compare their performance on the Movie Lens dataset. Due to large size of data, recommendation system suffers from scalability problem. Hadoop is one of the solutions for this problem.

Keywords : Hadoop, Collaborative filtering, Machine Learning, HDFS, RSSI

I. INTRODUCTION

Recommendations system is a type of information filtering system which attempts to predict the preferences of a user, and make suggests based on these preferences. There are a wide variety of applications for recommendation systems. These have become increasingly popular over the last few years and are now utilized in most online platforms that we use. The content of such platforms varies from movies, music, books and videos, to friends and stories on social media platforms, to products on e-commerce websites, to people on professional and dating websites, to search results returned on Google [1]. Due to the advances in recommender systems, users constantly expect good recommendations. They have a low threshold for services that are not able to make appropriate suggestions. If a music streaming app is not able to predict and play music that the user likes, then the user will simply stop using it. This has led to a high emphasis by tech companies on improving their recommendation systems. However, the problem is more complex than it seems. For recommendation, our proposed system uses collaborative filtering machine learning algorithm. Collaborative filtering (CF) is a machine learning algorithm which is widely used for recommendation purpose. Collaborative filtering finds nearest neighbor based on the similarities. The metric of collaborative filtering is the rating given by the user on a particular item. Different users give different ratings to items. Users, who give almost same rating to items, are the nearest neighbors. In case of User based collaborative filtering, based on the ratings given by the users, nearest neighbors has been find. Item based collaborative filtering predicts the similarity among items. To recommend an item, items which are liked by the user in his past have been found. Item which is similar to those items has been recommended [2]. Internet contains a huge volume of data for recommendation purpose. Due to size of data, if recommendation computation has been done in single system, then performance may degrade, and we cannot find an efficient solution. Hence we require distributed environment so that computation can be increased and performance of recommendation system gets improve .My goal is to apply a collaborative filtering algorithm in a rating website that collects users' information, such as location and gender, item's information, such as category and description, as well as ratings for items by users. There are many algorithms that could be applied on data to predict a user preference. User-based, Item-based, and Model-based methods are ways of predicting a user preference. The number of users, items, or clusters in each one respectively will determine the function performance [3]. However, the most well-known and common one is User-based Collaborative Filtering. In this algorithm, we predict an item's rate for a user by collecting information about this user and similar users.

Machine Learning:

Applying machine learning in real-time using Collaborative Filtering. Parsing data retrieved from a database and predicting user preference. Evaluating different approaches of recommender systems. What I was trying to do was to build a system that collects information and then uses the stored data in a machine learning algorithm [4]. Predicting users' preferences using data may give more accurate results than any algorithm that does not use previous data. Most systems like Amazon, eBay, and others suggest things to users based on similarities among users, items, or both. This will make those systems more personalized and efficient from a user's perspective. Commercial and trading systems gain trust and profits using such systems if they successfully predict what users want at what time and where. The datasets were created and averaged [5]. The corresponding measurements and analysis were made. The remaining part of this report is as follows: Related Works, Measurement Setup, Proposed Work, Measurements Analysis, Results, Discussions, Future Work.

II. RELATED WORK

2.1 Collaborative Filtering

In the collaborative filtering algorithm, the system has a recorded set of items and users and how the users rated those items. Then algorithm is used to predict the rating for a user who has not rated the item yet. A rating for an item can be predicted from the ratings given to the item by users who are similar in taste to the given user. The traditional collaborative filtering algorithms include User-based, Item-based, and Model based methods. To explain how these methods works we are going to use the following notations [6]. "Let U be a set of N users and I a set of M items. The traditional collaborative filtering algorithms include User-based, Item-based, and Model based methods. To explain how these methods works we are going to use the following notations. "Let U be a set of N users and I a set of M items. Vui denotes the rating of user u U on item i I, and SI stands for the set of items that user u has rated".

2.2 User Based Collaborative Filtering

In this method, we predict the user behavior against a certain item using the weighted sum of deviations from mean ratings of users that previously rated this item and the user mean rate. The weight that we previously mentioned can be calculated using Pearson Correlation according to the following formula:

$$\overline{v_u} = \frac{\sum_{i \in S_u} v_{ui}}{|S_u|}$$

The prediction formula is stated as below:

$$w(a,u) = \frac{\sum_{i \in S_a \cap S_u} (v_{ai} - \overline{v_a})(v_{ui} - \overline{v_u})}{\sqrt{\sum_{i \in S_a \cap S_u} (v_{ai} - \overline{v_a})^2 \sum_{i \in S_a \cap S_u} (v_{ui} - \overline{v_u})^2}}$$

2.3 Model-Based Collaborative Filtering

In this method, the system will use an unsupervised learning method to partition the space and then classify the users using a similarity metric to a segment or a cluster. To avoid having large clusters, systems use different methods to generate more small practical clusters. This process starts with an initial set of clusters that contains only one user, and then repeatedly assigns users to these clusters based on a similarity metric. Limiting features may become necessary to reduce clustering complexity. The system will create vectors for each segment and match the user to the vector. Nonetheless, the user may be classified as belonging to more than one cluster with a measure of similarity strength .

$$dist(a,u) = \sqrt{\frac{\sum_{\{i \in S_a \cap S_u\}} (v_{ai} - v_{ui})^2}{|\{i \in S_a \cap S_u\}|}}$$

III. PRELIMINARIES

3.1 Defination1

Recommended system: A recommendation system is a type of information filtering system which attempts to predict the preferences of a user, and make suggests based on these preferences. There are a wide variety of applications for recommendation systems. These have become increasingly popular over the last few years and are now utilized in most online platforms that we use [7]. The content of such platforms varies from movies, music, books and videos, to friends and stories on social media platforms, to products on ecommerce websites, to people on professional and dating websites, to search results returned on Google. Often, these systems are able to collect information about a user's.

3.2 Definition 2

Collaborative Filtering: Collaborative Filtering techniques make recommendations for a user based on ratings and preferences data of many users. The main underlying idea is that if two users have both liked certain common items, then the items that one user has liked that the other user Search Engine Architecture.

3.3 Definition 3

Content Based Recommendation: Content Based Recommendation algorithm takes into account the likes and dislikes of the user and generates a User Profile. For generating a user profile, we take into account the item profiles (vector describing an item) and their corresponding user rating. The user profile is the weighted sum of the item profiles with weights being the ratings user rated. Once the user profile is generated, we calculate the similarity of the user profile with all the items in the dataset, which is calculated using cosine similarity between the user profile and item profile. Advantages of Content Based approach is that data of other users is not required and the recommender engine can recommend new items which are not rated currently, but the recommender algorithm doesn't recommend the items outside the category of items the user has rated.

IV. PROPOSED APPROACH

4.1 Proposed System

For recommendation, our proposed system uses collaborative filtering machine learning algorithm. Collaborative filtering (CF) is a machine learning algorithm which is widely used for recommendation purpose. Collaborative filtering finds nearest neighbor based on the similarities. The metric of collaborative filtering is the rating given by the user on a particular item. Different users give different ratings to items. Users, who give almost same rating to items, are the nearest neighbors. In case of User based collaborative filtering, based on the ratings given by the users, nearest neighbors has been find. Item based collaborative filtering predicts the similarity among items. To recommend an item, items which are liked by the user in his past have been found. Item which is similar to those items has been recommended. Internet contains a huge volume of data for recommendation purpose. Due to size of data, if recommendation computation has been done in single system, then performance may degrade, and we cannot find an efficient solution.

Hence we require distributed environment so that computation can be increased and performance of recommendation system gets improve. My goal is to apply a collaborative filtering algorithm in a rating website that collects users' information, such as location and gender, item's information, such as category and description, as well as ratings for items by users. There are many algorithms that could be applied on data to predict a user preference. User-based, Item-based, and Model-based methods are ways of predicting a user preference [8]. The number of users, items, or clusters in each one respectively will determine the function performance. However, the most well-known and common one is User-based Collaborative Filtering. In this algorithm, we predict an item's rate for a user by collecting information about this user and similar users.

Memory-based algorithms approach the collaborative filtering problem by using the entire database. As described by Breese et. al , it tries to find users that are similar to the active user (i.e. the users we want to make predictions for), and uses their preferences to predict ratings for the active user. This page will talk about the general ideas; for specific equations and implementations, consult the Breese etc. All paper and/or our code.

Item-based collaborative filtering is a model-based algorithm for making recommendations. In the algorithm, the similarities between different items in the dataset are calculated by using one of a number of similarity measures, and then these similarity values are used to predict ratings for user-item pairs not present in the dataset. The system will search group for users by using k-means to find the distance between users, the group of users and clustering of users. The system clustering with the k- means algorithms by measuring the distance of each data point from the center of the 10 groups by using Euclidean distance and calculated information will be stored in the database.

4.2 User Based CF

The dataset is firstly loaded into Hadoop distributed file system (HDFS). Then we perform User-based CF using Mahout [9]. We take rating matrix, in which each row represents user and column represents item, corresponding row-column value represents rating which is given by a user to an item. Absence of rating value indicates that user has not rated the item yet. There are many similarity measurement methods to compute nearest neighbors. We have used Pearson correlation coefficient to find similarity between two users [10]. Hadoop is used to calculate the similarity. The output of the Hadoop Map phase i.e. userid and corresponding itemid are passed to reduce phase. In reduce phase, output has been generated and sorted according to userid. Output again has been stored in HDFS.

4.3 Item Based CF

3.2 Item Based CF Dataset is loaded into HDFS, then using Mahout we performs Item based CF. Past information of the user, i.e. the ratings they gave to items are collected. With the help of this information the similarities between items are builded and inserted into item to item matrix. Algorithm selects items which are most similar to the items rated by the user in past. In next step, based on top-N recommendation, target items are selected.

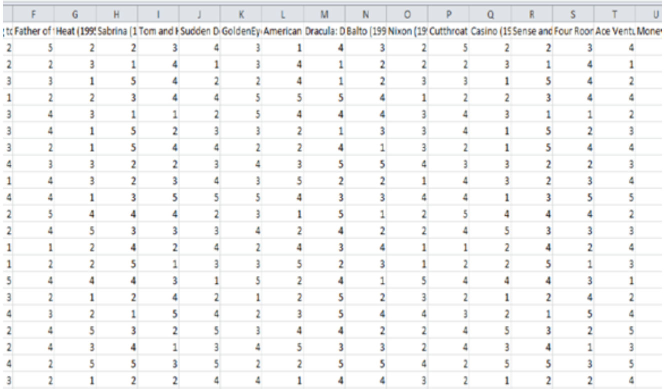
V. EXPERIMENTAL ANALYSIS

5.1 Baseline methods

We try out the following simple baseline methods to give us an idea of the performance to expect from. The reason behind these inappropriate results is, RSSI provides advantages but further research is necessary for its sensitive reaction due to environment. RSSI ranging can be inaccurate and inconsistent especially in indoor environment

[8]. More over the received signal strength can vary considerably over small distances and small time scales, due to multipath fading received signals or path loss can exhibit wide variations even when d changes by as little as a few centimeters in the case of 802.15.4 radios[8].

The data user given movies rating



	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
; tc	Father of	Heat (199	Sabrina (1	Tom and H	Sudden D	GoldenEy	American	Dracula: D	Balto (199	Nixon (19	Cutthroat	Casino (19	Sense and	Four Roon	Ace Ventu	Mone
2	5	2	2	3	4	3	1	4	3	2	5	2	2	3	4	
2	2	3	1	4	1	3	4	1	2	2	2	3	1	4	1	
3	3	1	5	4	2	2	4	1	2	3	3	1	5	4	2	
1	2	2	3	4	4	5	5	5	4	1	2	2	3	4	4	
3	4	3	1	1	2	5	4	4	4	3	4	3	1	1	2	
3	4	1	5	2	3	3	2	1	3	3	4	1	5	2	3	
3	2	1	5	4	4	2	2	4	1	3	2	1	5	4	4	
4	3	3	2	2	3	4	3	5	5	4	3	3	2	2	3	
1	4	3	2	3	4	3	5	2	2	1	4	3	2	3	4	
4	4	1	3	5	5	5	4	3	3	4	4	1	3	5	5	
2	5	4	4	4	2	3	1	5	1	2	5	4	4	4	2	
2	4	5	3	3	3	4	2	4	2	2	4	5	3	3	3	
1	1	2	4	2	4	2	4	3	4	1	1	2	4	2	4	
1	2	2	5	1	3	3	5	2	3	1	2	2	5	1	3	
5	4	4	4	3	1	5	2	4	1	5	4	4	4	3	1	
3	2	1	2	4	2	1	2	5	2	3	2	1	2	4	2	
4	3	2	1	5	4	2	3	5	4	4	3	2	1	5	4	
2	4	5	3	2	5	3	4	4	2	2	4	5	3	2	5	
2	4	3	4	1	3	4	5	3	3	2	4	3	4	1	3	
4	2	5	5	3	5	2	2	5	5	4	2	5	5	3	5	
3	2	1	2	2	4	4	1	4	4	3	2	1	2	2	4	

Table 1 User gives movies rating

User_Id	Movie_Id	Rating
501	124	5
501	133	2
501	140	4

Table 2 The calculation of the similarities between the User 1 and User 3

	Father of the bride	Golden eye	Casino	Four rooms	Money train	Get shorty	Assassins
User_1	1	4	?	3	?	3	2
User_3	3	?	1	?	2	1	2

5.1.1 Global average:

The global average technique serves as a simple baseline technique. The average rating for all users across all movies is computed. This global average serves as a prediction for all the missing entries in the ratings matrix.

5.1.2 User average:

All users exhibit varying rating behaviors. Some users are lenient in their ratings, whereas some are very stringent giving lower ratings to almost all movies. This user bias needs to be incorporated into the rating predictions. We compute the average rating for each user. The average rating of the user is then used as the prediction for each missing rating entry for that particular user. This method can be expected to perform slightly better than the global average since it takes into account the rating behavior of the users into account.

5.1.3 Movie average:

Some movies are rated highly by almost all users whereas some movies receive poor ratings from everyone. Another simple baseline which can be expected to perform slightly better than the global average is the movie average method. In this technique, each missing rating entry for a movie j is assigned the average rating for the movie j.

5.1.4 Adjusted average:

This simple method tries to incorporate some information about the user i and the movie j when making a prediction for the entry ri j. We predict a missing entry for user I and movie j, by assigning it the global average value adjusted for the user bias and movie bias.

The adjusted average rating is given by
ri j = lobalav + (uav(i)ij) + (mav(j)ij)

The user bias is given by the difference between the average user rating and the global average rating. The movie bias is given by the difference between the average movie rating and the global average rating. Consider the following example which demonstrates the working of the adjusted average method.

5.2 Dateset:

For experiment, we have used MovieLens dataset of size 1M. The dataset contains 10000054 ratings and 95580 tags applied to 10681 movies by 71567 users. There are three files, movies.dat, ratings.dat and tags.dat. Ratings data file has at least three columns; those are UserId, given by user to movie.

5.3 Result Analysis:

For movie recommendation, important factor is the list of recommended items as soon as possible. Since we are using Hadoop, speedup and efficiency varies as number of nodes varies. To analyze this we have obtained the number of movies which are recommended as threshold changes, Speedup and efficiency according to number of nodes.

VI. FUTURE WORK AND CONCLUSION

Most obvious ideas is to add features to suggest movies with common actors, directors or writers. In addition, movies released within the same time period could also receive a boost in likelihood for recommendation. Similarly, the movies total gross could be used to identify a user's taste in terms of whether he/she prefers large release blockbusters, or smaller indie films. However, the above ideas may lead to over fitting, given that a user's taste can be highly varied, and we only have a guarantee that 20 movies Collaborative filtering is the most successful and popular algorithm in the recommender system's field. It helps customers to make a better decision by recommending interesting items. Even though this algorithm is the best, it suffers from poor accuracy and high running time. To solve these problems, this paper proposed a recommendation approach based on user clustering by using the Euclidian distance to calculate two users to cluster dataset. This method combines clustering and neighbors' vote to generate predictions. In the future there may be techniques, fuzzy c-means in the group stages of the first system to provide a more effective segmentation.

VII. REFERENCES

[1]. A Survey of Collaborative Filtering Techniques;https://www.hindawi.com/journals/aai/2009/421425/.,2017, Vol. 13(7).

[2]. Google News Personalization: Scalable Online Collaborative Filtering; Das et al; https://www2007.org/papers/paper570.pdf

[3]. Intro to Recommender Systems: Collaborative Filtering;http://blog.ethanrosenthal.com/2015/11/02/intro-to-collaborative -filtering/"

[4]. Zhan J, Hsieh CL, Wang IC, Hsu TS, Liau CJ, Wang DW. Privacy-preserving collaborative recommender systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews).

[5]. Gong S. A collaborative filtering recommendation algorithm based on user clustering and item clustering. Journal of Software. 2010; 5(7):745-52.

[6]. Manvi SS, Nalini N, Bhajantri LB. Recommender system in ubiquitous commerce. In international conference on electronics computer technology 2011 (pp. 434-8). IEEE.

[7]. Pu P, Chen L, Hu R. A user-centric evaluation framework for recommender

systems. In proceedings of the fifth ACM conference on recommender systems 2011 (pp. 157-64). ACM.

[8]. Hu R, Pu P. Acceptance issues of personality-based recommender systems. In proceedings of the third ACM conference on recommender systems 2009 (pp. 221-4).ACM."

[9]. Pathak B, Garfinkel R, Gopal RD, Venkatesan R, Yin F. Empirical analysis of the impact of recommender systems on sales. Journal of Management Information Systems. 2010; 27(2):159-88. " ,

[10]. Witten IH, Frank E, Hall MA. Data mining: practical machine leaning toots and techniques. Morgan Kaufmann Publishers, Elsevier; 2011."

Cite this article as :

Bheema Shireesha, Navuluri Madhavilatha, Chunduru Anilkumar, "Movie Recommended System by Using Collaborative Filtering", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 1, pp. 09-15, January-February 2019.
Available at
doi : https://doi.org/10.32628/CSEIT19511
Journal URL : http://ijsrcseit.com/CSEIT19511