International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 6– No. 9, March 2014 – www.ijais.org

Non-Personalized Recommender Systems and User-based Collaborative Recommender Systems

Anil Poriya
Thakur College of Engineering and technology
University of Mumbai

Neev Patel
Thakur College of Engineering and technology
University of Mumbai

Tanvi Bhagat
Thakur College of Engineering and technology
University of Mumbai

Rekha Sharma, Ph. D
Thakur College of Engineering and Technology
University of Mumbai

ABSTRACT

Recommender Systems have become an important part of large e-commerce websites. One can safely say, they are the bread and butter of large E-Commerce websites. We may have seen the "customers who bought item1 also bought item2" feature of sites such as amazon.com and found it useful. This is exactly what recommender systems strive to achieve. The basic notion behind introducing recommender systems in websites is simple: to help the customers or users using the website in making their decisions. In general the goal of any recommendation system is to present users with a highly relevant set of items. Recommendation algorithms can be generally classified into three types (i) Non-Personalized, (ii) Content-Based, and (iii) Collaborative Filtering algorithms. Apart from these three approaches, we also have hybrid approach wherein we can combine the above mentioned algorithms to improve the performance of recommender systems.

Literature survey done on recommender systems shows that a lot of work has been carried out in this area and this paper gives an insight into two very popular recommender systems: Non-personalized and Collaborative recommender systems. The paper gives an insight into two approaches of Non-personalized recommender systems and the User-based approach of Collaborative recommender systems.

General Terms
Non-Personalized recommender systems, Collaborative Filtering techniques, Hybrid recommender systems.

Keywords

User-Based Collaborative Filtering Technique, Item-Based Collaborative Filtering, Content-Based Filtering, Pearson-Correlation.

1. INTRODUCTION
Information on the internet is huge and so is the search for any particular data as internet is ever developing and growing at a tremendous speed. This provided the internet users with many opportunities such as sharing knowledge, information and opinions with other users. All the information was kept on the internet and gradually the information kept getting huge and led to the problem of information overload. Recommendation system is a specific type of information filtering technique that attempts to present information items(such as movies, music, news) that are likely of interest to the user. Recommender systems help users navigating through large product assortments, in making decisions in an e-commerce scenario and overcome information overload. They are of great importance for the success of e-commerce and IT industry nowadays, and gradually have gained popularity in various applications. The fact that web contains tremendous amount of information and it is really difficult to go through each of them is one of the major factors why recommender systems are a priority in any website or web application. Technology has dramatically reduced the barriers to publishing and distributing information. It is time to create technologies that can help us sift through all the available information to find that information which is most valuable to us. The paper aims to look at the different aspects of Non-personalized and Collaborative recommender systems and study them. The paper specifically focuses on Pearson-correlation algorithm for user-based collaborative recommender systems.

1.1 Motivation
The concept of recommender systems generally grows out of the idea of information reuse and persistent preferences. It is an idea that does not begin with computers and technology. It's an idea that one can find in cavemen, ants and other creatures too. We may have seen ants running around in our house. The ants follow in a line from the ants that went before and found food. This is because ants have genetically evolved to leave markers for other ants. These markers serve as a recommender to other ants, showing them the way to food. The similar scenario can be seen in humans. People are more likely to follow something if majority of other users have liked and done that particular thing. Thus the motivation for these projects comes from the fact that in today's world recommending an item to a user has gained much importance and popularity as well. Sometimes users have less time to browse a site and are looking for quick recommendation of products that are hot-trending or which they would probably like. Also some users, quite acceptably are confused when they see a long list of item. They are in a fix- whether he/she would like this or that one. Thus recommender systems can keep a track of each user's taste/likes and accordingly recommend specific items to specific users.

International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 6– No. 9, March 2014 – www.ijais.org

1.2 Terms Used and Definitions

Prediction: Prediction is an estimate of how much a user will like an item. Predictions make bold statements. They are often scaled to match some rating scale. They are also often tied to search or browsing for specific products.

Recommendations: Recommendations do not make bold statements like predictions. They are just suggestions for items that user might like. They are often presented in the form of "top N lists"

Explicit Rating: Explicit ratings, as the name suggests, are the ratings which are taken from the users explicitly. That is, the system may ask the users what they think about a particular item and the users give their views in the form of ratings.

Implicit Rating: Implicit ratings are inferred from users' actions and are not taken or asked directly from user. Data may be collected from users' actions which express preferences.

2. RELATED WORK

A lot of work has been carried out in the area of recommender systems. One of the earliest implementations of collaborative filtering based recommender systems is Tapestry [1], a system which relied on the explicit opinions of people from a close-knit community, such as an office workgroup. Recommender systems can be thought of as an important means for information filtering. Additionally, they are also viewed as a potential method to solve the information overload problem. Amongst all the information filtering techniques, collaborative filtering is the most important technique. Currently, almost all of the large-scale e-commerce systems(such as EBay, Flipkart, Amazon, etc) have used CF methods in order to recommend products to their customers. In addition to this, various Music and News web sites also require CF methods to do personalization. Subsequently, several ratings-based automated recommender systems were developed.

Recommender Systems have (i) the information that the system has prior to the recommendation process, referred to as background data, (ii) the information that the user must feed to the system to generate recommendations, referred to as input data, and (iii) an algorithm that combines the background and input data to generate appropriate recommendations [4]. Other technologies have also been applied to recommender systems, including Bayesian Networks, Clustering and Horting [2]. This paper focuses on the Non-personalized approach and the user-based collaborative filtering approach that can be used to generate useful predictions or recommendations to a naïve user who can just look upon the results and approve of the recommendations.

3. NON- PERSONALIZED APPROACH

Non personalized recommender systems are the most simple type of recommender systems. As suggested by the name, these type of recommender systems do not take into account the personal preferences of the users. The recommendations produced by these systems are identical for each customer. In case of E-Commerce websites, the recommendations can either be manually selected by the online retailer, based on the popularity of items or the recommendations can be the top-N new products [13]. For example, if we go to amazon.com as an anonymous user it shows items that are currently viewed by other members. These systems recommend items to consumers based on what other consumers have said about the items or rated them on an average. As seen earlier recommendations are simply suggestions or list of items that user might like and these recommendations are independent of the consumer. Non-personalized recommender systems mainly use two types of algorithms: Aggregated opinion recommender and Basic product association recommender.

3.1 Aggregated Opinion Approach [12]

There are various online websites which make use of non-personalized recommender systems by displaying the average customer ratings. Some of the famous restaurant guides which suggest restaurants to the users use the aggregated opinion recommender approach. It displays the restaurants with a score which is an average of the ratings given to that restaurant by other customers. This score is basically a measure of how good the restaurant is. Some guides come with descriptors for restaurants and 4 scores(Food, Ambiance, Cost, Quantity). These scores generally range from 0 to 5. The average score is then calculated and displayed along with that restaurant's name. The average score [17] calculated for most of such guides is as given below

Score = round(MEAN(ratings)*10) or simply as

Score =MEAN(ratings*10) **(1)**

Other examples which make use of aggregated opinion recommenders include travel websites (Trip advisor) which give reviews and ratings of different places around the world or movie websites which display movie charts with top-N movies (having the highest average ratings).Averages are useful for an overall sense of what the population feels. However these averages lack context during recommendations. In above examples one bad rating or review has a lot of weight and can pull down an otherwise excellent rating. To tackle this issue, some websites tallied the percentage of people who rated a particular item "good" or "bad". This leads us to the concept of product association recommender.

3.2 Product Association Recommender [14][15]

Non-personalized product association recommenders can provide useful non-personalized recommendations in a context. Majority of the online shopping websites such as amazon.com or flipkart make use of product association recommenders by providing "people who bought item1 also bought item2" feature. Such recommendations are based on what is present in the user's cart. That is, recommendations may not be necessarily specific to the user but specific to what the user is currently doing (viewing/buying). Using the taxonomy of recommender systems we can view product association recommender systems are ephemerally personalized recommender systems. The basic idea for these systems is: People who did some X also did Y. The simple computation of this ranking can be: Percentage of X-buyers who also bought Y.

This is illustrated in the formula below

$$\frac{(X\ AND\ Y)}{X} \tag{2}$$

However this formula does not compensate for the overall popularity of Y. The above formula can be adjusted in the following manner by looking at whether X makes Y more likely than not X (!X)

$$\frac{\frac{(X\ AND\ Y)}{X}}{\frac{(!X\ AND\ Y)}{!X}} \tag{3}$$

This formula focuses on increase in Y associated with X [17]. Another solution to the drawback of formula (2) is using the Association rule mining which uses the lift metric. It is basically non-directional association [13][15].

$$\frac{P(X\ AND\ Y)}{\left(P(X)*P(Y)\right)} \tag{4}$$

More generally association rules look at baskets of products, not just individuals.

3.3 Advantages and Disadvantages

The advantage of this method is that it is easy to implement as only the popular or the highly rated items are displayed to the users. Secondly the data for these recommender systems is easy to collect [1][13].

However, the recommendations in this system are the same to all users and lack personalization and hence might not appeal to everyone. Also these systems face challenges in clustered diverse population (The banana trap problem [16][17]).

4. COLLABORATIVE APPROACH [1][3][16][18]

Unlike content-based algorithms which use user's profile or an item's profile to find matching items with the user, collaborative filtering algorithms use past user's behavior to recommend items to the user. Collaborative recommendation is probably the most familiar and most widely implemented technology. Collaborative Filtering is the one of the most successful recommendation technique. The basic idea of CF-based algorithms is to provide item recommendations or predictions based on the opinions of other like-minded users. The opinions of users can be obtained explicitly from the users or by using some implicit measures. We can divide the entire process of CF-based recommendation generation into three sub-tasks namely, representation, neighborhood formation, and recommendation generation.

First a particular scheme is used to model the products that have already been purchased by a customer. This is carried out in the representation sub-task. The next step is to identify the similar or neighboring customers. This is handled by the neighborhood formation sub-task. Finally the recommendation generation sub-task focuses on finding the top-N products from the neighborhood of customers. It can either be Most-frequent Item recommendation or Association Rule-based recommendation.

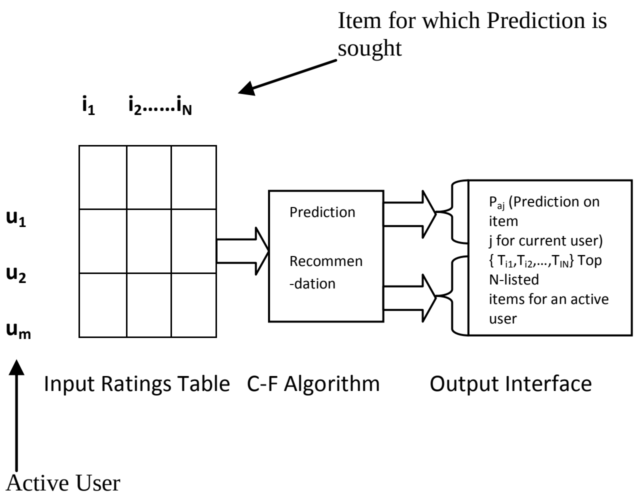

Figure 1.The Basic Collaborative Filtering Process

A schematic diagram of CF algorithm is shown in the above figure. In this picture, we can see the matrix m x n represents user-item data. There is a rating score of each user m on item n at each entry of the matrix. Each individual rating has a numerical scale from 0 to 5. The 0 means the user has not yet rated that item[1].

4.1 Variants of Collaborative Filtering Process [13]

Memory-based Collaborative Filtering Algorithms [13]

Memory-based algorithms utilize the entire user-item database to generate a prediction. These systems employ statistical techniques to find a set of users, known as neighbors, that have a history of agreeing with the target user(i.e., they either rate different items similarly or they tend to buy similar set of items). Once a neighborhood of users is formed, these systems use different algorithms to combine the preferences of neighbors to produce a prediction or top-N recommendation for the active user. The techniques, also known as nearest-neighbor or user-based collaborative filtering, are more popular and widely used in practice.

Model-based Collaborative Filtering Algorithms [13]

Model-based collaborative filtering algorithms provide item recommendations by first developing model of user ratings. Algorithms in this category take a probabilistic approach and envision the collaborative filtering process as computing the expected value of a user prediction, given his/her ratings on other items.
This paper mainly focuses on user-based collaborative filtering using the Pearson-correlation algorithm. The same algorithm can also be applied for item-based collaborative filtering technique.

4.2 User Based Collaborative Filtering

Collaborative filtering approaches build a model from a user's past behavior as well as similar decisions made by other users; then use that model to predict items(or ratings for items) that user may have an interest in. One of the most common example of collaborative filtering is the IMDB website. Here the user can view the most popular movies which have been rated by other users. The user can rate movies based on his/her personal tastes and then subsequently the user can receive a list of recommended movies for him/her based on user's taste(which movie has been rated what). This feature is extremely beneficial for the users as well as the website because a movie that seems excellent to one person may seem dull for another. This is known as user-based collaborative filtering. It basically tries to find the users which are similar to the current user. There are many algorithms to calculate the similarity between the two users in CF systems [1]. We are going to concentrate on the Pearson Correlation Algorithm. It is the most chosen algorithm to use in CF systems. Pearson correlation only computes the similarity between the two users who rate a same item. The Pearson Correlation Score is a measure of how well two sets of data fit on a straight line. One interesting aspect of the Pearson Score is that it corrects for grade inflation. That is, if one product has consistently higher scores than another, there can still be a perfect correlation- if the difference between the ratings is consistent.

The code for this algorithm:

1. First finds the reviewers that reviewed both products. It then
2. Computes the sums and the squared sums of the ratings for the two products, and
3. Computes the sum of the reviews of the products. Finally,
4. It uses these results to compute the Pearson Correlation Score

This algorithm will return a value between -1 and 1, where 1 means two products have exactly the same ratings. Essentially Pearson correlation is cosine similarity over the mean-centered vectors restricted to common items. The cosine similarity [1][2] is given as

$$simil(x,y) = cos\left(\vec{x},\vec{y}\right) = \frac{\vec{x}.\vec{y}}{\left\|\vec{x}\right\|_2 \; X \; \left\|\vec{y}\right\|_2}$$

$$= \frac{\sum_{i\epsilon I_{xy}} r_{x,i}\; r_{y,i}}{\sqrt{\sum_{i\epsilon I_{xy}} {r_{x,i}}^2 \sum_{i\epsilon I_{xy}} {r_{y,i}}^2}}$$

Figure 2.The Cosine Similarity Formula

The Pearson correlation algorithm [1][2] computes the similarity between user x and user y as shown below.

$$sim(x,y) = \frac{\sum_{s\epsilon S_{xy}}(r_{x,y} - \bar{r}_x)(r_{y,s} - \bar{r}_y)}{\sqrt{\sum_{s\epsilon S_{xy}}(r_{x,s} - \bar{r}_x)^2 \sum_{s\epsilon S_{xy}}(r_{y,s} - \bar{r}_y)^2}}$$

Figure 3.The Pearson Correlation Formula(Adjusted Cosine Similarity)

Where S is the set of items where both user x and user y rated.

In User-Based collaborative filtering, we will take inputs from a user who has rated a set of movies and find the users who have similar taste as the current user by comparing the ratings of the current user with other users and derive the list of users who are most similar to the current user. The example proposed in this paper is that of movie recommendations. Different users give different ratings for a set of movies. The ratings of the current user are taken and compared with the ratings of other users. Finally the algorithm displays the users which are most similar to the current user.

```
{
    var reviewer = new Reviewer { Name = "Tanvi" };
    reviewer.AddReview("Inception", 2.5);
    reviewer.AddReview("Tangled", 3.5);
    reviewer.AddReview("Oceans11", 3.0);
    reviewer.AddReview("Snitch", 3.5);
    reviewer.AddReview("Zombieland", 2.5);
    reviewer.AddReview("American Hustle", 3.0);
    return reviewer;
}
public static Reviewer BuildReviewer7()
{
    var reviewer = new Reviewer { Name = "Neev" };
    reviewer.AddReview("Inception", 1.5);
    reviewer.AddReview("Tangled", 2.0);
    reviewer.AddReview("Oceans11",2.0);
    reviewer.AddReview("Snitch", 2.5);
    reviewer.AddReview("Zombieland", 1.0);
    reviewer.AddReview("American Hustle", 1.0);
    return reviewer;
}

public static Reviewer BuildMyReviews()
{
    var reviewer = new Reviewer { Name = "Änil" };
    reviewer.AddReview("Inception", 4.5);
    reviewer.AddReview("Tangled", 5.0);
    reviewer.AddReview("Oceans11", 4.0);
    reviewer.AddReview("Snitch", 4.5);
    reviewer.AddReview("Zombieland", 4.0);
    reviewer.AddReview("The Dark Knight", 5.0);
    return reviewer;
```

Figure 4. A Screenshot of Reviews Built for some users

The algorithm will then calculate the Pearson score for each user with respect to the current user based on the ratings of each user. The results will then be displayed with the user having the highest correlation score displayed first and so on. The scores range from -1 to 1 where 1 indicates that the user is exactly similar to the current user and -1 indicates that the user is not at all similar to the current user.

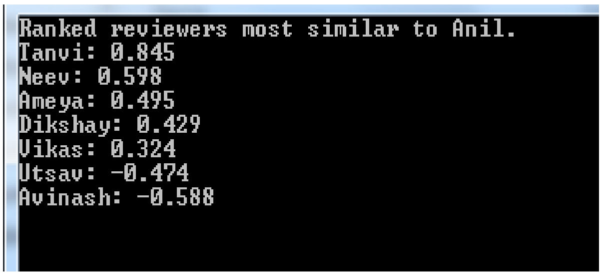

Figure 5. A Screenshot of the result of Pearson correlation score

There is one more type which is the Item-Based collaborative filtering which is based on the similarity of items selected by the current user to other items. In simple words, the "customer who bought item1 also bought item2" feature. In Item-Based collaborative filtering, the customer will buy a product from a list of entities and would be recommended that have been bought along with the item that customer has selected.
The Pearson correlation algorithm to compute similarity between two items is give by [1][2]:

$$sim(i,j) = \frac{\sum_{u \epsilon U}(R_{u,i} - \bar{R_i})(R_{u,j} - \bar{R_j})}{\sqrt{\sum_{u \epsilon U}(R_{u,i} - R_i)^2}\sqrt{\sum_{u \epsilon U}(R_{u,j} - R_j)^2}}$$

Figure 6.The Pearson Correlation Formula for Item Based Similarity

Here R_i is average number of item i, $R_{u,i}$ is number of ratings user u gives on item i.

4.3 Advantages and Disadvantages

The collaborative recommender systems overcome the lack of personalization involved with non-personalized recommender systems [16]. Also no item data is needed for this approach and it is domain independent. The computational time is low for model based approaches [13].

However these systems require some kind of rating feedback and may be cold start for new users and new items. Also the memory based approaches require lots of memory thereby increasing the computational time.

5. CONCLUSION

This paper describes Non-personalized and User-based collaborative recommender systems. The paper also states the advantages and disadvantages of both the systems. Recommender systems are a powerful new technology for extracting additional value for a business from its user databases. These systems help users find items they want to buy from a business. Recommender systems benefit users by enabling them to find items they like. Conversely, they help the business by generating more sales. Recommender systems are a crucial tool in e-commerce on the web. Recommender systems are being stressed by the huge volume of user data in existing corporate databases, and will be stressed even more by the increasing volume of user data available on the web. New technologies are needed that can dramatically improve the scalability of recommender systems.

5.1 Future scope

The future scope would be to try and incorporate dimensionality reduction techniques to allow CF-based algorithms to scale to large data sets and at the same time produce high- quality recommendations. Also a productive and better approach towards recommendation is the Hybrid model. The Hybrid approach [3] overcomes the limitation of content-based and collaborative recommender systems. It basically combines the two methods by inheriting the advantages of each and eliminating the disadvantages. In general, hybrid recommenders are systems that combine multiple recommendation techniques together to achieve a synergy between them. Hybrid approaches can be implemented in several ways: by making content based and collaborative-based predictions separately and then combining them; by adding content based capabilities to a collaborative approach(and vice versa); or by unifying the approaches into one model [3]. Several studies empirically compare the performance of the hybrid with the pure collaborative and content-based methods and demonstrate that the hybrid methods can also be used to overcome some of the common problems in recommender systems such as cold start and sparsity problem. Netflix is a good example of hybrid systems. They make recommendations by comparing the watching and searching habits of similar users(collaborative) as well as by offering movies that share characteristics with films that a user has rated highly(content-based).

6. ACKNOWLEDGMENTS

We would like to show our greatest appreciation to our HOD, Dr. Rekha Sharma. Without her encouragement and guidance this paper would not have materialized. We thank her for constant support and for helping us to understand the different concepts and related topics. We would also like to thank Mrs. Vidyadhari. R. Singh for her beneficial inputs at every stage of the paper. We are also highly indebted to the course "Introduction to Recommender Systems" by University of Minnesota on coursera.org which really got us interested in this topic.

7. REFERENCES

[1] Badrul Sarwar, George Karypis, Joseph Konstan, John Riedl "Analysis of Recommendation Algorithms for E-Commerce", GroupLens Research Group/Army HPC Research Center, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455.

[2] Badrul Sarwar, George Karypis, Joseph Konstan, John Riedl "Item-Based Collaborative Filtering Recommendation Algorithms", GroupLens Research Group/Army HPC Research Center, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455.

[3] Gediminas Adomavicius and Alexander Tuzhilin "Toward the Next Generation of Recommender Systems: A survey of te State-of-the-Art and Possible Extensions", IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6, June 2005.

[4] Rohini Nair and Kavita Kelkar "Implementation of Item and Content based Collaborative Filtering Techniques based on Ratings Average for Recommender Systems", International Journal of Computer Applications(0975-8887), Volume 65- No.24, March 2013.

[5] Prem Melville and Vikas Sindhwani, "Recommender Systems", IBM T.J. Watson Research Center, Yorktown Heights, NY 10598.

[6] Recommender Systems in e-Commerce: Methodologies and Applications of Data Mining, Dr. Bharat Bhasker , K Srikumar,July 29, 2010.

[7] Meisamshabanpoor and Mehregan Mahdavi "Implementation of a Recommender System on Medical Recognition and Treatment", International Journal of e-Education, e-Businees, e-Management and e-Learning, Vol.2, No.4, August 2012.

[8] Recommender Systems Handbook, Ricci, F.; Rokach, L.; Shapira, B.; Kantor, P.B. (Eds.)2011.

[9] Hill, W., Stead, L., Rosenstein, M., and Furnas, G. (1995) "An Algorithmic Framework for Performing Collaborative Filtering", in Proceedings of ACM SIGIR'99. ACM press.

[10] Aggarwal, C. C., Wolf, J.L., Wu, K., and Yu, P.S.(1999) "Horting Hatches an Egg: A New Graph-theoretic Approach to Collaborative Filtering", in Proceedings of the ACM KDD'99 Conference, San Diego, CA. pp. 201-212.

[11] Andreas Geyer-Schulz and Michael Hahsler " Comparing two Recommender Algorithms with Help of Recommendations by Peers.

[12] Gleb Beliakov, Tomasa Calvo and Simon James "Aggregation of preferences in recommender systems".

[13] Manisha Hiralall "Recommender systems for e-shops" Vrije university, Amsterdam.

[14] Ayhan Demiriz "Enhancing Product Recommender Systems on Sparse Binary Data".

[15] Debajyoti Mukhopadhyay, Ruma Dutta, Anirban Kundu and Rana Dattagupta "A Product Recommendation System using Vector Space Model and Association Rule".

[16] Robin Burke "Intergrating Knowledge-based and Collaborative-filtering Recommender Systems" Recommender.com, Inc. and Information and Computer Science, University of California, Irvine, CA 92697.

[17] "Non-Personalized Recommender Systems with Pandas and Python" by Marcel Caraciolo, Artificial Intelligence in Motion, A blog about scientific python, Data, Machine Learning and Recommender systems(aimotion.blogspot.in).

[18] Ben Schafer, Dan Frankowski, Jon Herlocker and Shilad Sen "Collaborative Filtering Recommender Systems".