Journal for Research | Volume 02| Issue 03 | May 2016
ISSN: 2395-7549

Similarity Measures for Recommender Systems: A Comparative Study

Mr. Sridhar Dilip Sondur
PG Student
Department of Information Science & Engineering
R. V. College of Engineering, Bengaluru

Mr. Amit P Chigadani
PG Student
Department of Information Science & Engineering
R. V. College of Engineering, Bengaluru

Dr. Shantharam Nayak
Professor
Department of Information Science & Engineering
R. V. College of Engineering, Bengaluru

Abstract

Recommender Systems have the ability to guide the users in a personalized way to interesting items in a large space of possible options. They have fundamental applications in e-commerce and information retrieval, providing suggestion that prune large information spaces so that users are directed towards those items that best meets the needs and preferences. A variety of approaches have been proposed but collaborative filtering has been the most popular and widely used which makes use of various similarity measures to calculate the similarity. Collaborative Filtering takes the user feedback in the form of ratings in an application area and uses it to find similarities and differences between user profiles to generate recommendations. Collaborative Filtering makes use of various similarity measures to calculate the similarity or difference between the users. This paper provides an overview on few important similarity measures that are currently being used. Different similarity measures provide different results against same input parameters. So, to understand how various similarity measures behave when they are put in different contexts but with same input, few observations are made. This paper also provides a comparison graph to help understand the results of different similarity measures.

Keywords: Recommender systems, Collaborative filtering, Similarity measures

I. INTRODUCTION

The many e-commerce sites provide millions of products on sale. Choosing among so many products becomes a challenging job for the customer. Recommender Systems emerge in response to this problem. Recommender Systems are used for providing quality recommendations which helps to guide the customer in making decisions in buying the products. Recommender Systems are basically used in e-commerce sites in which the input to the system will be the analysis of buying behavior of the customers which is used to produce the recommendation list of items. Recommender Systems changed the way people find products, information and the main goal of the recommender systems is to provide the customer with accurate and good quality recommendations. Almost all the recommender systems usually start by searching for group of customers who have purchased or rated similar items and overlap the current user's purchased or rated items [1]. There are many implementations of recommender systems which are based on various factors and are applied for different contexts such as Bio-inspired retail recommender system [9], hyper parameter optimization for recommender system [10] or recommender system based on semantic similarity [7].

One of the first and widely used recommender technologies is Collaborative Filtering [1] [2] [3] [4]. Collaborative Filtering (CF) filters the information for a user based on the collection of user profiles having similar interests. The collaborative filtering algorithms are classified as user-based [3] [5] and item-based [1] [4]. Recommender systems need to store information about the user preferences known as the user profile. Users' profiles can be collected either explicitly or implicitly. One can explicitly ask users to rate what they have used/purchased. Such a profile is filled explicitly by the user ratings. An implicit profile is based on passive observation and contains user's historic interaction data. Based on this strategy, the items that other users have purchased which are similar to the target user are recommended. The similarity between the two users is calculated with the help of the ratings made by the other users. Finding similarity between these users is the most crucial task because the accuracy and the quality of the recommendations rely majorly on them. There are many similarity measures for finding similarity between users and items which throws back the degree of closeness and the degree of separation between users and items. Choosing a perfect similarity measure is very crucial for collaborative filtering and hence for recommender system because different similarity measures will provide different results in different contexts of the information [8].

In general, depending on the property of the users or items information or the measure itself, the similarity measure provides a single numeric value which is the distance or similarity between two users or items. This paper provides a brief observation on different similarity measures that can be used for collaborative filtering algorithms. The observation provides a broad picture on how different similarity measures work on different data in different contexts.

II. RECOMMENDATION TECHNIQUES

In order to implement its core function, identifying the useful items for the user, a RS must predict that an item is worth recommending. In order to do this, the system must be able to predict the utility of some of them, or at least compare the utility of some items, and then decide what items to recommend based on this comparison. The prediction step may not be explicit in the recommendation algorithm but we can still apply this unifying model to describe the general role of a RS [12]. Francesco Ricci et. Al. has provided differences between 6 different techniques. 3 important of them are explained as follows:

A. Collaborative Filtering:

Collaborative Filtering is a popular recommendation algorithm that provides recommendations on the behaviors or ratings of other users in the system. The basic idea behind this technique is that other users' information can be selected and aggregated in such a way as to provide a reasonable prediction to the current user's preference [13]. If user agrees about the quality and relevance about the items, then they will likely agree about other items. Collaborative Filtering algorithms are of two types, (i) User-based, (ii) Item-based. In user-based CF, the target user's preferences are matched with all the other similar user's preferences to create the recommendation list. In Item-based CF, the history of items purchased of the target user is matched with other users and the list of recommendations is generated.

B. Contend-Based Filtering:

The content based approach provides recommendations which are based on information on the content of items rather than on other user's opinions. It uses a machine learning algorithm to induce the profile of the user preferences from examples based on a feature description of the content. The content of an item can be structured or unstructured. If we consider the content of a movie as director, writer, cast etc., then each of these attribute can be considered as a feature. But in the case of unstructured items such as text data, deciding on the feature set is more difficult. Content-based recommenders treat suggestions as a user-specific category problem and learn a classifier for the customer's preferences depending on product traits. This approach is based on information retrieval because content associated with the user's preferences is treated as query to the system and unrated items are scored with similar items [12].

C. Hybrid Recommender Systems:

Hybrid recommender systems combine two or more recommendation techniques to gain better performance with fewer of the drawbacks of any individual one. Most commonly, collaborative filtering is combined with some other technique in an attempt to avoid the ramp-up problem [11]. One way is to combine content based and collaborative filtering algorithms in such a way that they produce separate ranked lists of recommendations then merge them to make up the final recommendations. Some notable examples of hybrid recommender systems are Weighted and Switching hybrid recommender systems. A weighted hybrid recommender is one in which the score of a recommended item is calculated from the results of all of the available recommendation algorithms in the system. For example the simplest combined hybrid recommender systems would be a linear combination of recommendation scores. Switching Hybrid recommender system (SH) uses some criterion to switch between recommendation techniques. Example of (SH) recommender system is the DailyLearner that uses a content\collaborative hybrid. In this hybrid content based recommendation algorithm is employed first then collaborative if the first results are not satisfactory.

III. SIMILARITY MEASURES

The most important and crucial step in collaborative filtering algorithms is to find similar items and users. After finding the similar users and items, it is easy to reason about the similarity between these users and items and finally choosing a group of users and items which are most similar to the target user [7].

Following are some of the popular similarity measures metrics that are used in collaborative filtering which are also used for observation in this paper.

A. Euclidean Distance:

The basis of many measures of similarity and dissimilarity is euclidean distance. The distance between vectors X and Y is defined as follows:

$$d(x,y) = \sqrt{\sum_{i}^{n} (x_i - y_i)^2}$$

Where x_i and y_i are rating score of an item given by two different users for the same item n is number of commonly rated items.

In other words, euclidean distance is the square root of the sum of squared differences between corresponding elements of the two vectors. Note that the formula treats the values of X and Y seriously: no adjustment is made for differences in scale. Euclidean distance is only appropriate for data measured on the same scale. As you will see in the section on correlation, the correlation coefficient is (inversely) related to the euclidean distance between standardized versions of the data.

B. Pearson Correlation Coefficient:

Unlike the Euclidean Distance similarity score (which is scaled from 0 to 1), this metric measures how highly correlated are two variables and is measured from -1 to +1. Similar to the modified Euclidean Distance, a Pearson Correlation Coefficient of 1 indicates that the data objects are perfectly correlated but in this case, a score of -1 means that the data objects are not correlated. In other words, the Pearson Correlation score quantifies how well two data objects fit a line.

$$PC(u,v) = \frac{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)(r_{vi} - r_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)^2 \sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}}$$

C. Cosine Similarity:

Usually cosine similarity metric is used for estimate the similarity between two instance a and b in information retrieval that the objects are in the shape of vector xa and vector xb and calculating the Cosine Vector (CV) (or Vector Space) similarity between these vectors indicate the distance of them to each other.

$$\frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

where A_i is rating of user A and B_i is rating of user B for the same item n is number of commonly rated items.

In the context of item recommendation, for computing user similarities, this measure can be employed in which a user u indicates vector xu, where xui = rui if user u has rated item i and for unrated item considers 0. The similarity between two users' u and v would then be calculated as:

$$CV(u,v) = \cos(X_a, X_b) = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2} \sqrt{\sum_{j \in I_v} r_{vi}^2}}$$

Where ruv once more indicates the items rated by both u and v. A shortcoming of this measure is that it does not examine the differences in the mean and variance of the ratings made by users u and v.

D. Jaccard Coefficient:

The Jaccard coefficient, which is sometimes referred to as the Tanimoto coefficient, measures similarity as the intersection divided by the union of the objects. For text document, the Jaccard coefficient compares the sum weight of shared terms to the sum weight of terms that are present in either of the two documents but are not the shared terms. The formal definition is:

$$SIM_J(\vec{t_a}, \vec{t_b}) = \frac{\vec{t_a} \cdot \vec{t_b}}{|\vec{t_a}|^2 + |\vec{t_b}|^2 - \vec{t_a} \cdot \vec{t_b}}.$$

Where t_a and t_b are ratings of user A and user B for the union of items between the users.

The Jaccard coefficient is a similarity measure and ranges between 0 and 1. It is 1 when the ta = tb and 0 when ta and tb are disjoint, where 1 means the two objects are the same and 0 means they are completely different. The corresponding distance measure is DJ = 1 − SIMJ and we will use DJ instead in subsequent experiments.

IV. OBSERVATION AND EVALUATION OF SIMILARITY MEASURES

The observation is of 4 important similarity measures. The input to the system is the books-rating dataset with different users having different or similar ratings to the various books. All the similarity measures provide similar or different results depending on various factors such as the properties of the information, the context. The following table shows the results of the observation of all the 4 measures.

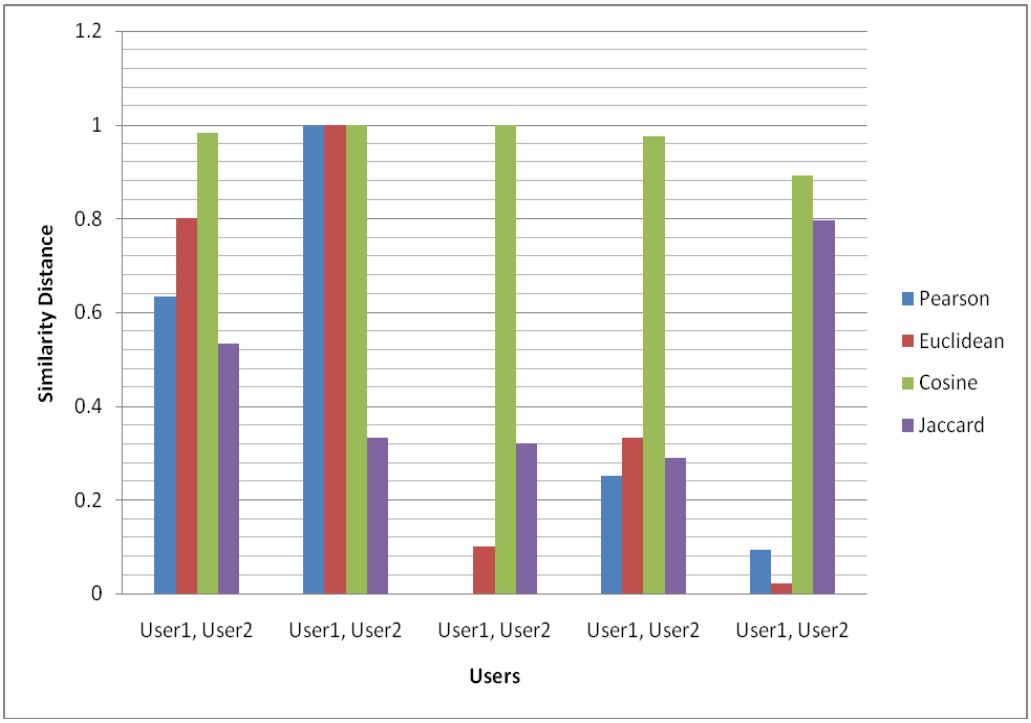

Fig. 1: Observation Results for Similarity Measures

Different similarity measure algorithms used in our survey are Pearson correlation, Euclidean distance, Cosine similarity and Jaccard coefficient. The algorithms used in this paper behave differently in different context. Majority of the algorithms showed the same result in finding the similarity between the users. The resulting values are scaled in the range of 0 to 1 for Euclidean distance, Cosine similarity and Jaccard coefficient, whereas the values for Pearson correlation are from -1 to 1. Value 1 in all the four algorithms represent completely similar and value 0 represents completely dissimilar. Value -1 in Pearson correlation represents the negative similarity between the entities.

There are several constraints while choosing the best algorithm to be used to measure the similarity, for example Pearson coefficient algorithm requires minimum number of objects (items) to be greater than 2 to measure the similarity. Pearson correlation, Euclidean distance and Cosine similarity algorithms consider only the common items that have been rated for measuring the similarity, whereas Jaccard coefficient considers the common items as well as the items that are present in either of the entity. Here entity refers to the collection of rated items by the user. In the following discussion we have tried to make the decision or recommend items based on the values from Fig. 1.

In our first experiment User1 had rated for six items in total and User2 had rated for four items. Both the users had rated for four common items among which, for three items they gave the same score. The other item differed in a factor of 3 points. Cosine similarity function showed the value 0.982 which is almost nearer to 1 (completely similar), whereas the value for Pearson correlation and Euclidean distance showed a value 0.63 and 0.8 respectively. Through visual analytics we can say that Pearson correlation and Euclidean distance showed the acceptable value since the users rating were not completely similar but almost similar. Hence Cosine similarity function has not much effect when only one of the items rated is different. Its value tries to head towards 1.0, though it should not be the case. Jaccard coefficient showed the value of 0.53 which can be considered as partially similar. Jaccard coefficient considers the items that are not in common between the users along with the common items. Since User2 had given rating only for four items compared to six items of User1, Jaccard coefficient showed lesser similarity than the rest three algorithms. So Jaccard coefficient is not a good choice to opt when we want to consider only the common item ratings. In our second experiment, User1 had rated for three items and User2 for four items. Users rated two mutual items, out of which both the users gave same rating for respective items.

All the algorithms except Jaccard coefficient resulted in value 1.0 since all the common items were rated with the same score. As mentioned before Jaccard coefficient considers non common items too while calculating the similarity, hence it showed a value different from 1.0. In our third experiment User1 had rated only one item and User2 had rated four items. The only item rated by User1 was also rated by User2. The difference in rating score was 8. Since there was only one common item between the users, Pearson correlation showed a value 0.0 as mentioned earlier this algorithm is not suitable for less than two common items. On the other hand Cosine similarity value headed to 1.0 because only one item had different rating. Euclidean distance proved to show possibly an acceptable value 0.1 since the users were not much similar. User2 had rated for three more items and hence Jaccard coefficient showed a value of 0.3199. The ratings made by User2 for remaining items were quite similar to the one made by User1. Hence the value headed towards partially similar. In our entire above observation, we didn't explain why we chose Jaccard coefficient algorithm in our experiment. It is best applicable in cases where users have rated for same number of

items and possibly for different or same item. Therefore Jaccard coefficient could also be helpful in recommending the items to users based on the number of times they have viewed or purchased the items. Rest of the experiments also showed positive results. We have included only few of them in this paper which has covered all the possible scenarios.

V. CONCLUSION

Recommender system helps any organization to improve their business growth. In this paper we explained about the techniques used for Recommender systems. We discussed various similarity measures which help to achieve Collaborative filtering technique. Comparison of various similarity measure algorithms has been proved to show that each algorithm works better and provide accurate results in different scenarios. All the four algorithms explained showed a positive result in most of the experiments. We have also tried to explain which all algorithms can be used in each scenario. Choosing the right algorithm also makes the recommender system work better. This paper shows the comparison study of various algorithms required to achieve collaborative filtering.

Our future work will be on the recommender system based on item-to-item based collaborative filtering. This technique is rather finding similarity between items than to find similar users. So, how these similarity measures behave when item-to-item based collaborative filtering is used with same input parameters is our future work.

REFERENCES

[1] Greg Linden, Brent Smith, and Jeremy York, "Amazon's item-to-item collaborative filtering", IEEE Computer Society, Feb – 2003.
[2] Badrul M. Sarwar, George Karypis, Joseph Konstan , and John Riedl, "Recommender Systems for Large-scale E-Commerce: Scalable Neighborhood Formation Using Clustering"
[3] Zhi-Dan Zhao, Ming-Sheng Shang, "User-based Collaborative-Filtering Recommendation Algorithms on Hadoop", International Conference on Knowledge Discovery and Data Mining, 2010.
[4] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, "ItemBased Collaborative Filtering Recommendation Algorithms", 2001.
[5] Maddali Surendra Prasad Babu, Boddu Raja Sarath Kumar, "An Implementation of the User-based Collaborative Filtering Algorithm", (IJCSIT, Vol. 2 (3), 2011.
[6] Chen Sun, Rong Gao and Hongsheng Xis, "BIG DATA BASED RETAIL RECOMMENDER SYSTEM OF NON E-COMMERCE", 5th ICCCNT, 2014
[7] Karamollah Bagheri Fard, Mehrbakhsh Nilashi, Mohsen Rahmani, Othman Ibrahim, "Recommender System Based on Semantic Similarity", IJECE, Vol. 3, No. 6, December 2013, pp. 751~761.
[8] Anna Huang, "Similarity Measures for Text Document Clustering", NZCSRSC 2008, April 2008.
[9] Soumya Banarjee, Neveen I. Ghali, Arup Roy, Aboul Ella Hassanein. "A Bio-Inspired Perspective towards Retail Recommender System: Investigating Optimization in Retail Inventory". (12th ISDA 2012) IEEE Press.
[10] Simon Chan, Philip Treleaven, Licia Capra, "Continuous Hyperparameter Optimization for Large-scale Recommender Systems", 2013 IEEE International Conference on Big Data.
[11] Tranos Zuva, Sunday O. Ojo, Seleman M. Ngwira and Keneilwe Zuva, "A Survey of Recommender Systems Techniques, Challenges and Evaluation Metrics", IJETAE, Volume 2, Issue 11, November 2012
[12] Francesco Ricci, Lior Rokach, Bracha Shapira and Paul B. Kantor, "Recommender Systems Handbook".
[13] Michael D. Ekstrand, John T. Riedl and Joseph A. Konstan, "Collaborative Filtering Recommender Systems", FTHCI Vol. 4, No. 2 (2010) 81–173
[14] Robin Burke, "Hybrid Recommender Systems: Survey and Experiments", California State University, Fullerton.