Collaborative Filtering Based Recommender System With User Data

A dissertation submitted in the partial fulfillment for the award of Degree of

Master of Technology

In

Software Engineering

By

Sumit Jain (Roll No. 2K15/SWE/18)

Under the Guidance of

Dr. Rajni Jindal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

2015-2017

CERTIFICATE

This is to certify that the thesis entitled **"Collaborative Filtering Based Recommender System With User Data"** submitted by **Sumit Jain** in partial fulfillment of the requirements for the award of degree Master of Technology in Software Engineering, is an authentic work carried out by him under my guidance. The content embodied in this thesis has not been submitted by him earlier to any institution or organization for any degree or diploma to the best of my knowledge and belief.

Date:

Dr. Rajni Jindal
Head of the Department
Department of Computer
Science and Engineering
Delhi Technological University

DECLARATION

I hereby declare that the thesis entitled **"Collaborative Filtering Based Recommender System With User Data"** which is being submitted to Delhi Technological University, in partial fulfillment of requirements for the award of degree of Master of Technology (Software Engineering) is an authentic work carried out by me. The material contained in the report has not been submitted to any university or institution for the award of any degree.

Sumit Jain
2K15/SWE/18

ACKNOWLEDGEMENT

I would like to take this opportunity to express my appreciation and gratitude to all those who have helped me directly or indirectly towards the successful completion of this work.

Firstly, I would like to express my sincere gratitude to my guide **Dr. Rajni Jindal**, **Head of the Department, Department of Computer Engineering**, **Delhi Technological University, Delhi** whose benevolent guidance, encouragement, constant support and valuable inputs were always there for me throughout the course of my work. Without her continuous support and interest, this thesis would not have been the same as presented here.

Also I would like to extend my thanks to the entire staff in the Department of Software Engineering, DTU for their help during my course of work.

Last but not the least I would like to express my sincere gratitude to my parents and friends for constantly encouraging me during the completion of work.

Sumit Jain

2K15/SWE/18

ABSTRACT

Consumers currently enjoy a surplus of goods (books, videos, music, or other items) available to purchase. While this surplus often allows a consumer to find a product tailored to their preferences or needs, the volume of items available may require considerable time or effort on the part of the user to find the most relevant item. Recommendation systems have become a common part of many online business that supply users books, videos, music, or other items to consumers. These systems attempt to provide assistance to consumers in finding the items that fit their preferences. This report presents an overview of recommendation systems. The classical methods for collaborative recommendation systems are reviewed and implemented, and an examination is performed contrasting the performance among the various models. Collaborative filtering is one of the well known and most extensive techniques in recommendation system its basic idea is to predict which items a user would be interested in based on their preferences. Recommendation systems using collaborative filtering are able to provide an accurate prediction when enough data is provided, because this technique is based on the user's preference. User-based collaborative filtering finds the similarities among the users and predict the unknown rating based on weighted average of similarity score of ratings of similar user. Item-based collaborative filtering works on grouping of similar items and give recommendation based on these groups. Hybrid based collaborative filtering combine the rating prediction process of above two methods. Mean Absolute Error (MAE) metric is used to evaluate these techniques of collaborative filtering.

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Recommendation Systems

All time we need to deal with options and choices. What cloth to have? What film to watch? What share to purchase? What to study? This decision domain has big size: Netflix's selection contains around 18,000 movies, Amazon.com has list of around 510,000 e-book's titles in the Kindle store . To get relevant information in this enormous space is big challenge. Even easy decision like what film should I watch this coming Sunday ? , may be difficult with no prior knowledge of the options. Generally, user depends on recommendation and option from their friends or the advice given by experts to take decision and find new things. They question with sale person over the air conditioner , they study reviews printed in newspaper's entertainment portion, or they take suggestion of book from librarian. They may trust on their local theater manager or news stand to select their options, or turn on television and see whatever happens to be playing. But these methods have their limitation of recommending new stuff, particularly for knowledge finding. There may be chances of an independent film or novel that a user will like, but no one in his friend circle has listen of it. There may be new music band in one state whose music will never cross the local region. Computer-aided technologies give the opportunity to expand the small group of people from whom users get suggestions. They can get users past history and preferences that they and their peers can not identify, potentially giving more fine selection result.

Over last 10 years ,good research work has been done to automatically suggest items to people and for this various type of methods have been given. In recent times, Recommender Systems Handbook was presented, providing deep discussion on various of recommendation technique and areas. This report, however, is mainly focused on collaborative filtering based Recommender Systems , a class of technique that suggest items to people based on the ratings given by other one for those items.

Interest is growing in problems related to recommendation. Techniques for learning and predicting user rating are only one part of a broader user experience. In recommender system ,it has interaction with user, both to understand the user's preferences and give suggestions; these concerns has challenges for user interface and interaction plan. Systems require correct data to figure out recommendations and preferences, this lead to task on how to gather reliable data and lessen the noise in user ratings data . People may have many different aims and wants when they use systems, from basic needs of information to complex requirements of privacy of their preferences. E-commerce field is showing increase in the demand for personalized services,that's why recommendation systems are evolving as an important business application. Amazon.com, for example, gives personalized item recommendations on the basis of prior bought items. Other examples , film recommendations in Netflix, song recommendations in Pandora and friend suggestions in Facebook. Any software applicaton which suggests an product to buy, to subscribe, or to spend can be considered as a recommender system. In this broad way, an advertisement can also be considered as recommendation. In this thesis, we mainly study a narrower definition of *personalized* recommendation system that is based on recommendations using user specific data. There are mainly two approaches for personalized recommendation systems: content based filtering and collaborative based filtering. The former one uses domain information users and items. The domain information may correspond to user data such as age, gender, occupation, or location, or to item data such as genre, producer, or length in case of movie recommendation system. The latter one, collaborative filtering (CF) does not use user and item data , while exploit the partially filled rating matrix. The rating matrix contains ratings of items (columns) by users (rows) for example, one to five stars as in Netflix movie recommendation system .The rating matrix can also be filled with user activity such as click through during a web search, in which chosen hyper-link may be interpreted as a positive value. In general, the rating matrix is sparse means it is partially filled and some empty region in matrix, since it is not possible that each user experienced and provided ratings for all product.

We can also classify recommender systems according to their main goals. Recommending good items may be the most important goal in many recommender systems. Amazon.com, for example, tries to suggest items which are potentially attractive to particular users. Another goal is optimizing utility, the profit to the company for instance. This can be seen as a slight

modification to the first goal, as a weighted sum of recommended items. Lastly, predicting unseen ratings on an item by a user is also a popular use of recommender systems. Netflix, for example, estimates how many stars will be given by a user to a movie. Based on this prediction, it can recommend movies to those who gave high scores. This thesis, among the various recommender systems, focuses on collaborative filtering (CF), mainly for rating prediction. As collaborative filtering algorithms work mainly on rating matrix which is similar across different domains, the results and conclusions in this thesis can be used to any CF-based recommender systems independent of the domain.

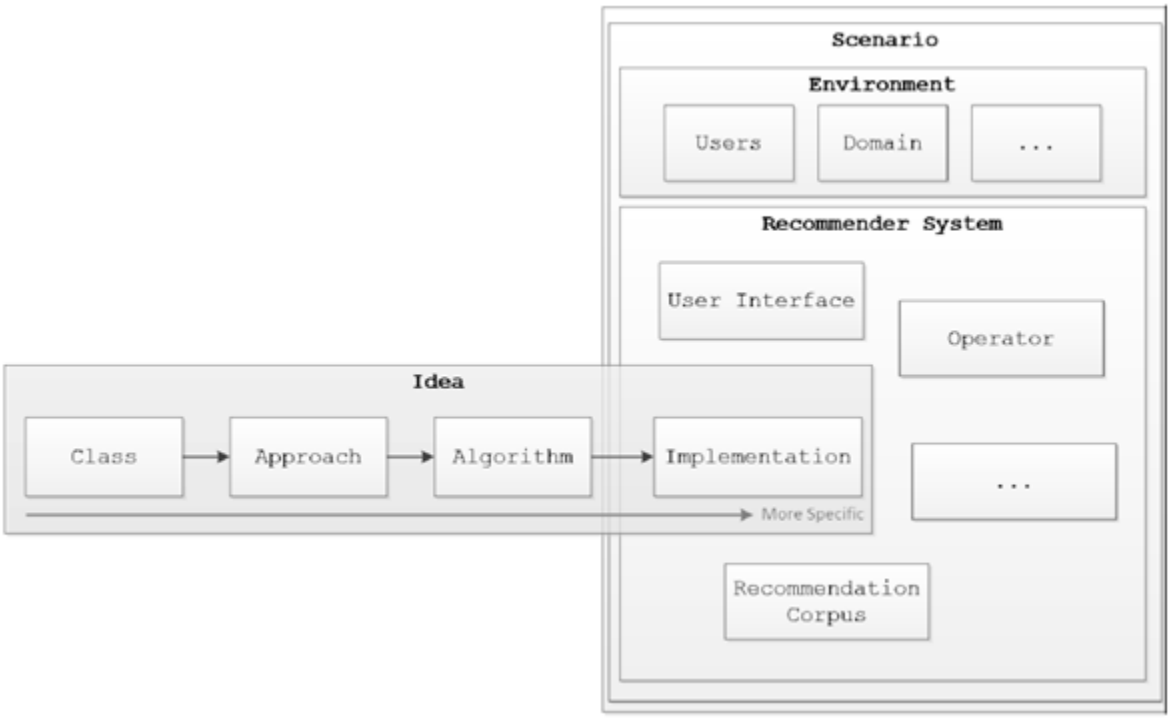


Figure 1 Recommendation system concept

A "recommender system" is a fully functional software application that define at least one implementation to make suggestions. In addition to this, recommender systems contains several other components , like user interface, recommendation corpus and operator that runs the application. Some recommender systems may use two or more recommendation techniques.

1.1 History of Recommender Systems

The ability of computers to give recommendations was found early in computing history. There was an early step in the field of automatic recommender systems that is Grundy , a computer based librarian. It groups users into stereotypes through short interview and use hard-coded data about different stereotypes ,book preferences to give recommendations, but it shows an early important entry in recommender systems field. In the early 1995, A solution to deal with overload in online data space is given using collaborative filtering .A manual based collaborative filtering system called Tapestry was developed : it facilitates the user to ask for product in an information space, such as e-mail, based on other users' preferences or actions (provide all messages send by Jack). It needs effort from users side, but facilitates them to gather the reactions of prior readers of a part of correspondence to know its relevance . Automated collaborative filtering systems come soon, automatically understanding relevant preferences and summing them to give recommendations . GroupLens also try to use this method to find Usenet articles which are to be in interest of a particular user. Users only require to give ratings or do other noticeable actions; the system summed these with ratings or actions of another users to give personalized recommendations . In these systems, users do not get any direct information of another users' preferences, and they do not require to know what other users or products are in the application to get recommendations.

At that time, recommender systems and collaborative filtering was an topic of growing interest among people (information retrieval researchers ,computer interaction and machine learning). This interest has given a number of recommender systems in different domains, such as Ringo for songs, the BellCore Video Recommender for films, and Jester for jokes. Outside the computer science, Marketers has understood the recommendation for its capability to grow sales and get better customer experience.

Amazon.com is the well known application of recommender system technologies. Based on browsing history ,buying history, , and the product a user is currently searching, they suggest product for the user to make them purchase. After adoption of Amazon, recommender system,

based on collaborative filtering, has been integrated into many e-commerce and online systems. A major motivation behind doing this is to grow product sale. Customers can buy an product if it is recommended to them but may not find it out otherwise. Some companies, such as NetPerceptions and Strands, have done work on recommendation system and services to e-commerce. The recommendation techniques has gone beyond collaborative filtering to include content-based technique , Bayesian inference , case-based reasoning strategy. These methods harness the knowledge of actual content or attributes of the product to be suggested and user rating pattern. Hybrid recommender systems have been emerged as different recommendation technique, likely to be matured.,It actually combine multiple algorithms into single system that works on the strengths of their component algorithms. Collaborative filtering, has become an efficient method, both as single one and hybridized with content-based method. Research work on recommendation algorithms gathered significant attention in 2008 when Netflix announced the Netflix Prize to make the movie recommendation more efficient. The main purpose of this activity was to make a recommendation algorithms that could excel their internal CineMatch algorithm in online tests by 10%. It sparked people, both in academia and amongst hobbyists. The $1M prize was decided for the vendors who place on efficient recommendations.

1.2 Motivation

Today, it is becoming a challenging problem to select the correct item to buy or use because of increasing . number of items available. Some ecommerce site like amazon has listed more than 10 millions product on web. While growth in choice size gives more opportunities to buyer to have the products fulfilling his personal requirement, it might in the meantime overwhelm him with excessively numerous choices. Recommender Systems (RS) handle this issue giving personalized recommendations for products or services ,digital content on web , that fulfill the user's requirements and tackle constraints better than mainstream products.

Now adays, using recommendations from others by words, letters, reference, travel guides and media reports are common practices. Recommender systems can improve this type of activity by helping people to search or explore for available items, such as, movies , restaurants ,books, articles, music , web pages etc. Recommender systems give suggestions to users of the products that are considered to be based on the users' preferences . That's why Recommender system has

become an essential application for e-commerce and information retrieval, helping users to search those products that are more appropriate for users needs and tastes by decreasing the large space of choices .

1.3 Problem Statement

Recommendation system are used in various applications and have tried to give users correct recommendation to fulfill the user needs and to make benefits to companies. Collaborative filtering is well known and effective technology in recommendation systems. The goal of these techniques is to predict the ratings for the items that user has not rated and to achieve this goal, various similarity measure is used like Pearson correlation, cosine-based ,Euclidean distance and Tanimoto coefficient etc .These similarity measures are used in both user based and item based collaborative filtering .In user based similarity between users is assessed while in item based similarity between items is assessed and we compare all these methods .Using both item based and user based method ,hybrid based method is introduced which try to make recommendation better.

1.4 Organization of thesis

Rest of our work can be summarized as below:-

Chapter 2 This chapter presents the literature review of the recommender system

Chapter 3 This chapter gives the detail of collaborative filtering recommender system and various techniques used .

Chapter 4 This chapter explains the User based collaborative filtering and algorithm.

Chapter 5 This chapter illustrates about Item based collaborative filtering and its algorithm

Chapter 6 Hybrid based collaborative filtering is explained here and defined algorithm.

Chapter 7 Shows the result and analyze the data.

Chapter 8 Concludes our work and explains about its future work.

CHAPTER 2

LITERATURE REVIEW

Recommender systems are software that use data filter techniques and algorithms to provide personalized suggestion with an aim of aiding user in taking decision process. Recommender frameworks has different use in application areas i.e. online books buying, web based shopping, online inn appointments, music and movie suggestions etc. The concept of recommendation is not new but in use from many years, the difference is due to more number of users asking for recommendations among thousands to millions of choices. It has turned into a dull task to prescribe somebody properly without separating the information for pertinent decisions. It relies on a few factors like user s give rating to collection of items based on their liking level, their preferences, gender, occupation, age, region or locality, group *etc*.

Several prevalent sites that are utilizing recommendation engine to filter options are listed below

Amazon, the prevalent e-commerce website, utilizes content-based recommendation. When you select a thing to buy, Amazon suggests other product other users bought based on that original product (as a matrix of product-to-chance-of-next-product buying).

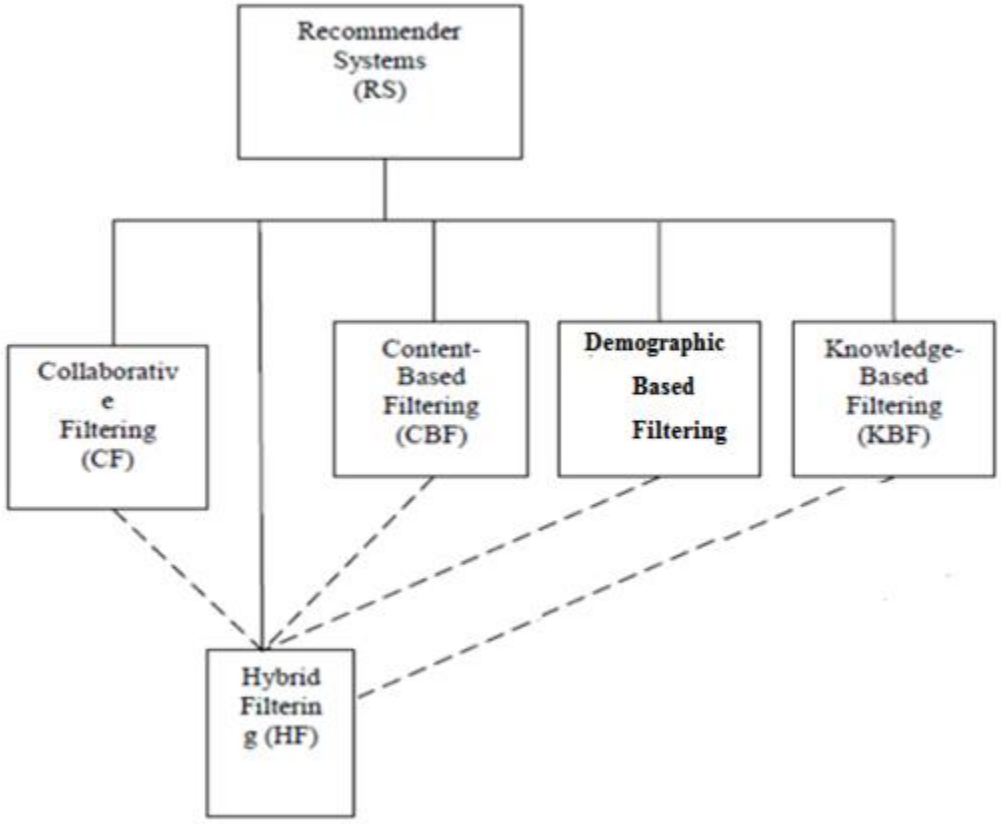
Hulu, a streaming-video site, make use of recommendation technique to recognize content that may be of users' interest.

Netflix, provide facility for video rental and streaming , is good example.

Other websites that uses recommendation engines are Google, Facebook, Twitter, Pandora, MySpace, Last.fm, Goodreads, Del.icio.us, and online news website. Considering the current situation, one can state that utilization of a recommendation engine is getting to be a standard part of a modern web presence .Recommender systems aid in matching users with product and for this task different recommender systems have been outlined by accessibility of exploitable information, feedback given by user, domain properties *etc*. Recommender Systems are classified based on approach or paradigm used to predict choices.

These are classified into five types given below:

- Content-Based Filtering (CBF)
- Collaborative Filtering (CF)
- Demographic-Based Filtering (DBF)
- Knowledge-Based Filtering (KBF)
- Hybrid Filtering (HF)



2.1 Content Based Filtering

Content-based filtering works on information about item and knowledge of user's preference. In content-based recommendation system, keywords are utilized to depict items; adding to it, user profile is made to show type of item which this user will like. In another ways, this algorithm attempt to prescribe items that are like to one that a user liked previously (or is looking at present)[13]. Specifically, many candidate items are matched with items rated in past by user and the best-similar items are suggested

Figure 2 Content based filtering

2.1.1 Stages in CBF

There are three stages of this recommendation procedure.

- Content analyzer: its main task is to present the content of items. it capable in extracting the data or main feature from item using feature extraction methods.

- Profile learner: In this process ,it gather data from the users preferences and attempt to sum up the information, at that point develop the user profile.

- Filtering components: In this procedure ,it attempt to compare the user profile's features to items' features. And afterward, the framework will prescribe things that fit for the user.

Here is a a flow chart of the procedure caught from Recommender System Handbook.

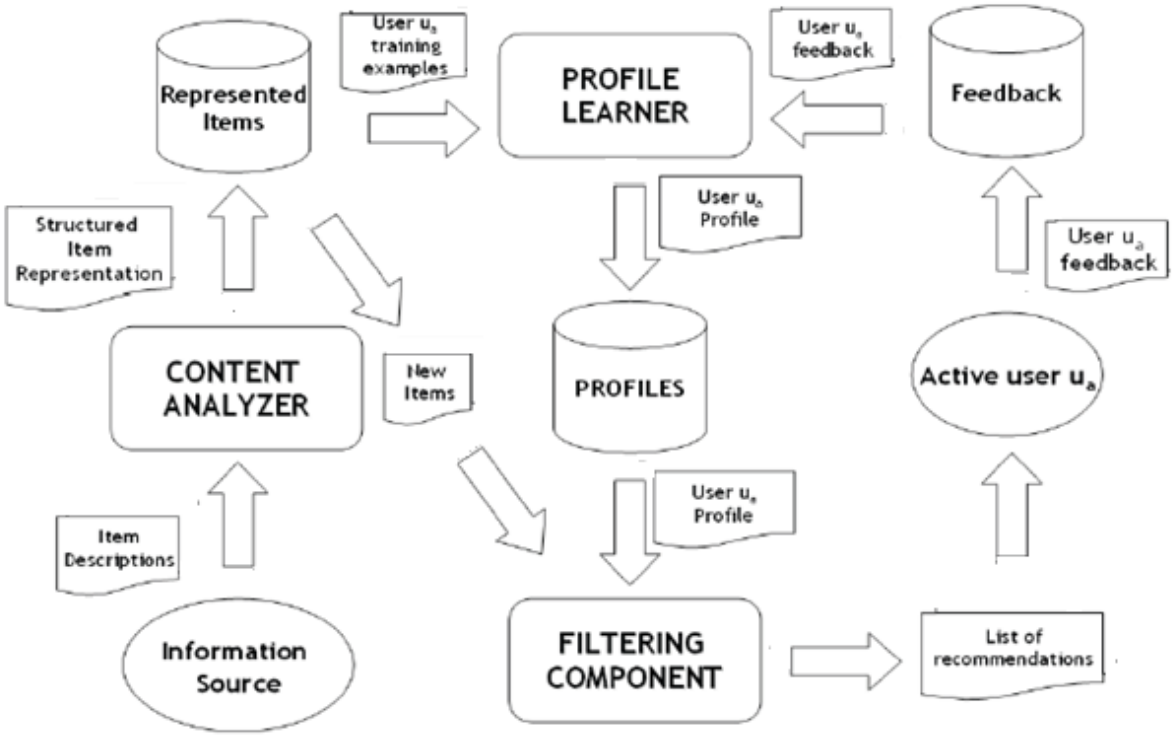


Figure 3 Content based filtering process

2.1.2 Advantages

User independence: collaborative filtering find the similarity among the users by using users' rating and afterward give the options to user to choose item. Rather, content-based method technique just need to investigate the user profile and items for giving suggestions.

No cold start: as opposite of collaborative filtering, it can suggest new items before considerable number of users rate them.

Transparency: collaborative method gives suggestions since unknown users have the similar taste as like you, but content-based method can reveal to you they suggestions the items on the basis of what features they have.

2.1.3 Disadvantages

Constrained content analysis : It will be difficult to recommend precisely if the content of item does not have proper and enough information to differentiate the items correctly.

Over-specialization: In content-based method ,it gives limit degree of novelty, because it need to compare profile feature and items feature. A fully perfect content-based filtering may not suggest surprised item.

New user: when there's insufficient data to assemble a good profile for a user, the recommendation couldn't be given accurately.

2.2 Demographic Based Approach

A recommender system based on demographic, suggests items to user with the help of user's demographic data which include age , birthday date and gender. In this demographic approach ,it makes the group of users on the basis of their demographic characteristics. For instance, the framework will keep the users into one group ,who belong to certain zip code. Likewise, the users will be in one group if they are of ages ranging from 20 to 30 years. The recommendation method working on demographic approaches accepts that users in the same category or group have similar preferences and interests. The demographic system keeps eye on users' purchase and rating behavior within the similar category or group. In case any new user start using the system, the system firstly will find the matching group based on the user's demographic information and then user into that group. At that point, the framework will prescribe items to user on basis of purchasing behavior or rating given by users in the similar group. An example of demographic information based recommendation system is Grundy . The motivation behind the framework is to prescribe books to library guests in view of their own data that is accumulated from them through an intelligent conversation . Other current example of a demographic data based recommendation system is LIFESTYLE FINDER. The system utilizes demographic groups data from marketing research field to suggest a various product and services, and it collects data from users by conducting short survey .

The main advantage of the demographic based system is that this system does not need to maintain any history of ratings given by user as in the case of collaborative filtering and content based method .But it has some drawback in demographic-based approach. The first drawback is that demographic system faces problem of how to find the category or group ,to which the user belongs in case of the user is new for the system. The second drawback is to find the preferences and interests of users in the similar category. The third drawback of this demographic method is that it do well when system has the demographic data available. But, it is difficult to collect this type of data. Accordingly, few recommendation systems utilize the demographic approach due to its drawback. In addition to this, the correctness of demographic

data based recommendation systems is less than content or collaboration filtering based recommendation systems

2.3 Collaborative based Filtering

Collaborative filtering based recommender system collects and analyze big size information related to users' activities, behaviors, and preferences , to anticipate what users may like on the basis of similarity from other users. A key preferred standpoint of the collaborative filtering method is that it doesn't depend on content analyzable using machine and hence it is able to do precisely prescribing complex things, for example, films without requiring a "comprehension" of the thing itself. Numerous algorithms have been utilized as a part of measuring item similarity or user similarity in recommender systems. For instance, the Pearson Correlation ,cosine value and k-nearest neighbor (k-NN) approach.

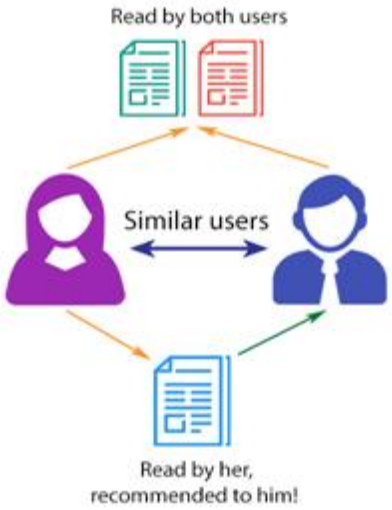


Figure 4 Collaborative based filtering

Collaborative filtering based recommender system has some advantage that is ,it does not need knowledge about item features .For example ,in movie recommender system it can suggest movie without knowing the internal knowledge of movie .But it has some limitations also that is It has problems of new user who has not given any rating to any item .Another is new item for which no user has given any rating . Sparsity problem is also considerable. In this case if number size of item is large then user-rating matrix may be sparse and it is difficult to get the users

who rated same item .This system is popularity bias that make recommendation to only popular items. Following image depicting collaborative and content based filtering.

2.4 Knowledge- Based Filtering

Knowledge based filtering recommender systems utilizes the structure knowledge to give choices to user and also make preferences. In this type of recommender systems ,they keep knowledge of type of items that are liked by user, so that, a connection is created between user requirements and accurate suggestion to that user. These systems are based on a specific domain and if an item fits in that domain, then items would be recommended to a user [15]. The similarity measure is based on how much a user demand matches the generated predictions. The knowledge base could be implemented by collecting requirements of user on particular product and that was asked to the user. On consulting the knowledge base, the required products are recommended. Knowledge-based systems attempt to learn rules and then use logic to make their recommendations. These systems work best in situations where ratings are sparse, due to the low frequency of their occurrence like house or car purchases, or where requirements need to be more precisely specified. Burke describes these systems as more of a conversational system as opposed to information filtering. There are basically two types of knowledge-based systems: one is constraint-based, that work by satisfying rules, and second is case-based,similarity metrics, system. The first might apply to home purchases. A prospective buyer specifies a price range and the systems works to provide them with available houses within that range. This type has a greater similarity to query-type systems than any other type of recommender systems.The second could be used in a local food finder that attempts to find nearby restaurants with food similar to other restaurants that the user has rated highly.

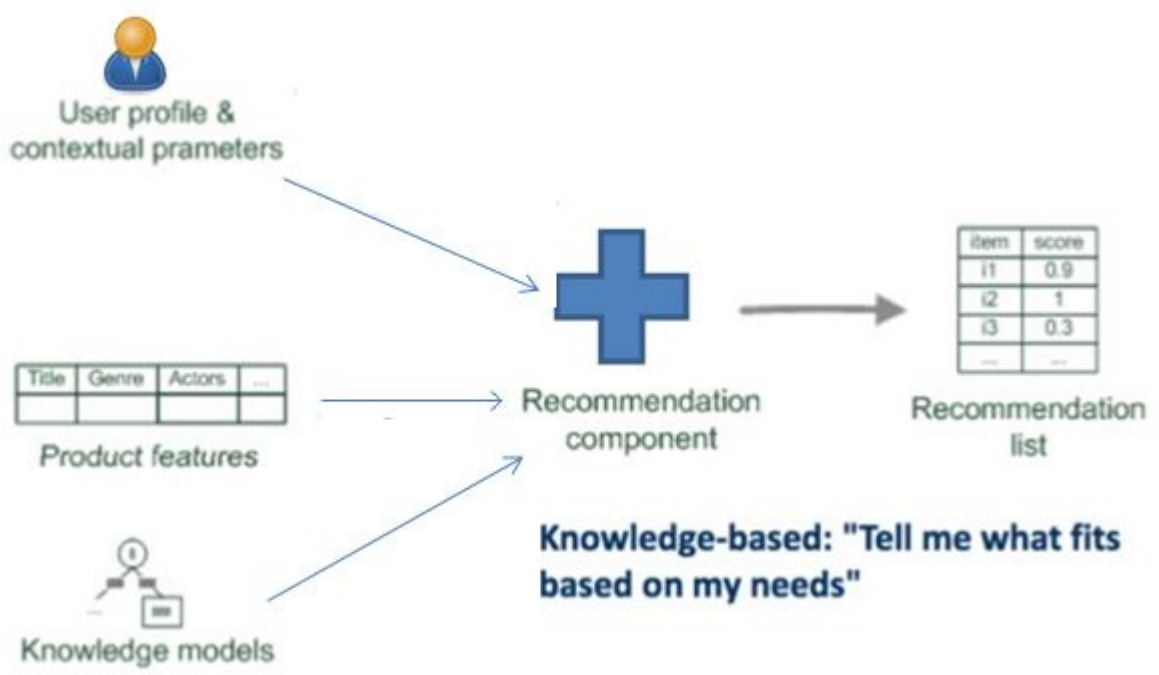


Figure 5 Knowledge based filtering

2.5 Hybrid Based Recommendation Systems

In this type of recommender system two or more approaches combined together , keeping in mind the handling of specific limitations of a single approach to alleviate ,for example, user-user based collaborative filtering and item-item based collaborative filtering combined together to give better result for recommendation .By this technique sparse data of user item rating can be handled.

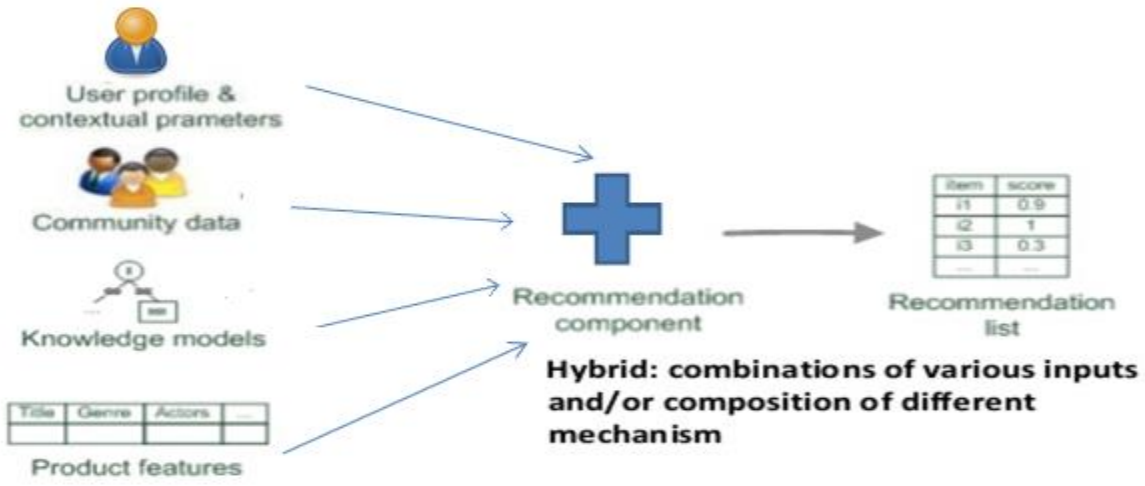


Figure 6 Hybrid based recommendation system

there exist seven hybridization techniques that are explained briefly below:-

Hybridization method	Description
Weighted	The scores (or votes) of several recommendation techniques are combined together to produce a single recommendation.
Switching	The system switches between recommendation techniques depending on the current situation.
Mixed	Recommendations from several different recommenders are presented at the same time
Feature combination	Features from different recommendation data sources are thrown together into a single recommendation algorithm.
Cascade	One recommender refines the recommendations given by another.
Feature augmentation	Output from one technique is used as an input feature to another.
Meta-level	The model learned by one recommender is used as input to another.

Table 1 Hybridization method

Weighted:- It is the most simple architecture of hybrid system. In this approach , required items are scored independently by both fused recommender system, though the final result is calculated by linear summation of intermediate values. Here , relevant weights for each part is determined by practical means .Content-based recommenders can predict any item, but collaborative recommender can score item only if it is rated by peers.

Mixed:- In numerous area it is not feasible to get score of item by using both of the recommenders method in hybridization, reason is that content space or rating matrix may be too

sparse. Mixed hybridization techniques create set of recommendations for each component of hybrid technique independently, and combine the sorted candidates by their rank before presenting to user. But combining the anticipated items from both of recommenders creates it difficultly in evaluation of the improvement of the single segment.

Switching: In some cases, more than two recommendation methods related content filtering and collaborative filtering technique are comprised to develop hybrid systems . In switching case,we sort recommenders, then if first one will not be able to give a recommendation with high value of confidence, then we move to next one and tried it, and go on. Unexpectedly, another switching hybrids may choose single recommenders depending to the kind of user of item. although, this method considered that there is some relevant switching reason used.

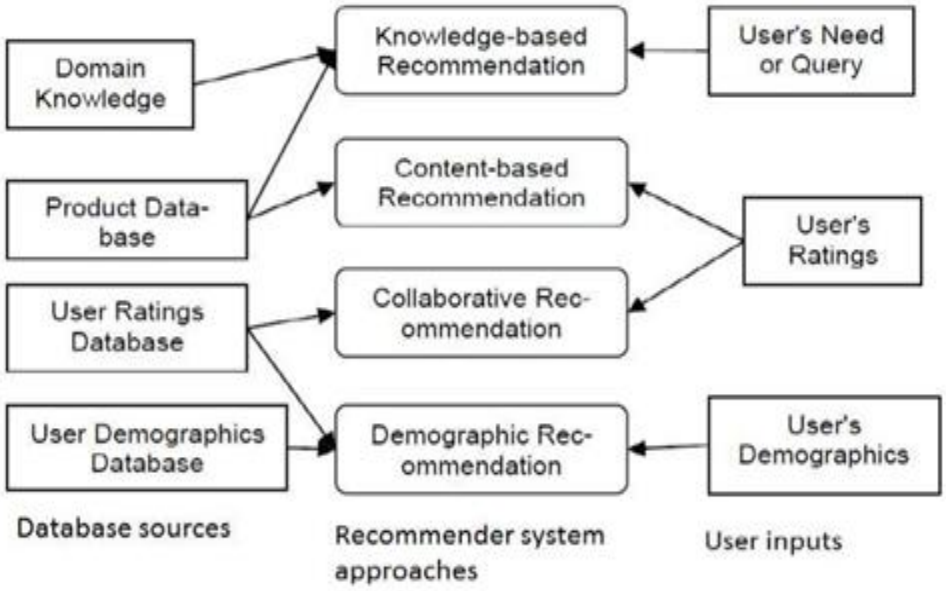
Feature Combination: In feature combination approach ,it works with one recommendation module, which is aided by second module passively. Rather than manipulating the features of the contributing module independently, they are infused into the actual recommender algorithm.

Feature Augmentation :- The feature augmentation strategy is similar to feature combination in some ways. Yet , rather than utilizing the contributing domain's raw features, feature augmentation aid their main recommender by features passed to contributing recommender. Normally, feature augmentation is used when there is well designed primary component which need extra sources of knowledge. The fact that most of applications do expectation to recommendations in real time situation and usually augmentation works offline. All in all,, is better than feature combination, because feature augmentation add a fewer features to primary recommender module.A feature argumentation recommender example is, fusion of content-based and collaborative methods to anticipate new items of user interest to the user. Its Content-Boosted Collaborative Filtering (CBCF) algorithm learns built a content-based model on training data to give unknown ratings of items. The collaborative recommendation as part of by the actual recommender uses the produced dense rating. CBCF beats weaknesses of both CF and CBF methods, and altogether enhanced the anticipation of the recommender framework.

Cascade: The cascade hybrids concept is little bit similar to feature augmentation. In this method, the cascade model' primary recommender create candidate selection , and allow the secondary one to refine result. For example, the main component may give same score to items that may be ranked again using the secondary component.

Meta-Level: This sort of hybrids utilize model learned through contributing recommender giving input to actual one. In spite of the fact that the meta-level hybrids general schematic looks like feature augmentation but there is a big difference in both one. Rather than providing actual recommender with extra features, a meta-level contributing recommender give a fully new domain of recommendation. Notwithstanding it is not generally fundamentally achievable to create a model that fits the recommendation logic of primary module.

All these recommendation techniques require different type of knowledge of domain that is shown below.



2.6 Trade off between recommendation approaches

This table depict the advantages and limitation for different recommender approach:-

Approach	Advantages	Limitations
Collaborative Filtering Approach	- No domain Knowledge required - Quality of recommendation increases over time - It can identify cross genre niches - Implicit feedback is sufficient.	- New-user problem(cold start problem) - Gray sheep problem - Scalability problem - New-item problem(First-rate problem)
Content-based Filtering Approach	- No domain knowledge required - New item recommendation - Quality of recommendations increases over time - Implicit feedback is sufficient.	- New- user problem(Cold start problem) - Limited content analysis problem - Over-specialization - Scalability problem
Knowledge-based Filtering Approach	- No cold start problem - No over- specialization problem - Prone to preference changes - No scalability problem	- Needs domain knowledge - Does not learn over time

Table 2 Trade off among recommendation techniques

2.7 Desired Characteristics of Recommender System

Recommender system is application software that create relevant purchase choices for the customers. Although the idea of relevance and effectiveness of suggestions is subjective and pecific to the domain under consideration and the actual requirements of the business model, we briefly discuss some of the general properties of a good RS.

Quality of Prediction: -

The key requirements of a good recommender system are to generate meaningful recommendations so that a customer's interest in the portal is maintained. This need translates into generating recommendations that are aligned with a user's preference. Deciphering a user's preference from the extremely limited (available) information is a major challenge for RS design. However, the accuracy of recommendation from a user's perspective is not the sole measure of the effectiveness of a RS. A recommender system's capability to enhance the visibility of items, especially niche products is also advantageous from both the customer as well as retailer's perspective.

Speed of Computation: -

Most of the big online retailers like Amazon have more than a million registered customers and an equally large number of items. Thus, an effective recommender system should be equipped to handle this information overload and generate relevant suggestions in reasonable time. Further, continuously new ratings, users, and items are added to the portal, and a slow recommendation strategy will hinder frequent updates to accommodate the same. Thus, the speed of processing is a major criterion for any RS design algorithm.

Applicability of Design: -

The design of recommender systems that effectively use available information entails considerable effort. A generic recommendation framework which can work across domains and with limited information can enjoy wider applicability; this is an added benefit over target (portal) centric designs.

2.8 Recommendation System Challenges

Recommender systems have become very popular for giving recommendations by utilizing numerous recommendation algorithms in various application domains and this huge work brings notice towards many challenges. These issues are given below:

Data Sparsity:-

If the user-item matrix contains rating details empty at various places than it is called the problem of data sparsity and this circumstance additionally prompts wasteful recommender frameworks which depend on nearest-neighbor algorithms for manipulating similarity between either user or item. It has classification in two parts that is reduced coverage problem and neighbor transitivity problem .

Reduced coverage means the system is not capable to provide suggestions for items ,the reason behind this is the data sparsity which shows entries belonging to items rated by numerous users is not present..

Neighbor transitivity issue emerges when users have not given rating to items that are in similar set then it causes difficulty to manipulate similar users for predicting rating.

Scalability:-

This type of problem arises when ratings given by the user and size of items grow extremely and it causes difficulty to recommender system to cop up with such a huge data because of constrained resources and computational complexity. So, it reaches across the boundary of acceptability.

Synonymy:-

Sometime items are similar but have given different names to them ,in that case recommender system will not be able to identify that similarity between those item and recommends them as different items. This cause to issue of suggesting similar items, is referred as synonymy problem.

Gray sheep and black sheep problem:-

The recommender systems has two very well-known problems that is Gray sheep and black sheep problem. The gray sheep problem happens when the choice of user does not match with any other user or group of users in agreement or disagreement consistently and on the other hand black sheep problem refers to a situation when the choice of user matches with few number of users or no at all. In this situation , recommender system becomes helpless and not capable of giving preferences .

Shilling attacks:-

Shilling attacks are of two types, push attacks and nuke attacks**.** At the point when opposite seller follow the unfair means to present more rating to their own product in comparison to other seller products then in that case it is called push attacks. On contrary if seller reduce the rivals or competitors rating then in that case it is nuke attacks. It may happen that users give biased recommendations due to their own possessions ,negative thought for competitors products. This type of circumstances should be handled in CF model. Recommender systems utilized Shilling attack models and its effectiveness has been anticipated. It is showed that item based CF algorithm is less effected then user CF algorithm . A better way to handle shilling attacks will be to lessen the global effect while performing data normalization in neighborhood based CF. The effects residual can be used in selecting neighbors.

Cold-start problem:-

When recommender system is not capable to give prediction because of lack of ratings initially then it point to the cold-start problem. This sort of circumstance happens when new user enters in to system ,do not have any rating records available for recommendation system or also when new item comes in to the system and nobody has given rating for that item. So, it causes difficulty for recommender system to give suggestions for new user and so the recommender systems aim is not fulfilled .

Besides all these problems discussed above, recommender systems suffers from many more challenges like generating preferences in cross-domains, context-aware recommendations, constrained based recommendations and many more .

Other challenges:-

In addition to above challenges of recommender system there arises issue of maintaining privacy of users information as one has to use it for collaborative filtering based recommendation generation and most of the time users are not willing to give their personal information for such tasks .

Another challenge is to filter the data using appropriate instances of data after removing noise. This process involves instances selection techniques based upon various criteria, for example, in MovieLens database only those instances are selected for experimentation in which a user has rated at least one movie. Similarly, other selection techniques can be applied for selection of instances for further testing.

CHAPTER 3

COLLABORATIVE FILTERING RECOMMENDETION SYSTEM

Collaborative filtering is an idea that is related to crowd sourcing. The basic idea is the use of large numbers of users and ratings to find similar items and users that can assist in the creation of predictions. Similarity measures are functions for determining how much one items is like another given a vector of features that describes them. Common similarity measures include cosine similarity and distance functions .For increased performance, item-item similarity is often used in conjunction with caching due to the lower volatility of their similarity measures. Requires a fairly significant number of ratings before any level of accuracy is guaranteed, however the accuracy of the systems will increase over time as more ratings, users, and items enter the system. New users and items need a certain number of ratings (items more so than users) before accurate predictions can be made even if the rest of the systems has achieved a higher level of accuracy.Now that most of the bias has been eliminated from the ratings data, another approach to predicting ratings must be found if greater improvement is sought. One approach is to find similar users that have rated the item and use those ratings in our prediction. However, that raises the question of determining similarity between users.

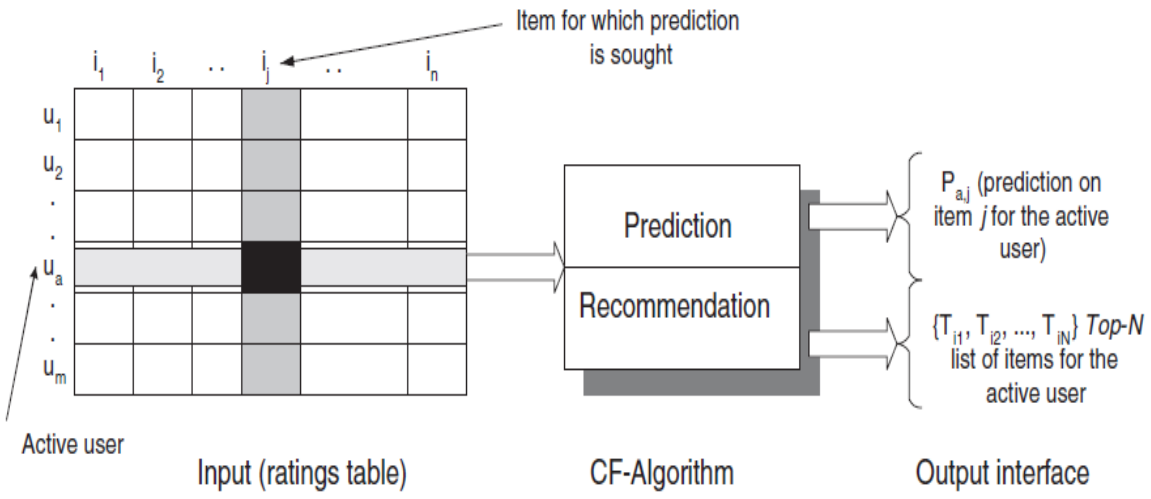


Figure 7 Prediction in collaborative filtering

3.1 Framework of Collaborative Filtering recommender system

The framework of Collaborative Filtering recommender system contains:

- Data Collection
- Pre-processing
- Collaborative Filtering.

First of all, wireless networks is used to collect the user data and saved in to the database . Afterword pre-processing operations are executed to ensure the data reliability and integrity. On the basis of these data, Collaborative Filtering algorithm using both User and Item is coded to give recommend items so that it can reduce effort and save time.

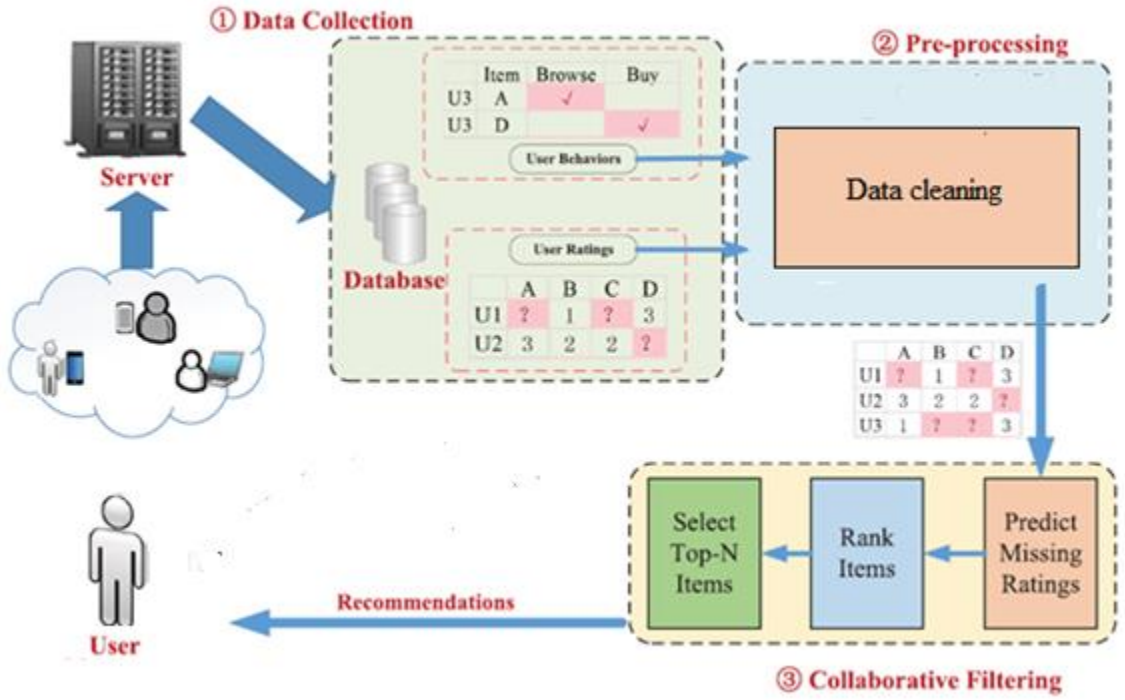


Figure 8 Framework of collaborative filtering

DATA COLLECTION:-

Data collection is the crucial part of the whole recommender framework. The assembled data comes in four classifications: production data ,demographic data, user rating , user behavior. Some sites give rating frameworks and help shoppers to rate things that they have encountered, for example films, music , web services .The consumer preferences is reflected by these ratings , get increasing attention through businesses. Besides, items can have different attributes which is required to rate respectively. Some systems used to rate ,give users chance to rate items in view of various criteria which can make rating data good . If we utilize all the data give above effectively than it will be important for the recommender system . while collaborative filtering does not require any information of the users and items , it concentrates on feedback given by user including user rating and user behavior.

PRE-PROCESSING:-

Due to advancement in mobile Internet technologies, the diverse user equipments and networks heterogeneity, has made various formats for the collected data. That why, data pre-processing is indispensable segment of recommender systems, which ensure thet the input data to the framework is reliable and complete. Presence of dirtiness in the data, raw data can not be used directly because transmission or equipment failure may produce error. When users are in high speed then error ratio may be high. And, in some case , some consumer rate item randomly , for example all items are rated the highest one value in order to save time, which reduces rating information reliability on whole. Special outlier detection algorithms may handle this type of problems to some level. For instance, training data is built by choosing piece of the ratings and a classifier model based on machine leaning algorithms is established, so that outliers are removed with accuracy.

COLLABORATIVE FILTERING:-

Collaborative filtering process include anticipating missing values, sort items with their rank and choosing highest N ranked items. we consolidate the upsides of the User-based collaborative filtering algorithm with Item-based collaborative filtering algorithm to recommend, and get improved results. Since the rating matrix has missing value , so collaborative filtering main task is to predict that missing values on the basis of known data. Then, Top-N items are recommended after ranking on prediction values.

3.2 User Ratings

Collaborative Filtering Recommender Systems depends on rating to recommend item to the user. Here preferences are of two type, that can be either explicit or implicit. In former one, user give value to item, and in later one, the system calculate that users whether like or dislike item by noticing their actions, for example screen clicks or seeing at purchases.

With regards to Collaborative Filtering, ratings are numerical value that show how much item will be liked or disliked by user, and are base for similarity measures to search same mind candidates.

Explicit vs Implicit:-

In explicit rating, user directly gives score to item, that is valuable because system need to say what the user will feel for the item. Nonetheless, psychology of human is very complex, and the taste of rating is not same at all time. A user may think high for given item, but decide to provide low rating due to some unknown reason like Viral trends, Social pressure.. Cool-factor. All that influences effect how will users rate the item, and also effect the recommendations .

Besides, the rating scale despite the fact that it is typically standardized, can be seen by various voters. There are two people who are liking movie a lot, may rate it in different way completely, the reason is one is more serious in doing movie rating, while second one regularly rates low value to keep inflation low.

The explicit rating has another problem of memory. Normally, users appraise the item at the moment when it user inspecting it for example book read just now or movie just watched. That type of items are fresh in memory, compared to another items used in past. This thing makes difficult for user to remain objective, and to give relevant rating .

Subsequently, "excellent" rating for a specific film viewed a month back could not confront to the opposition of new film seen now, however be remain equally rated as "excellent" .Since users generally do not change old one values, then both film will keep on the same score. If user remember old item then may be taste changes with time . If these rating is not corrected to show new preferences then these old values will affect recommendation task.

Implicit ratings attempt to limit the effect of psychology of human by not including user for appraisal work. In this case, Preferences are biased low because they do not depend on the user thinking .Normal way to assess the user taste is to examine how much time user is spending on particular item ,an algorithm may have difficulty to know whether user studying item with positive or negative attitude? And worse of this, if he leave the webpage open and gone to have coffee.

Inferring the taste can be cone using items purchased and view pages. The rationale thing behind is that purchased item is liked item, and also visited page is interested page. Normally,

one could purchase an thing for a companion, and wind up page by unimportant clicking link with mistake. sometimes, one or more person browses with same account, creating meshing of tastes, with low target.

The implicit ratings has advantage over explicit, that is user do not need to take pain to answer questions in training of algorithm. while, both one can be easily combined to add up the specific user's taste knowledge

Rating Scales:-

As opposed to Content Based Recommender System that mine the item content for similarities against other items content ,while in Collaborative Filtering, the similarities are users subjective preferences, so need some metric, to equate .Normally, there are three forms:

- Unary: Purchased or Good
- Binary: Like/Dislike
- Integer: 1 to N (example 1 to 5 stars)

The Unary scale is utilized to flag that user has has demonstrated enthusiasm for a specific thing. It may be implicit in the case of a purchase or a view page and explicit through button, for example Facebook "Like" or Google "+1 It informs nothing regarding the item user dislikes, but rather it is non-meddlesome, and by and, sufficiently large to find out about a user.

The system infer the unary scale, through the user's actions, rather than requesting input. While the client's is not expressively expressing an inclination, his inclination is as yet learned by method for the decisions he makes.

The Binary scale come under category of explicit, and works on two type of buttons, one for a positive and another for negative feeling. For example , in YouTube.com users press button on a "thumbs down" or "thumbs up". They give general emotions around an item , in any case they are great polarizer's with negligible meddling.

Last one, the Integer scale is like to a restaurant food rating or an hotel stars, and also come under category of explicit. It gives the biggest range to express taste, as it allows the user communicate himself with more direction. This type of rating is usually discovered for books or films.

3.3 Neighborhood selection

All the candidate users who are similar to target user ,are called Neighbor of target user ,can be used to recommend item, but , this become unreasonable due to large size user databases, and less correlation to target will introduce more error. To keep away from the consideration of users uncorrelated to target, in all algorithm a neighborhood selection step is introduced generally, to remove those unwanted candidates. Two principle techniques the K nearest - Neighbors, and Threshold Filtering.

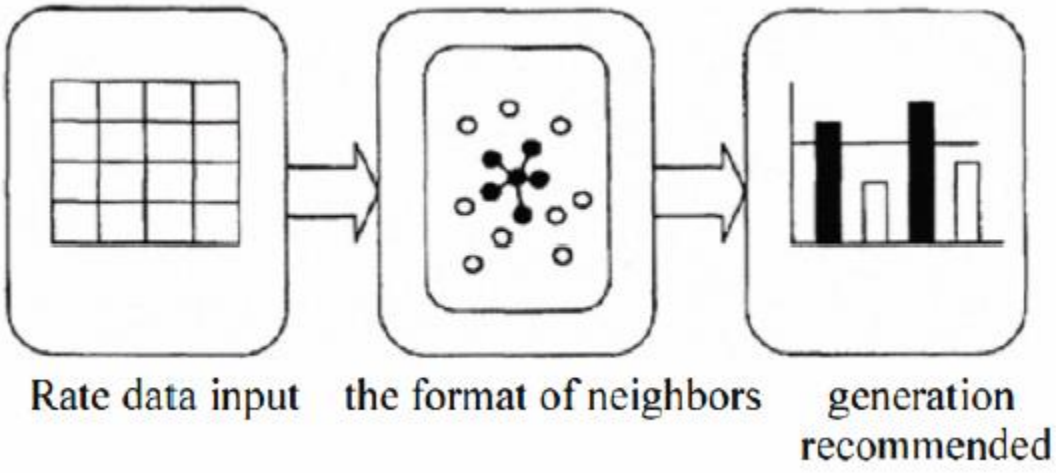


Figure 9 Neighbors formation in system

K Nearest Neighborhood-Based Rating Prediction

K Nearest Neighborhood method allow the selection of candidate user similar to target one ,In this method k most nearest neighbors were selected. Sometime users did not given rating to some items so common ratings between other user and active user is used by distance metric to calculate similarity .There are many focused similarity metrics for these algorithm. These are Pearson correlation, Cosine correlation, Tanimoto similarity and euclidean similarity. The Pearson correlation similarity metric is given below. Pearson correlation was used . in the GroupLens recommender system .

$$w_{au}^{P} = \frac{\sum_{\{y|r_y^a, r_y^u \neq \perp\}} \left(r_y^a - \bar{r}^a\right)\left(r_y^u - \bar{r}^u\right)}{\sqrt{\sum_{\{y|r_y^a, r_y^u \neq \perp\}} \left(r_y^a - \bar{r}^a\right)^2 \sum_{\{y|r_y^a, r_y^u \neq \perp\}} \left(r_y^u - \bar{r}^u\right)^2}}$$

Using the above equation n ,we calculate the similarity weights between users and among all the users most nearest neighbors were selected .These weights are used in combining the ratings of neighbors.

$$\hat{r}_y^a = \arg\max_{v \in V} \sum_{k=1}^{K} w_{au_k} \delta\left(v, r_y^{u_k}\right)$$

$$\hat{r}_y^a = \bar{r^a} + \frac{\sum_{k=1}^{K} w_{au_k}^{P} \left(r_y^{u_k} - \bar{r}^{u_k}\right)}{\sum_{k=1}^{K} \left|w_{au_k}^{P}\right|}$$

.

Algorithm KNN-Predict

Input: $\mathbf{r}^a$, $\mathbf{r}$, K
Output: $\hat{\mathbf{r}}^a$

for $(u = 1$ to $N)$ **do**
  $w_{au} \leftarrow \frac{\sum_{\{y|r_y^a, r_y^u \neq \perp\}} (r_y^a - \bar{r}^a)(r_y^u - \bar{r}^u)}{\sqrt{\sum_{\{y|r_y^a, r_y^u \neq \perp\}} (r_y^a - \bar{r}^a)^2 \sum_{\{y|r_y^a, r_y^u \neq \perp\}} (r_y^u - \bar{r}^u)^2}}$
end for
Sort w_{au}
for $k = 1$ to K **do**
  $u_k \leftarrow k^{th}$ closest neighbor to a
end for
for $y = 1$ to M **do**
  $\hat{r}_y^a \leftarrow \bar{r^a} + \frac{\sum_{k=1}^{K} w_{au_k} (r_y^{u_k} - \bar{r}^{u_k})}{\sum_{k=1}^{K} |w_{au_k}|}$
end for

Threshold Filtering

The Threshold Filtering method define the minimum value of similarity between the neighbors and target user to be chosen for calculation of final value rating. Using this type of strategy ,we can solve the problem of same size neighborhood at all time, but not able to avoid its own problems .If we set too high threshold value then results will have good correlated neighbors but in this case, some users may not easily correlated and result may have small size neighbors and have bad quality of recommendations. In another case, if we set the lower value of threshold then it will increase the size of the neighbors and fails for purpose of approach. In general, as threshold value is set low, it will drop the overall error in recommendation, but less number of users will be reliably recommended . Also, if the threshold value grow than it will cause increase in error in the recommendation, but more number of user will be recommended. This is trade off that should be considered carefully to each Recommender Systems .

Threshold based prediction

Input: User-Item Rating matrix R,threshold
Output: r rating of item not given
for each user *u* **do**
  Set Neighbour(u) to the v users similar to user u and having similarity greater than threshold
end for
for each user *v* in Neighbour(u) **do**
  Weighted add the rating of item of user v to r
end for
Rating of item for user u is r

3.4 SIMILARITY MEASURES

Recommender systems contain many similarity metrics that come from machine learning. They are important for recommender systems. Each similarity metrics are related with vector space methods; but there are various ways for defining the similarity. They can be categorized in a way that distance and degree measurement. There are different similarity calculation techniques for computing similarity between users. Since each similarity have different formulas, they give different measures from each other. Some similarity computation techniques are explained in the following sub headings.

In the Collaborative Filtering Systems, there is a mutual point that is establishment of similarity between users and items. Recommender Systems for the purpose of clarifying similar neighborhoods to the users or similarity computation between items. The similarity algorithms are:-

- Euclidean Distance Similarity
- Pearson Correlation Coefficient Similarity
- Tanimoto Coefficient Similarity
- Cosine vector Similarity

3.4.1 Euclidean Distance Similarity

Euclidean Distance Similarity method is working as users is a point in many items. The table has the rates of the each user to the each item. This metric converts Euclidean distance d between 2 such users. Distance value is smaller when these users are more similar. This method gives the value of *1/ (1+d)*.It never gives negative value as a similarity and when the value increases it means that they are more similar.

The equation is given in (1) as

$$r_2(x, y)=\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}=\sqrt{\sum_{i=1}^{2}(x_i - y_i)^2}$$

Table 3 Calculation of euclidean distance similarity

Euclidean	Item 1	Item 2	Item 3	Distance	Similarity to User 1
User 1	5.0	3.0	2.0	0.000	1.000
User 2	3.0	2.0	5.0	3.937	0.203
User 3	2.0	-	-	2.500	0.286
User 4	5.0	-	3.0	0.500	0.667
User 5	4.0	3.0	2.0	1.118	0.472

This method compares rates of the items for one item not for one user to items. Item similarity gives better results because user based similarity affected by mood of user or tastes of user can

change over time. Item similarities are more fixed and better for precomputation. It speeds up computation as runtime

3.4.2 Pearson Correlation Similarity

It is used for converting similarity value between two users or items by measuring obliquity of two preferences series to act collectively in a comparative and linear way. It considers preferences of conflicting users and items. It tries to get each users' or items' derivations from their average rates while recognizing linear adjustment between two items or users.

$$P,\ C(w,u)=\frac{\sum_i (r_{w,i}-\overline{r}_w)(r_{u,i}-\overline{r}_u)}{\sqrt{\sum_i (r_{w,i}-\overline{r}_w)^2 \sum_i (r_{u,i}-\overline{r}_u)^2}}$$

w and u may be the two users or items ,for them the coefficient similarity is calculated, i is an item, $r_{w,i}$ and $r_{u,i}$ are individual ratings from w and u for i, and average ratings of r_w and r_u are ,for user (or item) w and u [3, 4]. Table 5 shows Pearson Correlation Similarity of user1 and the others based on three items common.

Table 4 Pearson correlation similarity

	Item1	Item2	Item3	Correlation with user1
User 1	5.0	3.0	2.0	1.000
User 2	2.0	2.0	5.0	-0.764
User 3	2.0	-	-	-
User 4	5.0	-	3.0	1.000
User 5	4.0	3.0	2.0	0.945

3.4.3 Tanimoto Coefficient Similarity

This is a similarity that ignores the preference values so that it does focus on the value that the user given for the item. It only checks that the user expressed a preference or not. It is also named as Jaccard coefficient. Its formula is the number of items that both users showed their

interest, divided by the number of items that either user shows some interest. When they do not have any similar preference, the result will be zero. The similarity value cannot be greater than one. The equation for Tanimoto Coefficient Similarity is given below:-

$$T(A,B) = \frac{A.B}{|A|^2+|B|^2-A.B} = \frac{\sum_{i=1}^{n} A_i x B_i}{\sum_{i=1}^{n} A_i^2+\sum_{i=1}^{n} B_i^2-\sum_{i=1}^{n} A_i x B_i}$$

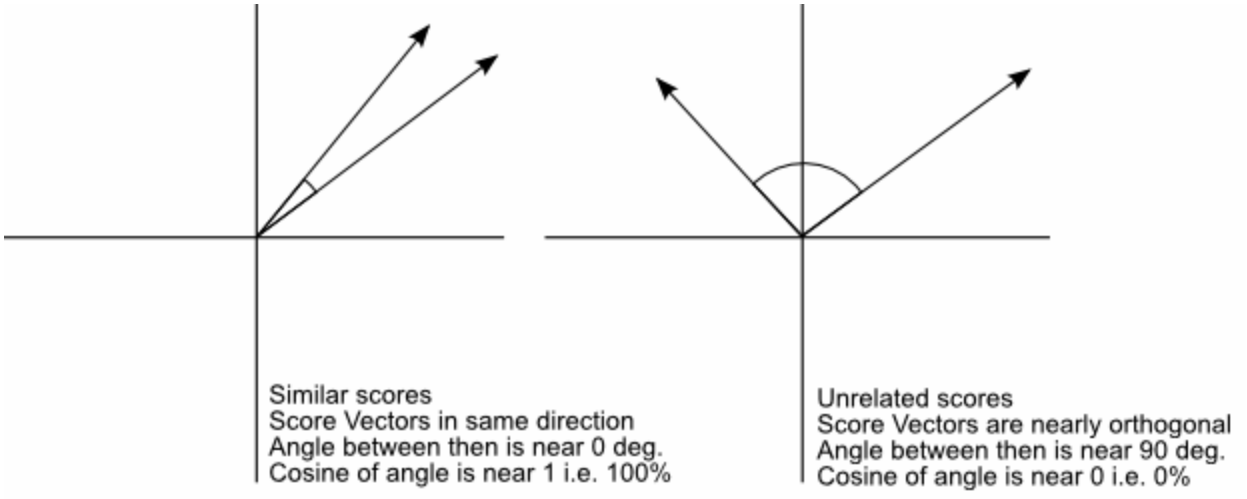
Table 5 Similarity measure between user using Tanimoto coefficient

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Similarity to User 1
User 1	√	√	√					1.0
User 2	√	√	√	√				0.75
User 3	√			√	√		√	0.17
User 4	√		√	√		√		0.4
User 5	√	√	√	√	√	√		0.5

3.4.4 Cosine Vector Similarity

Cosine vector similarity is statistics popular metrics. As it work only on angle between two vectors without their magnitude, it is an exceptionally helpful with missing preference data as long as it is capable to count the appearance of the term in the data . In the given formula, cosine vector similarity manipulate the angle in two vectors (the main Item *i* and the another Item *j*) of ratings in n dimensional item space. *R*k,i is the rating of the target Item *i* by User *k*. *R*k,j is the rating of the other Item *j* by user *k*. *n* is the total number of all rating users to Item *i*

and Item *j*.



$$sim(i,j) = \cos(\vec{i},\vec{j}) = \frac{\vec{i} \times \vec{j}}{\|\vec{i}\|^2 \times \|\vec{j}\|^2} = \frac{\sum_{k=1}^{n} R_{k,i} R_{k,j}}{\sqrt{\sum_{k=1}^{n} {R_{k,i}}^2 \sum_{k=1}^{n} {R_{k,j}}^2}}$$

When angle in two vectors is 0 degree nearly (then they are pointing to the same direction), Cosine similarity value, *sim(i,j)*, is 1, shows them very similar. When angle in two vectors is 90 degree nearly, *sim(i,j)* is 0,shows irrelevance. When the angle between two vectors is 180 degree nearly (then they are pointing in the opposite direction), *sim(i,j)* is -1, shows very dissimilar. In case the of collaborative filtering CF, *sim(i,j)* ranges from 0 to 1. The reason is that angle in two vector can not be greater than 90 degrees.

CHAPTER 4

USER-BASED COLLABORATIVE FILTERING

4.1 Introduction

User-based collaborative filtering method do prediction of items to end user by calculating similarity between same type of user using ratings given to items. To assess similarities among users User-based collaborative filtering require items explicit rating scores rating given by users and utilize k-nearest neighbor method to search nearest neighbors on the basis of user similarities. And after that, it gives prediction of items by summing up the neighbor user's rating values on the basis of weighted average of similarity.

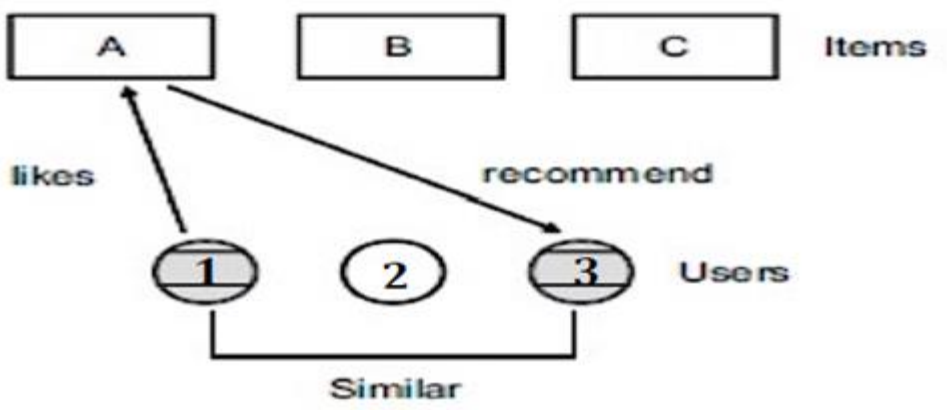


Figure 10 User based collaborative filtering

In the below figure User 1 & User 3 are showing similar rating behavior and both has good similarity score between them .If User 1 agree with Item A, then according to User-based collaborative filtering algorithm ,it will recommend User 3 with Item A .In figure given below,user1 liked two movie inception and forest gump and after calculating similarity, two user Alex and Chris are neighbor of user 1 and from these two ,system recommend two movies Dallas Buyer and lawless.

Figure 11 Example of user based recommender system

4.2 Algorithm

User Based Collaborative Filtering Algorithm

1 **Input**: User-Item Rating matrix R
2 **Output**: Recommendation lists of size l
3 **Const** l: Number of items to recommended to user u
4 **Const** v: Maximum number of users in $\mathcal{N}(u)$, the neighbours of user u
5 **For** each user u **Do**
6 **Set** $\mathcal{N}(u)$ to the v users most similar to user u
7 **For** each item i that user u has not rated **Do**
8 Combine ratings given to item i by neighbours $\mathcal{N}_i(u)$
9 **End**
10 Recommend to user u the top-l items with the highest predicted rating $\hat{r}_{ui}$
11 **End**

In the algorithm Line 1 define input of the User-Item Ratings matrix R and Line 2 is output for size L recommendation list . Line 3, which define L as system need. In the Line 4, v tells neighborhood size for users in finalizing the similar candidates set of target user. The bulk loop list all users in Line 5, and each one step assess the most similar k users of target in Line 6. Similarity function is used assess similar users , being the Cosine Distance , the Pearson

Correlation, and euclidean distance . The neighborhood selection algorithm is used to select the most similar users., being the Threshold Filtering algorithms or the Top N-Neighbors . After that , each item for that target user did not given rating in Line 7. In Line 8 we sum up the ratings showed by neighbors to estimate missing entry of rating. Finally in Line 10 , we present the recommendation to the target user.

4.3 Merits and Demerits

User based collaborative filtering does not require to analyze the feature of the item. It basically works on only ratings given by the user and calculation of user similarities without knowing about the item ,it can recommend the item to target user easily . Beside of this, it face the problem of new user who has not given any rating and new to the system. In that case it is difficult to get similar user and give recommendation. Another problem is scalability as there are millions of user for the system who are using it so it becomes slow in computation. For new item also, it cannot predict ratings till some similar users have given rating for it. Sparsity is major problem ,due to greater empty fields in rating matrix , it reduces the accuracy of the recommendation.

CHAPTER 5

ITEM-BASED COLLABORATIVE FILTERING

5.1 Introduction

Item-based collaborative filtering approach works on similarities between items that user already rated and other one. Item-based collaborative filtering maintain a list of items that user has rated and calculate similarity between them and others .And after that , it predict score of target one by summing up the previous preferences of target user on the basis of item similarities . In IBCF, preference data of users can be gathered through two ways. One way is that user rate item within numerical scale explicitly. The other one is that it analyzes click-through rate or user's purchase records. For example, here Item A and Item C are similar, . If User agree with Item A, IBCF will recommend user the item C.

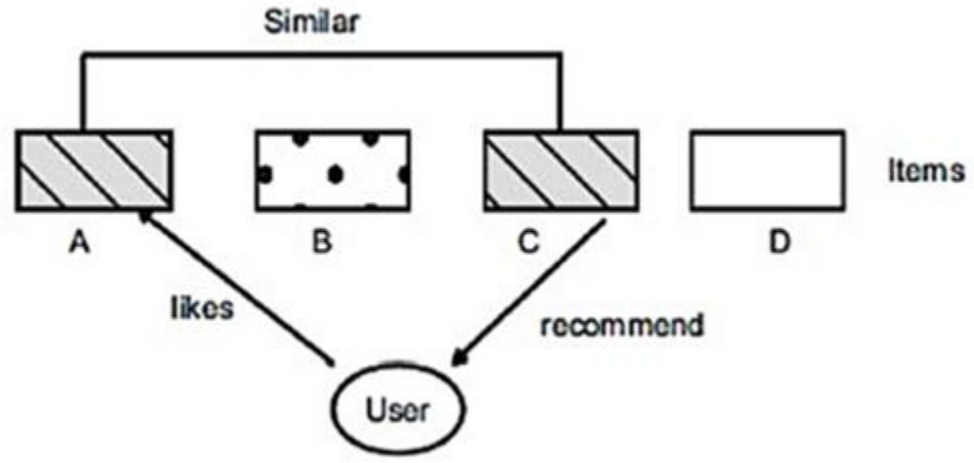


Figure 12 Item based collaborative filtering

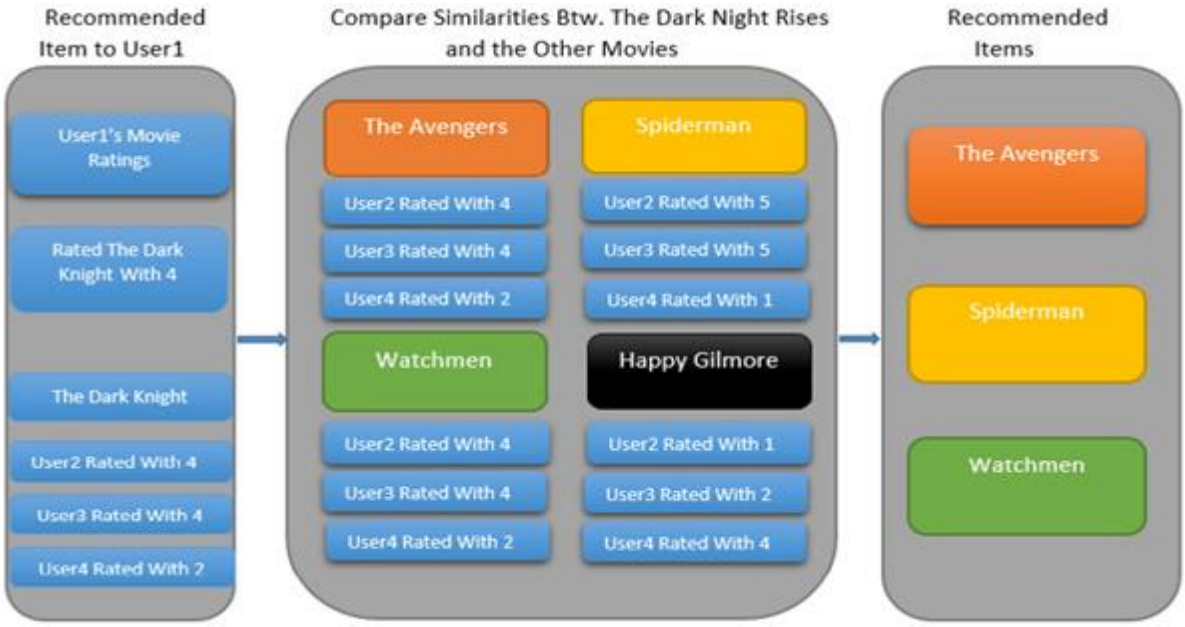


Figure 13 Example of item based recommender system

In this example ,user has rated movie The Dark night, which is similar to The Avengers, Watchman , Spiderman as user rating terms ,so system will recommend all three movie to the user.

5.2 Algorithm

Item Based Collaborative Filtering Algorithm

Input: User-Item Rating matrix R
Output: Recommendation lists of size l
Const l: Number of items to recommended to user u
Const j: Maximum number of items in $\mathcal{N}(i)$, the neighbours of item i
For each item i **Do**
  Set $\mathcal{N}(i)$ to the j items most similar to item i
  For each user u that has no rating for item i **Do**
    Combine ratings of user u in neighbours $\mathcal{N}_u(i)$
  End
  Recommend to user u the top-l items with the highest predicted rating $\hat{r}_{ui}$
End

In the algorithm Line 1 define input of the User-Item Ratings matrix R and Line 2 is output for size L recommendation list . Line 3, which define L as system need. In the Line 4, v tells neighborhood size for users in finalizing the similar candidates set of target user. The bulk loop list all items in Line 5, and each one step assess the most similar k items of target in Line 6. Similarity function is used assess similar users , being the Cosine Distance , the Pearson Correlation, and euclidean distance . The neighborhood selection algorithm is used to select the most similar users., being the Threshold Filtering algorithms or the Top N-Neighbors . After that , each item for that target user did not given rating in Line 7. In Line 8 we sum up the ratings showed by neighbors to estimate missing entry of rating. Finally in Line 10 , we present the recommendation to the target user.

5.3 Merits and Demerits

Item collaborative filtering does not require knowledge about item feature as it recommend items to the target user by calculating similarities between items using user-item rating matrix. The number of items is limited as compare to number of users in the system so it improve scalability and also similarity between items is more stable than between users which makes pre-computation work easy. Beside of this ,it has difficulty to handle the problem of entrance of new item in the system .It cannot predict which items are similar till we have ratings for this item.

CHAPTER 6

HYBRID-BASED COLLABORATIVE FILTERING

6.1 Introduction

This model combine the two collaborative filtering method and define new model .In this model rating prediction is based on user similarity in user-based collaborative filtering and item similarity in item-based collaborative filtering. User similarity define the user group ,user in the group are close to each other and each user in this group contribute to get rating of the item . similarly item similarity define the item group, all item in the group are close. Both these method give the unknown rating and these result can be combined to get final rating of the item

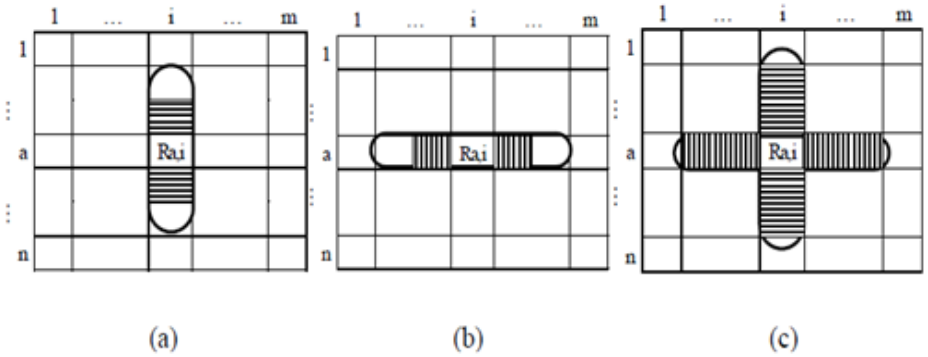


Figure 14 (a) Rating prediction based on user similarity in user-based CF; (b) Rating prediction based on item similarity in item-based CF; (c) Rating prediction based on hybrid similarity

Here we define sim(u,v) as user similarity for user u and v and sim(i,j) as item similarity for item i and j . $M(u)$ is a set of similar neighbors for user u and $M(i)$ is a set of similar neighbors for item i . We calculate these two sets and define two variable p and q ,where p is number of ratings given to target item by different user and q is total number of ratings to target

user given for different item . Using these two values we define balance factor φ,which is given below:-:

$$\phi = \begin{cases} 1 & M(u) \neq \varnothing; M(i) = \varnothing \\ \dfrac{p}{p+q} & M(u) \neq \varnothing; M(i) \neq \varnothing \\ 0 & M(u) = \varnothing; M(i) \neq \varnothing \end{cases}$$

If there is no value in $M(u)$ set then recommendation will be on the basis of item-item similarity and If there is no value in $M(i)$ set then recommendation will be on the basis of user-user similarity else in another case it will use the result of both one. Using above approach we predict the rating of item i ,to which user u has not given any rating ,is as follows:

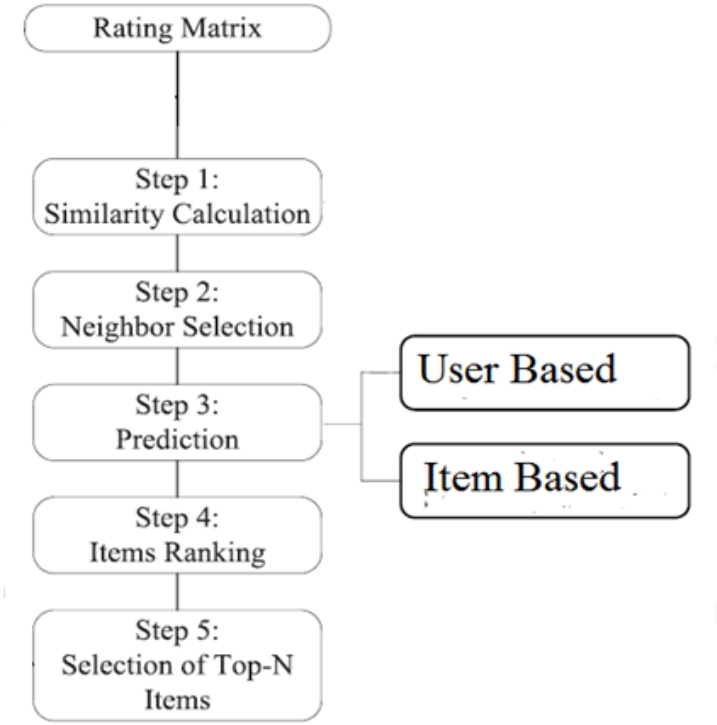
$$P(r_{u,i}) = \phi \times \left[\overline{r_u} + \frac{\sum_{v \in M(u)} sim\ (u,v)(r_{vi} - \overline{r_v})}{\sum_{v \in M(u)} sim\ (u,v)} \right] +$$

$$(1-\phi) \times \left[\overline{r_i} + \frac{\sum_{j \in M(i)} sim\ (i,j)(r_{uj} - \overline{r_j})}{\sum_{j \in M(i)} sim\ (i,j)} \right]$$

where, $P(r_{u,i})$ shows prediction value of user u to item i , φ is balance factor ,re r_u and r_v are mean rating of user u and user v for all the items respectively; r_i and r_j are mean rating of item i and item j for all the users respectively.

6.2 Flow graph

Below flow graph shows the step to find the recommendation using Hybrid collaborative filtering algorithm.It first calculate the similarity between user and item and then select the neighbor who are more similar than others..Using these neighbours it predict the unkown rating based on both user and item.Than rank the item on the basis of calculated rating and show the best N items to the user



6.3 Algorithm

Algorithm . Hybrid Collaborative Filtering Algorithm

```
1: Input: User-Item Rating matrix R
2: Output: Recommendation lists of size l
3: Const l : Number of items to recommended to user u
4: Const v: Maximum number of users in N(u), the neighbours of user u
5: Const j: Maximum number of items in N (i), the neighbours of item i
6: for each item i do
7:     Set N(i) to the j items most similar to item i
8: end for
9: for each user u do
10:    Set N(u) to the v users most similar to user u
11:    for each item i that user u has not rated do
12:        Get weighted sum of ratings given to item i by neighbours Ni(u)
13:        Get combine ratings of user u in neighbours Nu(i)
14:        Combine the both ratings for item i of user u with weight factor of available ratings
15:    end for
16: end for
17: Recommend to user u the top-l items with the highest predicted rating
```

In the algorithm Line 1 shows the User-Item matrix R of Ratings and Line 2 is output which is recommendation list of size l (define in Line 3), Line 4 and 5 define the two constant that are number of user in user neighbor set and number of item in item neighbor set. Line 6 -8 calculate the all items neighbor set then we iterate for each user in loop in line 9and find neighbor of that user in the iteration. Line 12 and 13 calculate the weighted rating based on user and item respectively and then we combine both one using weight factor to get final rating .Line 17 give the final recommendation by listing top l rated item

6.4 Performance Evaluation Criteria

Mean Absolute Error (MAE)

We use statistical accuracy metrics to assess the accuracy of the recommendation system. Mean Absolute Error (MAE) is mostly used metric system in the recommendation using collaborative filtering to compute the deviation of calculated ratings to the true actual ratings.In the below formula ,Total number of actual ratings in item set is N. p_i is the prediction of user's ratings. q_i is corresponding real ratings data set of users.

$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N}$$

It computes the average of the absolute difference between Npairs; prediction scores of users' ratings and actual user ratings for the user-item pairs in the dataset. Lower the MAE value, better is the recommendation system's accuracy of prediction of user ratings.

Root Mean Squared Error (RMSE)

The square root of the mean/average of the square of all of the error. MSE is very commonly used and makes for an excellent general purpose error metric for numerical predictions. Compared to the similar Mean Absolute Error, RMSE amplifies and severely punishes large errors.

$$RMSE = \sqrt{\frac{1}{n} \sum_{\{i,j\}} \left(p_{i,j} - r_{i,j} \right)^2}$$

CHAPTER 7

EXPERIMENT RESULTS

7.1 Dataset:

The data set used in this experiment contains around 10000 ratings of 1682 movies given by 896 user. This data is suitable for movie recommendation systems. It consists of user ID, movie ID and Rating .User ID is in range from 1 to 896 and movie ID is in range from 1 to 1682 . Ratings are given on a 5-star scale (only whole-star ratings) . The ratings range is from 1 (less interesting) to 5 (very interesting) as integer type.

Table 6 USER-ITEM Matrix raw dataset

Item / User	1035	1380	1287	3408	1201	…
1	5		5	4		…
6	5	5		5		…
10	5	5	3	4	2	…
26	2	4		2	2	…
…	…	…	…	…	…	…

7.2 Experiment Environment:

- Processor: 1.7 GHz Intel Core i3
- Memory: 4 GB 1600 MHz DDR3
- Operation System: Window 7 Home Basic
- Language: Java
- IDE: Eclipse mars

7.3 Results and Analysis

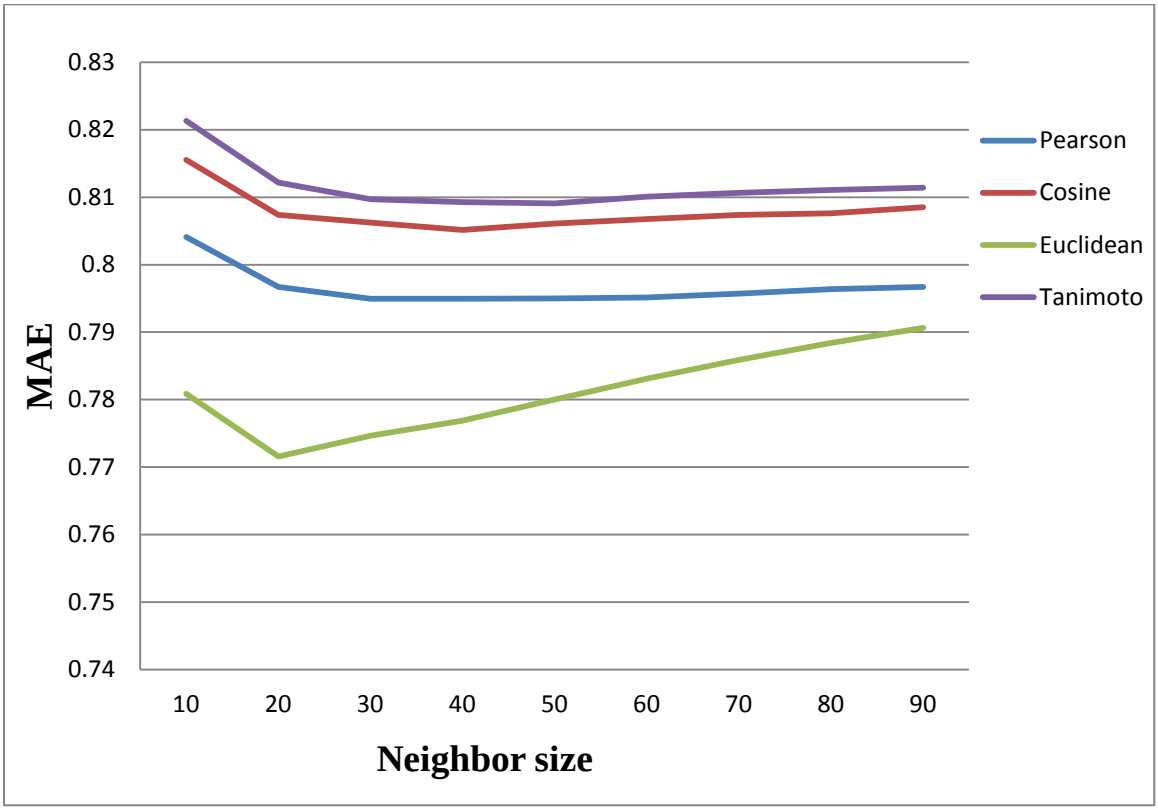


Figure 15 User Based Collaborative Filtering(UBCF)

User based collaborative filtering graph shows the comparison between different similarities used in algorithm. It has used pearson correlation ,Euclidean distance, tanimoto coefficient and cosine vector. This graph is depicting neighbor size on x axis and mean absolute error on y axis. The size of the Neighbor affects the prediction quality. In this graph Tanimoto similarity is giving more error as it does not work on rating value while check whether value present or not .In opposite of this Eucledean similarity shows minimum error as it work on actual distance between two ratings vector points. In between of two cosine and pearson similarities are present.Cosine work on actual angle between two vector and pearson calculate the similarity after isolating the common list of items which are rated by both user.

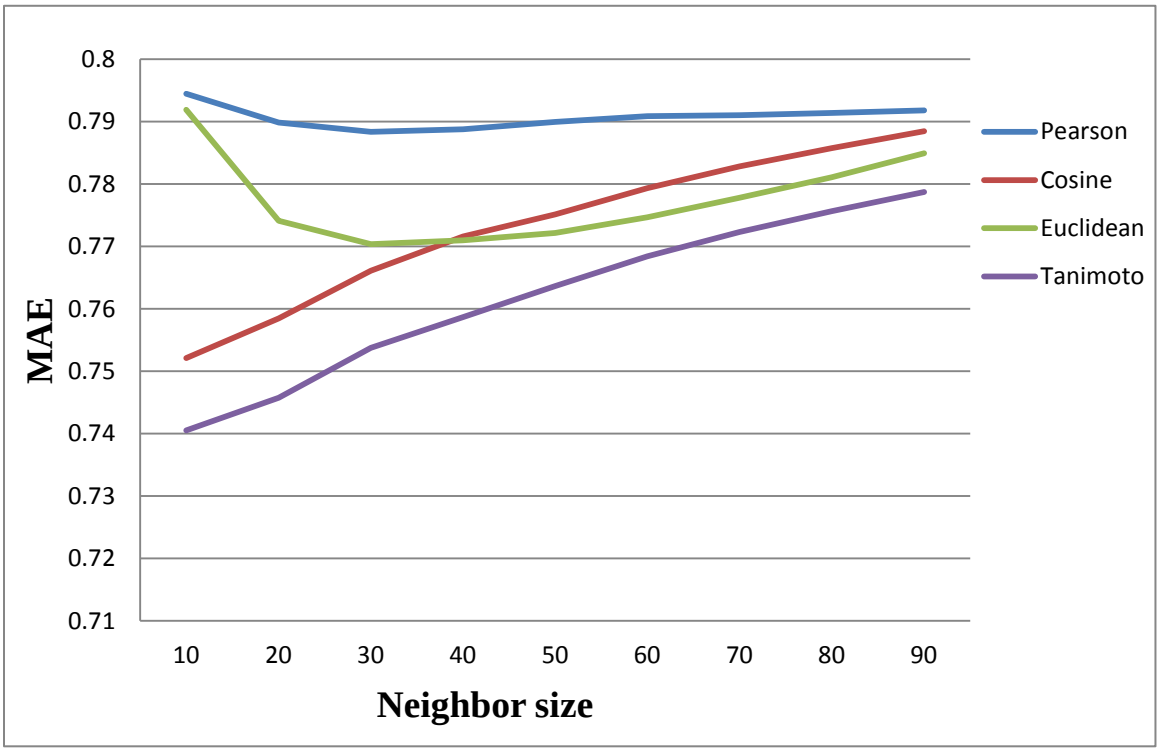


Figure 16 Item Based Collaborative Filtering(IBCF)

Item based collaborative filtering give different results for different similarity measure used in the algorithm. It shows the effect of neighbor size on result with error. In the graph pearson showed more error than other one as this algorithm find similarities between items and to calculate pearson similarity ,it find the common user who has given rating to item and which is small ,so this gives bad result. Tanimoto has given good result as it does not work on rating values just work on whether rating is present or not.

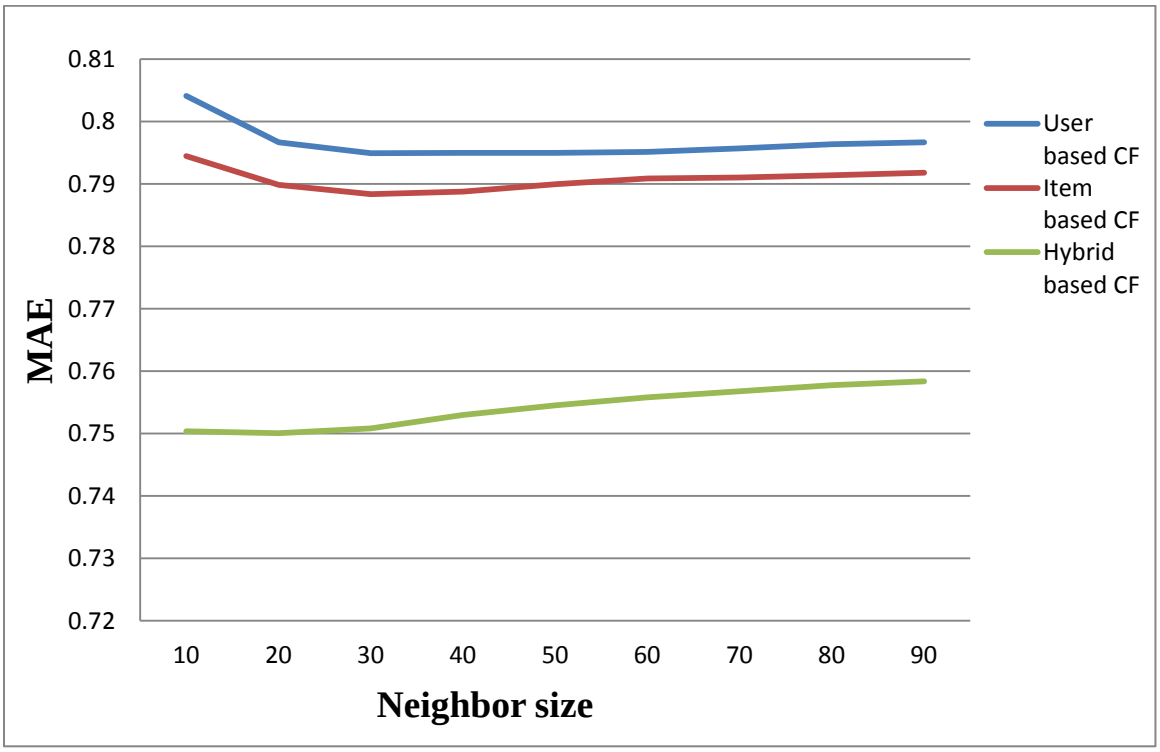


Figure 17 Hybrid Based Collaborative Filtering(HBCF)

Hybrid based collaborative filtering gives better result than user based and item based algorithm. In this algorithm result of both one is combined to get better result. For HBCF pearson similarity is used to get similar user and item .Individually both suffer from user-item rating matrix sparsity problem but after combining them this problem can be resolved to some extent. There may be a case when less number of similar users ratings available in UBCF in comparison of similar item ratings in IBCF than weightage will be given to prediction based on IBCF more and vice-versa. In this way ,this algorithm try to mitigate some error in rating prediction.

CONCLUSION AND FUTURE WORK

In this thesis we compared the different algorithms of collaborative filtering recommendation system. These are user based collaborative filtering(UBCF), item based collaborative filtering(IBCF) and hybrid based collaborative filtering (HBCF) .UBCF works on calculation of similarity between users of the system and IBCF calculates similarity among items of the system. For this calculation various similarity measures have been used like euclidean distance, pearson correlation, tanimoto coefficient , cosine vector etc Behavior of each similarity is different for both algorithms. To assess this behavior , Mean absolute error is calculated for varying size of the neighbor of similar group. In the case of UBCF, Euclidean has shown the minimum error while tanimoto has given larger error. In other case of IBCF, tanimoto has given least error while pearson has shown maximum error. Using both these algorithm, we designed new algorithm which combine the rating prediction done by both one. We make hybridization of UBCF and IBCF. These two algorithm face the problem of data sparsity in user-item rating matrix ,that is more number of empty fields in the matrix. Due to sparse data ,It effects the process of rating prediction for recommendation and produces error. So we combine the predicted rating of both one to mitigate some error. According to the experiment MAE of all three is given below

UBCF>IBCF>HBCF

Hybrid technique is giving better result as compare to other technique.

Collaborative filtering still needs more accuracy in rating prediction to recommend item to the user. So in future ,we can apply machine learning techniques in addition to this algorithm to improve the accuracy of rating prediction process. This combination algorithm may give better result as compare to present.

REFERENCES

[1] F Ricci, L Rokach, B Shapira, "Introduction to recommender systems handbook", Springer US, 2011.

[2] D. Asanov, "Algorithms and Methods in Recommender Systems", Berlin Institute of Technology, Berlin, Germany, 2011.

[3] Sunday O. Ojo, Seleman M. Ngwira and Keneilwe Zuva Tranos Zuva, "A Survey of Recommender Systems Techniques, Challenges," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 11, November 2012.

[4] J. Bobadilla, F. Serradilla, and a. Hernando, "Collaborative filtering adapted to recommender systems of e-learning," *Knowledge-Based Syst.*, vol. 22, no. 4, pp. 261–265, May 2009.

[5] Goldberg D, Nichols D, and Oki M B. Using collaborative filtering to weave an information Tapestry, Dec 1992, vol.

[6] L. Yanxiang, G. Deke, C. Fei, and C. Honghui, "User-based clustering with top- N recommendation", in Third International Conference on Intelligent System Design and Engineering Applications (ISDEA), pp. 1585–1589, Jan 2013.

[7] J. L. Sanchez, F. Serradilla, E. Martinez, and J. Bobadilla, "Choice of metrics used in collaborative filtering and their impact on recommender systems", in 2nd IEEE International Conference on Digital Ecosystems and Technologies, pp. 432– 436, Feb 2008.

[8] G. Friedrich and D. Jannach, "Tutorial: Recommender Systems," In Proceeding of the International Joint Conference on Artificial Intelligence, Barcelona, July 2011.

[9] J. Ben Schafer, J. Konstan, and J. Riedl, "Recommender Systems in E-Commerce," pp. 158–166, 1999 in *Proceedings of the 1st ACM conference on Electronic commerce,* pp. 158-166. ACM, 1999.

[10] Andrei-Cristian Prodan, "Implementation of a Recommender System Using Collaborative Filtering", Studia Universitatis Babes-Bolyai, Informatica, vol. 55, no. 4, pp. 70-84, 2010.

[11] R. Burke, "Hybrid Web Recommender Systems," pp. 377–408, Springer Berlin Heidelberg, 2007.

[12] J. Lee, M. Sun, and G. Lebanon, "A comparative study of collaborative filtering algorithms," *arXiv Prepr. arXiv1205.3193,* pp. 1–27, 2012

[13] Francesco Ricci, Lior Rokach and Bracha Shapira, "Content-based recommender systems: state of the art and trends, recommender systems handbook", Springer, 2011, pp. 73-105

[14] M.J. Pazzani, "A framework for collaborative, content-based and demographic filtering." in *Artificial Intelligence Review* 13, no. 5-6, pp. 393- 408, 1999.

[15] R. Burke. "Knowledge-based recommender systems," in *Encyclopedia of library and information systems* 69, no. Supplement 32, pp. 175-186, 2000.

[16] www.marutitech.com/recommendation-engine-benefits

[17] Adomavicius, G Tuzhilin, A. (June 2005). "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions". in *IEEE Transactions on Knowledge and Data Engineering 17* (6): 734–749.

[18] O'Mahony, Michael P., and Barry Smyth. "A recommender system for on-line course enrolment: an initial study." in *Proceedings of the 2007 ACM conference on Recommender systems*, pp. 133-136. ACM, 2007

[19] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the 10th international conference on World Wide Web, ser. WWW '01. New York, NY, USA: ACM, 2001.

[20] Schafer, J. Ben, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. in *Proceedings of the 1st ACM conference on Electronic commerce*, pp. 158-166. ACM, 1999.

[21] Breese, John S., David Heckerman, and Carl Kadie. "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 43-52. Morgan Kaufmann Publishers Inc., 1998

[22] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," *Adv. Artif. Intell.*, vol. 2009, no. Section 3, pp. 1–19, 2009

ABBREVIATIONS

CBF --Content Based Filtering

CF --Collaborative Filtering

DBF--Demographic-Based Filtering

F --Female

HBCF --Hybrid Based Collaborative Filtering

IB-- Item Based

IBCF-- Item-Based Collaborative Filtering

KBF--Knowledge-Based Filtering

KNN--K Nearest Neighbor

M-- Male

MAE-- Mean Absolute Error

PCS --Pearson Correlation Similarity

RMSE-- Root Mean Squared Error

TCS--Tanimoto Coefficient Similarity

UB--User Based

UBCF-- User-Based Collaborative Filtering