Synopsis of the thesis entitled

Personalized Recommendation Algorithms with Collaborative Filtering

（協調フィルタ方式による個人への物件推薦アルゴリズム）

Submitted by

Thomurthy Murali Mohan

For the award of the Degree of

DOCTOR OF ENGINEERING

IN

GRADUATE SCHOOL OF ENGINEERING

Under the esteemed Supervision of

Professor KOICHI HARADA

DEPARTMENT OF INFORMATION ENGINEERING,
GRADUATE SCHOOL OF ENGINEERING,
HIROSHIMA UNIVERSITY,
HIGASHI-HIROSHIMA,
JAPAN – 2015.

D E C L A R A T I O N

I hereby declare that the work which is being presented in this thesis entitled **"Personalized Recommendation Algorithms with Collaborative Filtering"** in fulfillment of the requirements for the award of Doctor of Philosophy submitted to the Department of Information Engineering, Hiroshima University, Higashi Hiroshima, Japan, in an authentic record of my own research work carried out under the guidance and supervision of **Professor KOICHI HARADA**, Graduate School of Engineering, Hiroshima University, Higashi Hiroshima, Japan.

The material presented in the thesis has not been submitted to any other university or Institute for the award of any degree.

City: Visakhapatnam, India

Date: 27th Aug 2014.

Thomurthy Murali Mohan.

GRADUATE SCHOOL OF ENGINEERING

CERTIFICATE

This is to certify that the thesis entitled **"Personalized Recommendation Algorithms with Collaborative Filtering"** is being submitted by **Thomurthy Murali Mohan** to the Hiroshima University for the award of **Doctor of Philosophy** in **Department of Information Engineering, Graduate school of Engineering, Hiroshima University, Higashi Hiroshima**, **Japan** is a bonafide work which is carried out under my guidance and supervision and has fulfilled the requirements for submission of thesis, which has attained the standard required for a **Ph.D** degree of this university. The results presented in this thesis have not been submitted elsewhere for award of any Degree.

KOICHI HARADA

Dedicated to

My family

A C K N O W L E D G E M E N T S

Firstly, I thank all mighty God for guiding me and taking care of me all the time. My life is so blessed because of Him.

Then, I deem it a great privilege to express my gratitude to my research director **Professor Koichi Harada, Graduate School of Engineering, Hiroshima University, Higashi Hiroshima, Japan** for his esteemed guidance, encouragement, noble thoughts and very valuable suggestions for the successful completion of my research work. He has always been very patient and understanding, and has taken extremely good care throughout the course of this research work. His patience in reading draft after draft to every paper, proposal and idea I wrote up continues to amaze me. He has always been available for detailed technical discussions, which has shaped my thinking. He has been instrumental in my development as a researcher, my writing skills have improved under his guidance.

I am thankful to **Dr. Balakrishna. Annepu** for his guidance and I would like to thank him for his intuitive comments and suggestions, which have greatly improved the clarity of this thesis and publications.

I am thankful to **Professor Chuzo Iwamoto, Associate Professor Tadashi Shima, Associate Professor Yasuhiko Morimoto**, **Graduate School of Engineering, Hiroshima University, Higashi Hiroshima, Japan for** his guidance and his intuitive comments and suggestions, which have greatly improved the clarity of this thesis and publications.

I express my deepest sense of gratitude to my parents, ***Smt. Satyavathi,*** my mother, ***Sri. Pydi Raju,*** my father, **Mr.** ***Ravi Kishore*** my brother, **Smt**. ***Sailaja***, Sister, with whose encouragement, I would not have achieved anything significant in my life.

I also express my deep heartfelt thanks to my wife ***Smt. Venkata Satya Sheela,*** for the cooperation extended.

Thomurthy Murali Mohan

TABLE OF CONTENTS

PROBABILISTIC PROTOTYPE

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

An explosive growth of enormous information on the web, created the universe as global village. It is a big problem for getting the relevant information from the internet. Personalized Recommendation Systems may be used to get relevant information from the internet. Recommender System is to generate significant recommendations to a collection of users for items or products that might interest them. This is a powerful new technology for extracting additional value for a business from its user databases and help users find items they want to buy from a business. Real world examples for the recommender systems are amazon.com (for books) and netflix.com (for movies). Collaborative filtering is one of the important techniques in personalized recommendation systems and predicting the interests of a user by collecting preference information from many users.

The Collaborative Filtering models can also hold with the situations where user profiles are supplied by observing user interactions with a system and dealt with user profiles that are obtained by requesting users to rate information items. Broadly, they are classified into (i) Memory-based Collaborative Filtering techniques such as the user-based, item-based and neighborhood-based Collaborative filtering algorithm (ii) Model-based Collaborative Filtering techniques such as Bayesian belief nets, Clustering, Singular Value Decomposition (SVD) and MDP-based Collaborative filtering and (iii) Hybrid Collaborative filtering techniques such as the Content-boosted Collaborative Filtering and Personality Diagnosis.

In this thesis, a comprehensive study has been involved to process huge data sets and uses the popular collaborative filtering algorithms as the basis for proposed modifications. In the

beginning, the problem of inaccurate finding and falling recommendation quality of the prediction will bring forth. Then, the user's interest words will be collected to build the user interest model. Finally, modified algorithms have been proposed, they are 1) User based Collaborative Filtering based on Pearson Correlation which is Memory-based technique, 2) Singular Value Decomposition based on Composite Prototypes which is Model-based technique and 3) Hybrid Collaborative Filtering based on the predictions-probabilistic prototype.

The experimentation is done with MovieLens dataset which is available for research purpose provided by the GroupLens Research Project agency at the University of Minnesota. The measured Mean Absolute Error (MAE) of the proposed model are compared with available models from literature and finally the performance analysis is done based on parameter MAE. The comparative analysis and comprehensive study shows that Content-boosted Collaborative Filtering algorithm puts forward for better performance among the other comparative algorithms and hence, feasible solutions will be obtained using Content-boosted Collaborative Filtering recommendation methods instead of other recommendation methods.

CHAPTER 1
INTRODUCTION

An explosive growth of information on the web, converted universe into a global village. While searching, getting the relevant information from the internet may not be possible for several times. The large growth in the amount of available information, the growing number of visitors to World Wide Web and the limitations of search engines create many challenges for recommendation systems. Personalized Recommendation Systems may be used to get relevant information from the internet. Recommender System is to generate significant recommendations to a collection of users for items or products that might interest them. This is a powerful new technology for extracting additional value for a business from its user databases and help users find items they want to buy from a business. Recommender systems real world examples are amazon.com [39] (for books) and netflix.com [41] (for movies). Thus the objective of Recommender System is to generate recommendations. Collaborative Filtering uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users. The fundamental assumption of Collaborative filtering [1] is that if two different users rate '*n*' items similarly, or have similar behaviors and hence will rate or act on other items similarly. Collaborative filtering is a widely used technique for information filtering in personalized recommendation systems and one of the important techniques which predicts the interests of a user by collecting preference information from many users.

Collaborative Filtering

Content-based filtering and collaborative filtering (CF) are two technologies used in recommender systems. Content-based filtering systems analyze the contents of a set of items together with the ratings provided by individual users to infer which non-rated items might be of interest for a specific user. Collaborative filtering methods accumulate a database of item ratings cast by a large set of users and then use those ratings to predict user's preferences for items. Collaborative filtering does not depend on the content descriptions of

items, but purely depends on preferences expressed by a set of users. These preferences can either be expressed explicitly by numeric ratings or can be indicated implicitly by user behaviors, such as clicking on a hyperlink, purchasing a book, or reading a particular news article. One major difficulty in designing content-based filtering systems lies in the problem of formalizing human perception and preferences. Practically it is not possible to formalize one user likes or dislikes a Movie or why he prefers one item over another. Like-wise, it is difficult to derive features which represent the difference between two items of extreme ends. Collaborative filtering provides a powerful way to overcome these difficulties. The information on personal preferences, tastes, and quality are all carried in either explicit or implicit user ratings. Collaborative filtering recommender systems have successfully been applied in areas ranging from e-commerce to computer-supported collaborative work.

The Collaborative Filtering are classified into (i) Memory-based Collaborative Filtering techniques such as the user-based, item-based and neighborhood-based Collaborative filtering algorithm; (ii) Model-based Collaborative Filtering techniques such as Bayesian belief nets, Clustering, Singular Value Decomposition (SVD) and MDP-based Collaborative filtering; and (iii) Hybrid Collaborative filtering techniques such as the Content-boosted Collaborative Filtering and Personality Diagnosis.

In Memory-based CF technique, User-based collaborative filtering systems depend on item rating predictions. The process of considering items to a user is based upon the opinions of people with similar likes or dislikes. Recommender systems helps to users to overcome information overload by providing personalized suggestions based on a history of a user's likes and dislikes. Memory-based CF techniques such as the User-based, Item-based CF techniques, Neighborhood-based CF computes similarity between users or items, and then uses the weighted sum of ratings or simple weighted average to make predictions based on the similarity values. Pearson correlation and vector cosine similarity are commonly used similarity calculations, which are usually conducted between co-rated items by a certain user or both users that have co-rated a certain item. To make top- N recommendations, neighborhood-based methods can be used according to the similarity values. Memory-based CF algorithms are easy to implement and have good performances for dense datasets.

Memory-based CF algorithms depend on user ratings, decreased performance when data are sparse, new users and items problems, and limited scalability for large datasets. Memory-based CF on imputed rating data and on dimensionality-reduced rating data will produce more accurate predictions than on the original sparse rating data.

In Model-based CF technique, the design and development of models can allow the system to learn to recognize complex patterns based on the training data, and then make intelligent predictions for the collaborative filtering tasks for test data or real-world data, based on the learned models. Model-based CF algorithms, such as Bayesian models, clustering models, and dependency networks, have been investigated to solve the shortcomings of memory-based CF algorithms. Usually, classification algorithms can be used as CF models if the user ratings are categorical, and regression models and SVD methods and be used for numerical ratings. .Model-based CF techniques need to train algorithmic models, such as Bayesian belief nets, clustering techniques, SVD or MDP-based ones to make predictions for CF tasks.

In Hybrid CF techniques combine CF methods with other recommender systems to alleviate shortcomings of either system and to improve prediction and recommendation performance. Every prediction technique has its own strengths and weaknesses; there is a need to combine different prediction techniques to increase the accuracy of recommender systems. The idea behind hybrid prediction techniques is that a combination of algorithms can provide more accurate recommendations than a single algorithm as disadvantages of one algorithm can be overcome by other algorithms. Most recommender systems use Collaborative Filtering methods to predict new items of interest for a user. While both methods have their own advantages, individually they fail to provide good recommendations in many situations. Incorporating components from both methods, a hybrid recommender system can overcome these shortcomings by Combine the information. Personality diagnosis (PD) is one of the important hybrid collaborative filtering techniques which describes given a user's preferences for some items to compute the probability that he or she is of the same personality type as other users, and the probability that he or she will like new

items. Traditional similarity-weighting techniques in that all data is brought to bear on each prediction and new data can be added easily and incrementally and probabilistic interpretation are the best aspects in Personality diagnosis.

Most recommender systems use Collaborative Filtering methods to predict new items of interest for a user. While both methods have their own advantages, individually they fail to provide good recommendations in many situations. Incorporating components from both methods, a hybrid recommender system can overcome these shortcomings by Combine the information.

Drawbacks of Filtering Techniques

The main drawbacks of these algorithms are Sparsity, cold-start (new user problem or new item), Synonymy, scalability, Gray Sheep and Shilling Attacks problems. Sparsity occurs when the user do not rate more items. In this case the sparse user-item rating value decreases which causes low value ratings in finding similar set of users. Cold-start is a basic problem concerned with an item that can only be recommended after rated by a user. Synonymy refers to the tendency of a number of the same or very similar items to have different names or entries. Gray sheep refers to the users whose opinions do not consistently agree or disagree with any group of people and thus do not benefit from collaborative filtering. Shilling Attacks refers to people may give tons of positive recommendations for their own materials and negative recommendations for their competitors.

Recommender System

Recommender System is to generate significant recommendations to a collection of users for items or products that might interest them. Recommender systems attempt to reduce information overload and retain customers by selecting a subset of items from a universal set based on user preferences and grew out of information retrieval and filtering. A wealth of information creates a poverty of attention, and a need to allocate that attention efficiently among the overabundance of information sources that might consume it, these recommender systems are present to give personalized recommendation on items to users based on the previous and present activity of the users and metadata about users and items. The design of

such recommendation systems depends on the domain and the particular characteristics of the data available. Recommender systems have evolved to fulfill the natural dual need of buyers and sellers by automating the generation of recommendations based on data analysis. It compares the user's profile to reference characteristics, and seeks to predict the 'rating' that a user would give to an item that they had not yet considered. These characteristics may be from the information.

Recommendation Process

Information recollection will be the base for the entire recommendation system which the recollection of users' personal preferences and information about items such as metadata, features extracted directly. This process is not performed by the recommendation system itself because the information collected presents incongruence or contradiction then the system will not be able to produce regular recommendations. Hence, proper attention should be adopted in collecting information that truly reflects the preferences of the users, or information that truly represent the items.

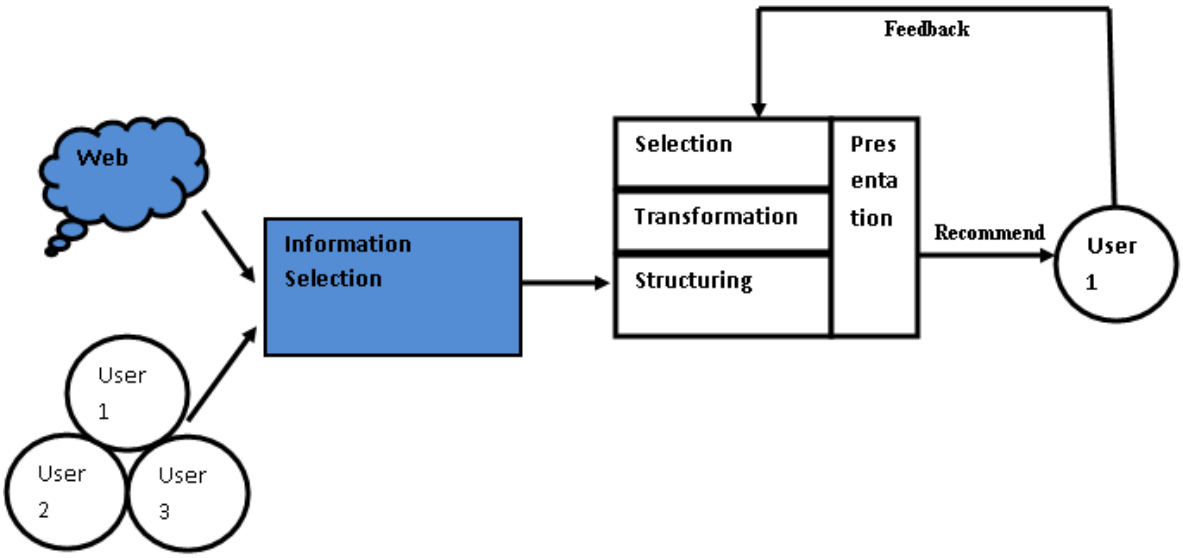

Fig. 1 Recommendation Process

Selection: consists of determining "which items are interesting or relevant enough for a user and removes all other items from the retrieved set of items". The way the selection of the items is done depends strictly on the approach taken to find items that are similar to the ones

the user consider relevant. Therefore, the key concept is defining similarity between items. The concept of similarity could be defined in terms of other users' profiles, of features extracted from the items, or in terms of metadata associated with the items.

Transformation: the main objective of the transformation is to perform some modifications to the items retrieved and proposed to addresses transformations such as summarization, change in the quality of the items, creation of thumbnails or snapshots for its further presentation.

Structure: The structure that the user will use to navigate through the different recommended items is related with the construction and organization. In addition to the above grouping the items according to certain characteristics, sorting the groups of items, sorting items inside these groups, linking items that have some relationship, etc. can be included.

Recommendation process: The recommendation process is associated with the presentation of the different retrieved and structured items to the final user. It deals with issues such as layouts, document formats, colors, fonts, and presentation medium. This process should be designed and executed carefully since the user will interact with the system by means of its results.

Feedback: The kind of feedback obtained from the user could be either implicit or explicit. In explicit feedback the user provides the system with information about how relevant the recommended items are. On the other hand implicit feedback is obtained from the user by analyzing his usage behavior. For instance, how much time he spends in looking at the retrieved items.

Recommendation System Advantages and Disadvantages

The advantages of a recommendation system are as follows

1. The recommendations on actual user behavior as their recommendations are not based on speculation, but on reality.
2. A recommendation system helps to discover things that are similar to what user already like which are not apparent recommendations.
3. Recommendation systems are always up-to-date as they are updated dynamically and the ability for a recommendation system to increase the activity is a highly advantageous.

The disadvantages of a recommendation system are as follows

1. Recommendation systems are intensive, database-driven applications that are significant to create and get running. Converting to a recommendation system takes time, energy, a lengthy commitment and development project.
2. Recommender system maintenance becomes a major task to keeping the system up and running even though it can, as it is with any significant database-backed system.
3. Sometimes people are unhappy with recommendations of a recommender system as they might be sometimes wrong. These systems are not just a technological challenge but they are also a social one.

Motivation

In 1992, the Tapestry system [80] introduced Collaborative Filtering. In 1994, the GroupLens system [114] implemented a Collaborative Filtering algorithm based on common users' preferences. Berry et al. carried out a survey of the computational requirements for managing LSI-encoded databases. Sarwar applied dimensionality reduction for only the user based Collaborative Filtering approach and included test users in the calculation of the model. Collaborative techniques use the similarity relation between users to generate recommendations for users. In 2001, another Collaborative Filtering algorithm was proposed which is based on the items similarities for a neighborhood generation of nearest items and is denoted as item-based Collaborative Filtering algorithm. According to

Burke recommender systems can be classified in five groups such as collaborative filtering, content-based, demographic, utility-based and knowledge-based techniques. The proposed thesis work is focused on collaborative filtering techniques. Later, Su et al. [33] proposed Collaborative filtering techniques as three main categories such as memory-based, model-based and hybrid CF algorithms.

The prediction quality of a collaborative filtering approach can be evaluated by comparing recommendations to a test set of known user ratings. These systems are typical measured using predictive accuracy metrics, where the predicted ratings are directly compared to actual user ratings. The most commonly used metric in the literature is Mean Absolute Error (MAE) [80] and it is defined as the average absolute difference between predicted ratings and actual ratings.

$$MAE = \frac{\sum_{\{u,i\}} \left| p_{u,i} - r_{u,i} \right|}{n}$$

where n is the total number of ratings over all users,
$P_{u,i}$ is the predicted rating for user i on item j, and
$r_{u,i}$ is the actual rating.

It is the most widely used metric in CF research literature, which computes the average of the absolute difference between the predictions and true ratings. The lower value of the MAE gives the better the prediction. Mean Absolute Error is a measure of the deviation of recommendations from their true user-specified values.

Characteristics of Mean Absolute Error are as follows

- It is impossible to use a particular item for estimating the value of MAE, if it is not rated.
- Errors in high rates or in low rates have the same impact on the MAE.

- The MAE may show a good accuracy for system that produce very accurately predicted item with low or average ratings but cannot predict correctly items with high ratings. In this case the system will not be able to produce a good recommendations to the user despite its MAE shows a good performance.
- MAE cannot be used to measure the accuracy of techniques that only produces a list of recommendations instead of a predicted rating for every item in the collection.

Cold start [41], data sparsity [3], scalability [54], synonymy [25], and so forth are the common problems in Recommender Systems. Other problems with collaborative recommendation systems include the subjective nature of the ratings, which is prone to user aversion, and the fact that user profiles may easily become outdated if users interests change and they do not use the system regularly enough for their profiles to be updated.

Sparsity [3] can be most of the users do not rate many items and hence the user ratings matrix is typically very sparse. This problem in Collaborative Filtering systems decreases the probability of finding a set of users with similar ratings. This problem often occurs when a system has a very high item-to-user ratio, or the system is in the initial stages of use. This issue can be mitigated by using additional domain information or making assumptions about the data generation process that allows for high-quality imputation.

The new items and new users create a significant challenge to recommender systems and collectively these problems are referred to as the coldstart [41] problem. The first of these problems arises in Collaborative Filtering systems, where an item cannot be recommended unless some user has rated it before. This problem applies not only to new items, but also to unclear items. As such the new-item problem is also often referred to as the first-rater problem. Since content-based approaches do not rely on ratings from other users, they can be used to produce recommendations for all items, provided attributes of the items are available. It is highly difficult to handle the new-user problem, since without

previous preferences of a user it is not possible to find similar users or to build a content-based profile. Scalability [54] is a serious problem in traditional collaborative filtering algorithms with computational resources going beyond practical or acceptable levels, when numbers of existing users and items grow tremendously. Synonymy [25] refers to the tendency of a number of the same or very similar items to have different names or entries. As such, research in this area has mostly focused on effectively selecting items to be rated by a user so as to rapidly improve recommendation performance with the least user feedback.

The main drawbacks of these algorithms are sparsity and cold-start problems. Sparsity occurs when the user do not rate more items. In this case the sparse user-item rating value decreases which causes low value ratings in finding similar set of users. Cold-start is a basic problem concerned with an item that can only be recommended after rated by a user. The most commonly used metric for prediction accuracy are Mean Absolute Error (MAE), Root Mean Square Error (RMSE), recall, precision and ROC sensitivity.

It is always desirable to design a Collaborative Filtering approach that is easy to implement, takes few resources, produces accurate predictions and recommendations, and overcomes all kinds of challenges presented by real-world Collaborative Filtering applications as mentioned above. Although there is no cure-all solution available yet, several researchers are working out solutions for each of the problems. At present, to alleviate the sparsity problem of Collaborative Filtering tasks, Singular Value Decomposition (SVD), which is a dimensionality reduction technique, is proposed for improvement. It is also remove unrepresentative users or items to reduce the dimensionalities of the user-item matrix directly.

OBJECTIVES

This section will briefly outline the objectives of the proposed work.

The main objective of the thesis is to minimize the drawbacks of various collaborative filtering recommendation systems and to improve the quality of various

prediction algorithms, which were measured by comparing the predicted values for the withheld ratings to the actual ratings. Finally, based on the modifications and performance evaluation, some modified collaborative filtering recommendation systems were proposed which would retain some of the best characteristics of the available methods and produce even better results.

The objectives are

a. To study the existing collaborative filtering algorithms.
b. To improve the quality of predictions by modifying the user-based collaborative filtering algorithm which function on memory-based collaborative filtering approach.
c. To improve the quality of predictions by introducing modifications into Singular Value Decomposition (SVD) which function on model-based collaborative filtering approach.
d. To improve the quality of predictions by making modifications to the Content-boosted Collaborative Filtering Algorithm which function on hybrid collaborative filtering approach.
e. To compare the metrics of the existing algorithms with the derived metrics of newly proposed and then compute predictions for the withheld items and establish their robustness during some uncertain conditions.
f. Measure the quality of various prediction algorithms and investigate the feasibility of these techniques for recommender systems.

ORGANIZATION OF THESIS

The thesis is organized into Seven Topics. An overview of these chapters is given below:

Chapter 1. Introduction

This chapter gives a brief introduction and basic concepts of the Recommender Systems and elaborative concepts of the classification of the collaborative Filtering techniques. The recommendation process in various collaborative filtering techniques are

presented and discussed in detail in this chapter. In addition to that, the architecture of collaborative filtering and the drawbacks, Similarity Computation measures, motivations, objectives, methodology and contributions of the present research work and chapter division of the thesis are presented.

Chapter 2. Literature Review

This chapter gives a review of the literature associated to the work presented in this thesis. The importance of the collaborative filtering area for researchers experimenting with new methodologies gave motivation to the present research work. Based on the above, the Problem Statement is presented.

Chapter 3. Memory-Based Collaborative Filtering Algorithm Based On User Similarity Using Pearson Correlation

This chapter presents the algorithms and methodology of the User-based and k-Nearest Neighbor algorithm (k-NN) collaborative filtering system algorithm based on the similarity. Pearson correlation coefficient is applied to measure the similarities of users or items which is better that other methods. The quality of predictions by introducing modifications to the User-based algorithm, programming environment, the quality of predictions evaluated based on the mean absolute error metric and the comparative statements are presented.

Chapter 4. Model-Based Collaborative Filtering Algorithm Based On Composite Prototypes

This chapter presents the algorithms and methodology of the Apriori algorithm and Singular Value Decomposition (SVD) collaborative filtering algorithm based on composite prototypes. The design and development of these models allowed the system to learn to recognize complex patterns based on the training data, and then make intelligent predictions for the collaborative filtering data based on the learned models. It computes singular values by the compact and corresponding singular vectors by the twisted factorization with inverse iteration. The quality of predictions by introducing modifications to the SVD algorithm,

programming environment, the evaluation process for improving quality prediction with the mean absolute error metric are presented.

Chapter 5. Hybrid Collaborative filtering based on probabilistic prototypes

This chapter presents the algorithm and methodology of the Content-boosted collaborative filtering algorithm based on probabilistic prototype. The key factor of this algorithm is to convert a sparse user-rating matrix into a full ratings matrix using content data. The quality of predictions by introducing modifications to the algorithm, programming environment, the quality of predictions evaluated based on the mean absolute error metric and the comparative statement are presented.

Chapter 6. Comparative Analysis and Conclusions

This chapter presents the brief comparison of the performance of the proposed collaborative filtering systems (a) Memory-Based Collaborative Filtering Algorithm Based On User Similarity Using Pearson Correlation, (b) Model-Based Collaborative Filtering Algorithm Based On Composite Prototypes, (c) Hybrid Collaborative filtering based on probabilistic prototypes, with that of the existing traditional collaborative filtering systems. The experimental evaluations, comparative analysis of the above collaborative filtering algorithms are discussed.

Chapter 7. References

This chapter gives the References address and links.

Finally, this chapter finally concludes the work with possible enhancements as future scope of this thesis.

CHAPTER II
LITERATURE REVIEW

The following sections deals with research reviews of various recommender systems, collaborative filtering and their classification of various algorithms which are available in the literature. This chapter further outlines the problem and approach of the present thesis work.

2.1 Review of Collaborative Filtering Techniques for Recommendation

[Markov B and Yoav S, 1997][1] made an attempt by combining both collaborative and content-based filtering systems, FAB may eliminate many of the weaknesses found in each approach. Online readers are in need of tools to help them cope with the mass of content available on the World-Wide Web. In traditional media, readers are provided assistance in making selections. This includes both implicit assistance in the form of editorial oversight and explicit assistance in the form of recommendation services such as movie reviews and restaurant guides

[John S.B, David H and Carl K, 1998][2] proposed Collaborative Filtering or Recommender Systems use a database about user preferences to predict additional topics or products a new user might like. They described several algorithms designed for this task, including techniques based on correlation coefficients, vector-based similarity calculations, and statistical Bayesian methods. They compared the predictive accuracy of the various methods in a set of representative problem domains.

[David M. P, Eric H and Lee G.C, 2000][3] proposed to estimate the growth of Internet commerce which has stimulated the use of collaborative filtering (CF) algorithms as recommender systems. Such systems leverage knowledge about the known preferences of multiple users to recommend items of interest to other users. They described and evaluated a new method called personality diagnosis (PD). Given a user's preferences for some items and computed the probability that user is of the same "personality type" as other users, and, in turn, the probability that he or she will like new items.

[Thomas T and Robin C, 2000] [4] made an attempt in electronic commerce applications, prospective buyers may be interested in receiving recommendations to assist with their purchasing decisions. They presented architecture for designing a hybrid recommender system that combines these two approaches and how such a recommender system can switch between the two methods, depending on the current support for providing good recommendations from the behaviour of other users, required for the collaborative filtering option.

[Badrul S, George K, Joseph K, and John R, 2001] [5] evaluated Recommender Systems applied knowledge discovery techniques to the problem of making personalized recommendation for information, products or services during a live interaction. These systems, especially the k-nearest neighbors' collaborative filtering based ones, are achieving widespread success on the web. By using traditional collaborative filtering system the amount of work increases with the number of participants in the system. New recommender system technologies are needed that can quickly produce high quality recommendations even for very large-scale problems.

[Jonathan L.H, Joseph A, Konstan, Loren G. T and John T.R, 2001][6] recommender systems have been evaluated in many, often incomparable, ways. They review the key decisions in evaluating collaborative filtering recommender systems: the user asks being evaluated, the types of analysis and datasets being used, the ways in which prediction quality is measured, the evaluation of prediction attributes other than quality, and the user-based evaluation of the system as a whole.

[Ken G, Theresa R, Dhruv G and Chris P, 2001] [7] proposed Eigentaste is a collaborative filtering algorithm that uses universal queries to elicit real-valued user ratings on a common set of items and applies Principal Component Analysis (PCA) to the resulting dense subset of the ratings matrix. For a database of n users, standard nearest-neighbor techniques require O(n) processing time to compute recommendations, whereas Eigentaste requires O(1) (constant) time.

[Prem M, Raymond J. M and Ramadass N, 2001] [8] described most recommender systems use Collaborative Filtering or Content-based methods to predict new items of interest for a user. They presented an elegant and effective framework for combining content and collaboration and

approach uses a content-based predictor to enhance existing user data, and then provides personalized suggestions through collaborative filtering. The experimental results that show how this approach, Content-Boosted Collaborative Filtering, performs better than a pure content-based predictor, pure collaborative filter, and a naive hybrid approach and also discuss methods to improve the performance of our hybrid system.

[Mimi M. R, Andrew W and Kimberly L, 2001][9] proposed a system based on information filtering technique, collaborative filtering, and its application in an educational setting. This system, called Altered Vista, supports the discovery of resources in a way that is sensitive to the context of users. They provides a means for supporting community-building activities by automatically recommending like-minded users for possible future collaboration.

[Steve C and Uwe A, 2002][10] proposed the immune system is a complex biological system with a highly distributed, adaptive and self-organizing nature. An artificial immune system (AIS) that exploits some of these characteristics and is applied to the task of film recommendation by collaborative filtering (CF). Natural evolution and in particular the immune system have not been designed for classical optimization.

[Zan H, Wingyan C, Thian-Huat O, Hsinchun C, 2002] [11] shows that recommendations comprise a valuable service for users of a digital library. Due to the similarity between our problem and a concept retrieval task, a Hopfield net algorithm was used to exploit high-degree book-book, user- user and book-user associations. Sample hold-out testing and preliminary subject testing were conducted to evaluate the system, by which it was found that the system gained improvement with respect to both precision and recall by combining content-based and collaborative approaches.

[Edward F. H, 2003][12] implemented truly online large margin version of the Perception ranking (PRank) algorithm, called the OAP-BPM (Online Aggregate Prank-Bayes Point Machine) algorithm, which finds a rule that correctly ranks a given training sequence of instance and target rank pairs. The OAP-BPM algorithm is an extension of this algorithm by approximating the Bayes point, thus giving a good generalization performance. The Bayes point

is approximated by averaging the weights and thresholds associated with several PRank algorithms run in parallel. In order to ensure diversity amongst the solutions of the PRank algorithms they randomly subsample the stream of incoming training examples.

[Michael C, Daniel Z, Hsinchun C, Michael H and David H, 2003] [13] described most existing Web search tools work only with individual users and do not help a user benefit from previous search experiences of others. They present the Collaborative Spider, a multi-agent system designed to provide post-retrieval analysis and enable across-user collaboration in Web search and mining. This system allows the user to annotate search sessions and share them with other users and also reported a user study designed to evaluate the effectiveness of this system.

[Michelle A, Daniel L, Marcel B, Harold B, Stephen G, Nancy H and Sean M, 2003] [14] gave an overview of the RACOFI (Rule-Applying Collaborative Filtering) multidimensional rating system and its related technologies. This would be exemplified with RACOFI Music, an implemented collaboration agent that assists on-line users in the rating and recommendation of audio (Learning) Objects. It lets users' rate contemporary Canadian music in the five dimensions of impression, lyrics, music, originality, and production.

[Andrew W, Mimi M. R, Kimberly L and David W, 2004] [15] reviews the information filtering techniques, collaborative information filtering techniques, which supports the discovery of resources in a way that is sensitive to the context of users. Moreover, via statistical clustering techniques, the system supports automated, personalized filtering and recommendation of relevant resources and like-minded users for particular user communities.

[Byeong M. K and Qing L, 2004] [16] estimated with the development of e-commerce and the proliferation of easily accessible information, recommender systems have become a popular technique to prune large information spaces so that users are directed toward those items that best meet their needs and preferences. While many collaborative recommender systems (CRS) have succeeded in capturing the similarity among users or items based on ratings to provide good recommendation, there are still some challenges for them to be a more efficient. They addressed three problems in CRS (user bias, non-transitive association, and new item problem) and provide

a new item-based probabilistic model approach in order to solve the addressed problems in hopes of achieving better performance.

[Kyung-Y. J, Dong-H. P and Jung-H. L, 2004][17] described about the growth of the Internet has resulted in an increasing need for personalized information systems. Introduced an autonomous agent, WebBot: Web Robot Agent, which integrates with the web and acts as a personal recommender system that cooperates with the user on identifying interesting pages. Hybrid components from collaborative filtering and content-based filtering, a hybrid recommender system can overcome traditional shortcomings.

[Ludovic D and Patrick G, 2004] [18] tells recently, a new community has started to emerge around the development of new information research methods for searching and analyzing semi-structured and XML like documents. The goal is to handle both content and structural information, and to deal with different types of information content (text, image, etc). They considered the task of structured document classification and proposed a generative model able to handle both structure and content which is based on Bayesian networks and show how to transform this generative model into a discriminate classifier using the method of Fisher Kernel. The model is then extended for dealing with different types of content information.

[Saverio P, Marcos A.G¸ Alves E.A.F, 2004] [19] assessed several Recommender Systems attempt to reduce information overload and retain customers by selecting a subset of items from a universal set based on user preferences. While research in recommender systems grew out of information retrieval and filtering, the topic has steadily advanced into a legitimate and challenging research area of its own. They took a connection-oriented perspective toward recommender systems research and posit that recommendation has an inherently social element and is ultimately intended to connect people either directly as a result of explicit user modeling or indirectly through the discovery of relationships implicit in extant data.

[Yu L, Liu L and Li X, 2004] [20] one of the challenges in web is ability of recommender systems to be adaptive to environment where users have many completely different interests or items have completely different content. Unfortunately, the traditional collaborative filtering

systems cannot make accurate recommendation for the two cases because the predicted item for active user is not consist with the common interests of his neighbor users.. Based on analysis, collaborative filtering based on item and user for Multiple-interests and multiple content recommendations is presented. Finally, experimentally evaluated the results and compare them with collaborative filtering based on user and collaborative filtering based on item, respectively.

[CaiNicolas Z, Sean M. McNee, Joseph A. K and Georg L, 2005][21] presented the topic diversification, a novel method designed to balance and diversify personalized recommendation lists in order to reflect the user's complete spectrum of interests. Their work builds upon prior research on recommender systems, looking at properties of recommendation lists as entities in their own right rather than specifically focusing on the accuracy of individual recommendations.

[Gediminas A and Alexander T, 2005][22] presented an overview of the field of recommender systems and describes the current generation of recommendation methods that are usually classified into the following three main categories: content-based, collaborative, and hybrid recommendation approaches.

[Gui-Rong X, Chenxi L, Qiang Y, WenSi X, Hua-Jun Z, Yong Y and Zheng C, 2005] [23] studied on Memory-based approaches for collaborative filtering identifies the similarity between two users by comparing their ratings on a set of items. In the past, the memory-based approaches have been shown to suffer from two fundamental problems: data sparsity and difficulty in scalability. They presented a novel approach that combines the advantages of these two kinds of approaches by introducing a smoothing-based method.

[Guy S, David H and Ronen I. B, 2005][24] described Markov decision processes (MDPs) provide a more appropriate model for recommender systems. MDPs introduce two benefits: they take into account the long-term effects of each recommendation and the expected value of each recommendation. To succeed in practice, an MDP-based recommender system must employ a strong initial model, must be solvable quickly, and should not consume too much memory.

[Manos P and Dimitris P, 2005] [25] discussed several prediction algorithms and evaluated, some of which are novel in that they combine user-based and item-based similarity measures derived from either explicit or implicit ratings. Both statistical and decision-support accuracy metrics of the algorithms are compared against different levels of data sparsity and different operational thresholds. The first metric evaluates the accuracy in terms of average absolute deviation, while the second evaluates how effectively predictions help users to select high quality items.

[Miha G, Blaz F and Dunja M, 2005] [26] discussed about experimental results of confronting the k-Nearest Neighbor (kNN) algorithm with Support Vector Machine (SVM) in the collaborative filtering framework using datasets with different properties. They concluded that the quality of collaborative filtering recommendations is highly dependent on the quality of the data. Furthermore, can see that kNN is dominant over SVM on the two standard datasets.

[Monica C and J.D. Tygar, 2005] [27] proposed an Current CAPTCHAs required users to solve objective questions such as text recognition or image recognition based on collaborative filtering. Collaborative filtering CAPTCHAs allow us to ask questions that have no absolute answer; instead, the CAPTCHAs are graded by comparison to other people's answers. They analyze the security requirements of collaborative filtering CAPTCHAs and find that although they are not ready to use now, collaborative filtering CAPTCHAs are worthy of further investigation.

[Yan Z W, Luc M, and Nicholas R. J, 2005] [28] evaluated and studied about Recommender systems have been widely advocated as a way of coping with the problem of information overload for knowledge workers. Given this, multiple recommendation methods have been developed. However, it has been shown that no one technique is best for all users in all situations. Specifically, this article presents the principled design of a marketplace and evaluates the market's capability to effectively coordinate multiple methods.

[Ya-Y. S and Duen-R L, 2005][29] collaborative filtering (CF) method has been successfully used in recommender systems to support product recommendation, but it has several limitations. This work explores two hybrid approaches each of which combines CF and customer demands to

improve quality of recommendation. Valuable content information is also included as a factor in making recommendations for re-ranking candidate products.

[Byron L.D.B, Francisco D.A.T.C and Valmir M.F, 2006][30] the process of getting user personal data may vary in many different ways, and can be done implicitly (through actions) or explicitly (through rates). After tracing actions or getting rates of the user, Computational Recommendation Technologies use information filtering techniques to recommend items. They described an approach to improve the recommendation quality in the first moments the user interacts with the system. The main idea is: (1) first of all, we describe the items with the general users opinion about them; and (2) after this, used modal symbolic structures to save this content in the user profile.

[Jun W, Arjen P. D. V and Marcel J.T. R, 2006][31] reformulates the memory-based collaborative filtering problem in a generative probabilistic framework, treating individual user-item ratings as predictors of missing ratings. The final rating is estimated by fusing predictions from three sources: predictions based on ratings of the same item by other users, predictions based on different item ratings made by the same user, and, third, ratings predicted based on data from other but similar users rating other but similar items.

[Michal A, Michael E and Alfred B, 2006][32] described, in recent years there has been a growing interest in the study of sparse representation of signals. They propose a novel algorithm for adapting dictionaries in order to achieve sparse signal representations. Given a set of training signals, authors seek the dictionary that leads to the best representation for each member in this set, under strict sparsity constraints and presented a new method—the K-SVD algorithm generalizing the K-means clustering process. K-SVD is an iterative method that alternates between sparse coding of the examples based on the current dictionary and a process of updating the dictionary atoms to better fit the data.

[Xiaoyuan S, Taghi M. K, 2006][33] considered as one of the most successful recommender systems, collaborative filtering (CF) algorithms can deal with high sparsity and high requirement of scalability amongst other challenges. Bayesian belief nets (BNs), one of the most frequently

used classifiers, can be used for CF tasks. In this work, they applied advanced BNs models to CF tasks instead of simple ones, and work on real-world multi-class CF data instead of synthetic binary-class data. Empirical results show that with their ability to deal with incomplete data, extended logistic regression on naïve Bayes and tree augmented naïve Bayes models consistently perform better than the state-of-the-art Pearson correlation-based CF algorithm.

[Alexandros N, Apostolos N. P, Yannis M and Tatjana W-D, 2007][34] concerned with the development of novel classification algorithms that can efficiently handle noise. To attain this, authors recognized an analogy between k nearest neighbor (kNN) classification and user-based collaborative filtering algorithms, as they both find a neighborhood so similar past data and process its contents to make a prediction about new data. The recent development of item-based collaborative filtering algorithms, which are based on similarities between items instead of transactions, addresses the sensitivity of user-based methods against noise in recommender system for this reason and focus on the item-based paradigm, compared to kNN algorithms to provide improved robustness against noise for the problem of classification, then proposed two new item-based algorithms, which are experimentally evaluated with kNN Classification.

[Arkadiusz P, 2007][35] developed a key part of a recommender system is a collaborative filtering algorithm predicting users' preferences for items. They describe different efficient collaborative filtering techniques and a framework for combining them to obtain a good prediction. The set of predictors used includes algorithms suggested by Netflix Prize contestants: regularized singular value decomposition of data with missing values, k-means, post processing SVD with KNN. They extending the set of predictors with the following methods: addition of biases to the regularized SVD, post processing SVD with kernel ridge regression, using a separate linear model for each movie, and using methods similar to the regularized SVD, but with fewer parameters. All predictors and selected 2-way interactions between them are combined using linear regression on a holdout set.

[Hyung J.A, 2007][36] presented a hybrid recommender system using a new heuristic similarity measure for collaborative filtering that focuses on improving performance under cold-start

conditions where only a small number of ratings are available for similarity calculation for each user.

[J. Ben S, Dan F, Jon H and Shilad S (2007][37] one of the potent personalization technologies powering the adaptive web is collaborative filtering. CF technology brings together the opinions of large interconnected communities on the web, supporting filtering of substantial quantities of data. They introduced the core concepts of collaborative filtering, its primary uses for users of the adaptive web, the theory and practice of CF algorithms, and design decisions regarding rating systems and acquisition of ratings and also discussed how to evaluate CF systems, and the evolution of rich interaction interfaces and closed the chapter with discussions of the challenges of privacy particular to a CF recommendation service and important open research questions in the field.

[Laurent C, Frank M, and Marc B, 2007][38] reviewed the main collaborative filtering methods proposed in the literature and compare them on the same widely used real dataset called MovieLens, and using the same widely used performance measure called Mean Absolute Error (MAE).

[Leo I, Anna L.G, Pasquale L, Marco D.G and Giovanni S, 2007][39] proposed an Electronic Performance Support System (EPSS) introduces challenges on contextualized and personalized information delivery. The main contribution is a content collaborative hybrid recommender which computes similarities between users relying on their content based profiles, in which user preferences are stored, instead of comparing their rating styles.

[Lijuan Z ,Yaling W, Jiangang Q and Dan L, 2007] [40] puts forward an improved collaborative filtering recommendation algorithm. Authors improved it in two aspects: First, they brought in a coefficient to coordinate the problem of inexact finding and falling recommendation quality which is caused by the fewer items when weighting the user similarity. Second, they collected the users' interest words implicitly when build the user interest model. At last, developed an online network bookshop as an example, tested and analyzed the three algorithms.

[Mandic .D, Vayanos. P, Boukis C, Jelfs. B, Goh S.L, Gautama T and Rutkowski T, 2007][41] proposed a novel stable and robust algorithm for training of finite impulse response adaptive filters is proposed. This is achieved based on a convex combination of the Least Mean Square (LMS) and a recently proposed Generalized Normalized Gradient Descent (GNGD) algorithm.

[Pingfeng L, Guihua N and Donglin C, 2007][42] proposed a hybrid approach combining semantic similarity with collaborative filtering is to generate the recommendation lists for users where the semantic similarity algorithm is adopted to get the nearest neighbors of the active user. The experiment results are presented which demonstrate that our approach is feasible.

[Robert M. B and Yehuda K, 2007][43] predicts Recommender systems based on collaborative filtering predict user preferences for products or services by learning past user-item relationships. The neighborhood-based approach leading to a substantial improvement of prediction accuracy, without a meaningful increase in running time. First, removed certain so-called “global effects" from the data to make the different ratings more comparable, thereby improving interpolation accuracy. Secondly showed how to simultaneously derive interpolation weights for all nearest neighbors. The interpolation weight is computed separately, simultaneous interpolation accounts for the many interactions between neighbors by globally solving a suitable optimization problem, also leading to improved accuracy.

[Xiaoyuan S, Russell G, Taghi M. K and Xingquan Z, 2007][44] proposed two hybrid CF algorithms, sequential mixture CF and joint mixture CF, each combining advice from multiple experts for effective recommendation. Hybrid CF models work particularly well in the common situation when data are very sparse. By combining multiple experts to form a mixture CF, our systems are able to cope with sparse data to obtain satisfactory performance.

[Akhmed U and Alexander T, 2008] [45] described an approach for incorporating externally specified aggregate ratings information into certain types of collaborative filtering (CF) methods. Theoretical insights gained from the analysis of this model-based method suggested a way to incorporate aggregate information into the heuristic item based CF method. Both the model-based and the heuristic item-based CF methods were empirically tested on several datasets, and

the experiments uniformly confirmed that the aggregate rating information indeed improves CF recommendations.

[Minchul Jung, Jehwan O and Eunseok L, 2008][46] describes the existing recommender systems generate recommendation usually using user's information previously collected. They proposed genetic recommend generating method for overcome this problem. Our method analyzes user`s real-time click-stream for grabbing user`s current intention, then uses genetic algorithm for generating appropriate recommendation.

[Saara H, Pauli M and Evimaria T, 2008][47] proposed two new matrix-decomposition problems: the nonnegative CX and nonnegative CUR problems, which give naturally interpretable factors. They extend the recently-proposed column and column-row based decompositions, and are aimed to be used with nonnegative matrices. Decompositions represent the input matrix as a nonnegative linear combination of a subset of columns (or columns and rows) from the input matrix.

[Shiqian M, Donald G and Lifeng C, 2008][48] focused on the linearly constrained matrix rank minimization problem is widely applicable in many fields such as control, signal processing and system identification. They proposed fixed point and Bregman iterative algorithms for solving the nuclear norm minimization problem and prove convergence of the first of these algorithms. By using a homotopy approach together with an approximate singular value decomposition procedure, get a very fast, robust and powerful algorithm that can solve very large matrix rank minimization problem and numerical results on randomly generated and real matrix completion problems demonstrate that this algorithm is much faster and provides much better recoverability than semi definite programming solvers such as SDPT3.

[Simon F, Yvonne H and Yang H, 2008][49] studied about Online recommenders are usually referred to those used in e-Commerce websites for suggesting a product or service out of many choices. Genetic algorithm (GA) is an ideal optimization search function, for finding a best recommendation out of a large population of variables. They presented a GA-based approach for supporting combined modes of collaborative filtering.

[Somnath B and Krishnan R, 2008][50] described about many real life datasets have skewed distributions of events when the probability of observing few events far exceeds the others. They observed that in skewed datasets the state of the art collaborative filtering methods perform worse than a simple probabilistic model. The test bench includes a real click stream dataset which is naturally skewed.

[Xiaoyuan S, Taghi M. K and Russell G, 2008][51] described as data sparsity remains a significant challenge for collaborative filtering (CF), they conjectured that predicted ratings based on imputed data may be more accurate than those based on the originally very sparse rating data. They proposed a framework of imputation-boosted collaborative filtering (IBCF), which first uses an imputation technique, or perhaps machine learned classifier, to fill-in the sparse user-item rating matrix, then runs a traditional Pearson correlation-based CF algorithm on this matrix to predict a novel rating.

[Yehuda K, 2008][52] there are two successful approaches to CF are latent factor models, which directly profile both users and products, and neighborhood models, which analyze similarities between products or users. They introduce some innovations to both approaches. The factor and neighborhood models can now be smoothly merged, thereby building a more accurate combined model. Further accuracy improvements are achieved by extending the models to exploit both explicit and implicit feedback by the users.

[Zhang L, Xiao B and Guo J (2008][53] proposed a hybrid approach to overcome the problem and defined a similarity weight to dealing with the data sparsity. Experimental results showed that their new approach can significantly improve the prediction accuracy of collaborative filtering.

[Antonina D, Felice F and Carlo T, 2009][54] considered that Web 2.0 applications innovated traditional informative services providing Web users with a set of tools for publishing and sharing information. The task of finding neighbors is difficult in environment such as social bookmarking systems, since bookmarked resources belong to different domains.

[Arno B, Elth O and Maarten V.S, 2009][55] predicts, the collaborative filtering (CF) algorithms used for recommendation operate on complete knowledge. They investigated how the well-known neighbourhood-based CF algorithm operates on partial knowledge; that is, how many similar users does the algorithm actually need to produce good recommendations for a given user, and how similar must those users be.

[DanEr C, 2009][56] discussed that the Collaborative filtering is one of the most successful technologies in recommender systems, and widely used in many personalized recommender areas with the development of Internet, such as e-commerce, digital library and so on. The K-nearest neighbor method is a popular way for the collaborative filtering realizations. Its key technique is to find k nearest neighbors for a given user to predict his interests. However, most collaborative filtering algorithms suffer from data sparsity which leads to inaccuracy of recommendation. Aiming at the problem of data sparsity for collaborative filtering, a collaborative filtering algorithm based on BP neural networks is presented.

[DeJia Z (2009][57] explained the development of recommender systems, the magnitudes of users and items grow rapidly, resulted in the extreme sparsity of user rating data set. Sparsity of users' ratings is the major reason causing the poor quality. To address this issue, an item-based collaborative filtering recommendation algorithm using slope one scheme smoothing is presented. Their approach predicts item ratings that users have not rated by the employ of slope one scheme, and then uses Pearson correlation similarity measurement to find the target items' neighbors, lastly produces the recommendations.

[Fuguo Z, 2009][58] examined the robustness of our topic-level trust-based recommendation algorithm that incorporate topic level trust model into classic collaborative filtering algorithm under the reverse bandwagon attack. The results of their experiments shows that topic-level trust based Collaborative Filtering algorithm offers significant improvements in stability over the standard k-nearest neighbor approach when attacked.

[HengSong T and HongWu Y, 2009][59] proposed a collaborative filtering recommendation algorithm based on the item classification to pre-produce the ratings. This approach classifies the

items to predict the ratings of the vacant values where necessary, and then uses the item-based collaborative filtering to produce the recommendations.

[Hyeong-J K and Kwang-S H (2009][60] predicts that cold-start problem is a primary factor causing performance loss in collaborative filtering and examined a fatal flaw of existing similarity measures in the coldstart condition and proposed a novel method, MSRV, using the moment of a random variable to solve the weaknesses of existing similarity measures that contain vector cosine similarity and correlation analysis-based methods.

[Hyeong-J K, Tae-H L and Kwang-S H, 2009][61] assessed the accuracy of predicting the user preference score is the most important element of collaborative filtering. They proposed novel similarity measures using difference score entropy of common rating items between two users. The proposed similarity measures can apply various weights according to the score difference, to evaluate the similarity.

[Hyoung D.K (2009][62] explained that in collaborative filtering, many neighbors are needed to improve the quality and stability of the recommendation. They proposed a consistency definition, rather than similarity, based on information entropy between two users to improve the recommendation. This kind of consistency between two users is then employed as a trust metric in collaborative filtering methods that select neighbors based on the metric.

[Jian-guo L and Bing-Hong W, 2009][63] proposed a spreading activation approach for collaborative filtering (SACF) by using the opinion spreading process, the similarity between any users can be obtained. The algorithm has remarkably higher accuracy than the standard collaborative filtering using the Pearson correlation. Furthermore, they introduce a free parameter to regulate the contributions of objects to user–user correlations.

[Jinbo Z, Zhiqing L, Bo X and Chuang Z, 2009][64] Proposed an optimized collaborative filtering recommendation algorithm based on item. While calculating the similarity of two items, they obtained the ratio of users who rated both items to those who rated each of them. The ratio is taken into account in this method.

[Mohammed N, Jenu S and Geun-Sik J, 2009][65] predicts each CF methods has their own advantage, though individually they possess several limitations. In order to minimize the limitation, they developed a hybrid recommender system incorporating components from both methods. Their approach includes a diverse-item selection algorithm that uses a diversity metric to select the dissimilar items among the recommended items from collaborative filtering, which together with the input is fed into content-based filtering.

[Paul T.B, Noraswaliza A and Yue X, 2009][66] considered that Recommender systems offered personalization to online activities due to their ability to recommend products that are unknown to the user. Existing methods require large amounts of training data which highlights a scalability problem of collaborative filtering, namely, the trade-off between accurate estimation prediction and the time required to calculate them

[Ping S and HongWu Y, 2009][67] proposed an item based collaborative filtering recommendation algorithm using the rough set theory prediction. This method employs rough set theory to fill the vacant ratings of the user-item matrix where necessary. Then it utilizes the item based collaborative filtering to produce the recommendation. The experiments were made on a common data set using different filtering algorithms.

[Prakash R, Juan L and Kendall N, 2010][68] proposed an implementation of a complete distributed e-learning system based on Collaborative filtering (CF) method using Agent Oriented Programming (AOP). The system has Intelligent Collaborative Filtering Based Tutoring System (ICFTS) capabilities that allow contents, presentation and navigation to be adapted according to the learner's requirements. In order to achieve that development, two concepts were put together: multi-agent systems and data mining techniques (specifically, the ARM algorithm).

[Pu W and HongWu Y, 2009][69] discussed about predicting products a customer would like on the basis of other customers' ratings for these products has become a well known approach adopted by many personalized recommendation systems on the Internet. They proposed a personalized recommendation algorithm combining slope one scheme and user based collaborative filtering. This method employs slope one scheme technology to fill the vacant

ratings of the user-item matrix where necessary. Then it utilizes the user based collaborative filtering to produce the recommendation.

[Reza S, Pedram P, George T, and Jean-P. H, 2009][70] proposed a distributed mechanism for users to augment their profiles in a way that obfuscates the user-item connection to an untrusted server, with minimum loss on the accuracy of the recommender system. They relied on the central server to generate the recommendations. However, each user stores his profile offline, modifies it by partly merging it with the profile of similar users through direct contact with them, and only then periodically uploads his profile to the server.

[RuLong Z and SongJie G, 2009][71] analyses the scalable collaborative filtering using clustering technology. Their approach can be implemented in two ways. One is based on the user clustering technology and the other is based on the item clustering technology. There is also a hybrid method using the user clustering and item clustering or bi-clustering.

[Sanjog R and Ambuj M (2009][72] proposed an approach for creating attack models and explores the importance of target item and filler items in mounting effective shilling attacks and the attack strategies are based on intelligent selection of filler items. Filler items are selected on the basis of the target item rating distribution. The filler item strategies for both all-user attacks and in segment attacks and showed through experiments that their attack strategies are the most effective attack strategies against both user-based and item-based collaborative filtering systems.

[SongJie G and HongWu Y, 2009][74] explained that personalized recommender systems consists services that produce recommendations and are widely used in the electronic commerce. Many recommendation systems employ the collaborative filtering technology. To solve the scalability problem in the collaborative filtering, they proposed a personalized recommendation approach joins the user clustering technology and item based collaborative filtering. Users are clustered based on users' ratings on items, and each cluster has a cluster center.

[SongJie G and HongWu Y, 2009][75] aiming the problem of data sparsity for collaborative filtering, a new personalized recommendation approach based on BP neural networks and item

based collaborative filtering is presented. The proposed method uses the BP neural networks to fill the vacant ratings where necessary and uses item based collaborative filtering to form nearest neighborhood, and then generates recommendations.

[SongJie Gong, 2009][76] proposed a new collaborative filtering personalized recommendation algorithm is proposed which employs the user attribute information and the item attribute information. This approach combines the user rating similarity and the user attribute similarity in the user based collaborative filtering process to fill the vacant ratings where necessary, and then it combines the item rating similarity and the item attribute similarity in the item based collaborative filtering process to produce recommendations.

[SongJie G, 2009][77] proposed a personalized recommendation algorithm joining case-based reasoning and item-based collaborative filtering. At first, it employs case-based reasoning technology to fill the vacant ratings of the user-item matrix. And then, it produces prediction of the target user to the target item using item-based collaborative filtering. The recommendation algorithm combining the case-based reasoning and item-based collaborative filtering can alleviate the sparsity issue and can produce more accuracy recommendation than the traditional recommender systems.

[SongJie G, HongWu Y and HengSong T, 2009][78] proposed an approach that combines the advantages of these two kinds of approaches by joining the two methods- memory-based CF and model-based CF. Firstly; it employs memory-based CF to fill the vacant ratings of the user-item matrix. Then, it uses the item based CF as model-based to form the nearest neighbors of every item. At last, it produces prediction of the target user to the target item at real time.

[Wolfgang W, Johannes H and Vivian P (2009][79] predicts their experiences from integrating item-based collaborative filtering into the Web 2.0 site linkfun.net and discussed the necessary steps to implement the selected Slope One algorithm in our real world application. It was necessary to conduct performance optimization to allow for recommendations without any delays in page generation on our site. Firstly, significantly reduced the data model by including

only items similarities for pairs of items where both items been rated by at least k users. Secondly, they precomputed recommended items for users.

[X. Su and T. M. Khoshgoftaar, 2009][80] introduced CF tasks and their main challenges, such as data sparsity, scalability, synonymy, gray sheep, shilling attacks, privacy protection, etc., and their possible solutions and presents three main categories of CF techniques: memory-based, model based, and hybrid CF algorithms. From basic techniques to the state-of-the-art, they attempted to present a comprehensive survey for CF techniques, which can be served as a roadmap for research and practice in this area.

[YiBo R and SongJie G, 2009][81] proposed a collaborative filtering recommendation algorithm based on singular value decomposition (SVD) smoothing. Their approach predicts item ratings that users have not rated by the employ of SVD technology, and then uses Pearson correlation similarity measurement to find the target users' neighbors, lastly produces the recommendations. The collaborative filtering recommendation algorithm based on SVD smoothing can alleviate the sparsity problems of the user item rating dataset, and can provide better recommendation than traditional collaborative filtering algorithms.

[Yongjian F, Jianying M and Xiaofei R, 2009][82] evaluated a Rough set which is a new mathematical tool that deals with incomplete and uncertain knowledge; it can improve the classification accuracy because of its characteristics. A rough set-based clustering collaborative filtering algorithm in e-commerce recommendation system is designed. They try to establish an classifier model based on rough set for the pre classification to items and give realization of clustering collaborative filtering algorithm and procedure of rough set algorithm, and carry on the analysis and discussion to this algorithm from multiple aspects.

[Zhang L, Xiao B and Guo J (2009][83] tells that current collaborative filtering based on clustering compute the whole set of items during the process of clustering or selecting nearest-neighbors, because the researchers believed if users have similar preferences on some of items, they will have the similar preferences on other items. They tried to propose a new collaborative filtering algorithm by using the localized preferences between users and design an algorithm

based on cluster model to find the localized preferences and then use the localized preferences between users to select neighbors for active users.

[Zibin Z, Hao M, Michael R. L and Irwin K, 2009][84] presented WSRec, includes a user-contribution mechanism for Web service QoS information collection and an effective and novel hybrid collaborative filtering algorithm for Web service QoS value prediction. To study the prediction performance, A total of 21,197 public Web services are obtained from the Internet and a large scale real-world experiment is conducted, where more than 1.5 millions test results are collected from 150 service users in different countries on 100 publicly available Web services located all over the world.

[A. Kumar, P. Thambidurai, 2010][85] described the survey of recent work in the field of web recommendation system for the benefit of research on the adaptability of information systems to the needs of the users. This issue is becoming increasingly important on the Web, as non-expert users are overwhelmed by the quantity of information available online, while commercial Web sites strive to add value to their services in order to create loyal relationships with their visitors-customers.

[David S, Montserrat B, Aida V and Karina G, 2010][86] proposed Modifications on classical similarity measures. They are based on a contextualized and scalable version of IC computation in the Web by exploiting taxonomical knowledge. The goal is to avoid the measures' dependency on the corpus pre-processing to achieve reliable results and minimize language ambiguity.

[Fuguo Zhang, 2010][87] examined the robustness of topic-level trust-based recommendation algorithm that incorporate topic-level trust model into classic collaborative filtering algorithm under the random attack. The results of their experiments show that topic-level trust based Collaborative Filtering algorithm offers significant improvements in stability over the standard k-nearest neighbor approach when attacked.

[Fuzhi Z, Sushi F, Dongyan J and Qing T (2010][88] evaluated an distributed recommender system, user information is stored on different nodes and proposed a DHT-based distributed collaborative filtering algorithm which uses the extreme ratings of users to generate the “fuzzy Key” for searching the similar neighbor information and introduces a weighting method to calculate the degree of similarity between users. When the weighting values are given, two factors, i.e. the appropinquity degree between the ratings of users and the inverse preference frequency of the ratings themselves, are taken into account.

[Hema B and Shikha M, 2010][89] presents Memetic Recommender System (MRS) based on the collaborative behavior of memes. Memetic Algorithms (MAs) are considered as one of the most successful approaches for combinatorial optimization. MAs are the genetic algorithms which incorporate local search in the evolutionary scheme. They a distinctive strategy to perform local search in mimetic algorithms. MRS works in 2 phases-In the first phase a model is developed based on Memetic Clustering algorithm and in the second phase trained model is used to predict recommendations for the active user.

[Jia R, Jin M, and Liu C, 2010][90] proposed a new similarity measure for clustering and its application. Firstly, they used a basic similarity function to discover neighbor vectors of items. Secondly, they calculated the cosine similarity of the neighbor vectors for clustering and thirdly, finished the clustering process by using adjusted DBSCAN. For those users having many known ratings, they adjusted the prediction function by adding a parameter which is the function of the item cluster size.

[Kimikazu K and Tikara H, 2010][91] discussed about computation of the collaborative technique, a singular value decomposition (SVD) is needed to reduce the size of a large scale matrix so that the burden for the next phase computation will be decreased. From their experiments, SVD means a roughly approximated factorization of a given matrix into smaller sized matrices. Webb showed an effective algorithm to compute SVD toward a solution of an open competition called "Netflix Prize". The algorithm utilizes an iterative method so that the error of approximation improves in each step of the iteration and give a GPU version of Webb's algorithm.

[Martín L-N, Yolanda B-F, Jose J. P-A and Rebeca P. D-R, 2010][92] addressed those problems by introducing a filtering strategy centered on the semantic properties that characterize the items and the users. Preliminary experiments are reported that prove the advantages of this strategy, especially in what concerns the treatment given to users with unique preferences and needs.

[Mustansar A.G, and Adam P-B, 2010][93] explained that Recommender Systems applied machine learning and data mining techniques for filtering unseen information and can predict whether a user would like a given resource. To date a number of recommendation algorithms have been proposed and suffer from scalability, data sparsity, over specialization, and cold-start problems resulting in poor quality recommendations and reduced coverage. Hybrid recommender systems combine individual systems to avoid certain aforementioned limitations of these systems.

[Mustansar A.G and Adam P-B, 2010][94] discussed about different types of CF techniques-collaborative filtering, content-based filtering, and demographic recommender systems. These systems suffer from scalability, data sparsity, and cold-start problems resulting in poor quality recommendations and reduced coverage. They proposed a unique cascading hybrid recommendation approach by combining the rating, feature, and demographic information about items.

[Prodan A, 2010][96] proposed one of the best methods for CF: the Matrix Factorization technique and described the implementation details of a framework created by them which uses Collaborative Filtering, shows some results obtained after experimenting with this framework.

[Qian W, Xianhu Y and Min S, 2010][97] proposed a hybrid user model, the recommender system based on this model not only holds the advantage of recommendation accuracy in memory-based method, but also has the scalability as good as model-based method. The user model is constructed based on item combination feature and demographic information, and it focuses on searching for set of neighboring users shared with same interest, which helps to improve system scalability. To enhance recommendation accuracy, each feature in user model is given a different weight when computing the similarity between users.

[Ryosuke F and Toshihiko W, 2010][98] proposed a fuzzy modeling approach for preference similarity model in collaborative filtering. In their approach, valid simplified fuzzy reasoning model is constructed through optimization of Mean Absolute Error. The model decides the weight of preference similarity from the value of correlation coefficient and the number of items.

[Sang H C, Young-S J, and Myong K. J, 2010][99] suggested hybrid methods improve the performance of recommendation algorithms. However, even though recent hybrid methods have helped to avoid certain limitations of CB and CF, scalability and sparsity are still major problems in large-scale recommendation systems. They proposed a novel hybrid recommendation algorithm HYRED, which combines CF using the modified Pearson's binary correlation coefficients with CB filtering using the generalized distance to boundary based rating. In the proposed recommendation system, the nearest and farthest neighbors of a target customer are utilized to yield a reduced dataset of useful information by avoiding scalability and sparsity problem when confronted by tremendous volumes of data.

[Teng-Kai F and Chia-Hui C, 2010][100] addresses the topic of social advertising, which refers to the allocation of ads based on individual user social information and behaviors. The allocation of advertisements based on both individual information and social relationships is becoming ever more important. In this study, they first proposed the notion of social filtering and compare it with content-based filtering and collaborative filtering for advertisement allocation in a social network. Second, they applied content-boosted and social-boosted methods to enhance existing collaborating filtering models and finally, an effective learning-based framework is proposed to combine filtering models to improve social advertising.

[WU Y and Tan X, 2010][101] presented a method- Singular Value Decomposition (SVD) is combined with hybrid collaborative filtering (CF) and proved to be an effective solution for sparsity problem. SVD is utilized in order to reduce the dimension of the user-page view matrix obtained from web usage mining. Afterwards, both low-rank matrices are employed in order to generate item-based and user based predictions. A framework for building automatic webpage recommendations in real-time platforms is designed.

[Xi C, Xudong L, Zicheng H, and Hailong S, 2010][102] presented RegionKNN- A novel hybrid collaborative filtering algorithm that is designed for large scale web service recommendation. Web service recommendations would be generated quickly by using modified memory-based collaborative filtering algorithm. Experimental results demonstrated that apart from being highly scalable, RegionKNN provides considerable improvement on the recommendation accuracy by comparing with other well known collaborative filtering algorithms.

[Xiao C.C, Run J.L and Hui Y C, 2010][103] a trust propagation model called TPM; proposes a hybrid index called TS index and a novel collaborative filtering recommendation algorithm called TPCF using TPM and TS index. The results of experiments using the dataset of Epinions.com, a popular ecommerce review website, show that TPCF is more attack resistant and improves the precision and coverage rate compared with the traditional collaborative filtering recommendation algorithm using Pearson's correlation coefficient.

[Yanhong G, Xuefen C, Dahai D, Chunyu L and Rishuang W, 2010][104] suggested that the traditional emphasize on user similarity may be overstated and there are additional factors having an important role to play in guiding recommendations. They proposed that trustworthiness of users must be an important consideration.

[Yehuda K, 2010][105] introduced a new neighborhood model with improved prediction accuracy. Unlike previous approaches that are based on heuristic similarities, the model neighborhood relations by minimizing a global cost function. Further accuracy improvements are achieved by extending the model to exploit both explicit and implicit feedback by the users. Past models were limited by the need to compute all pair wise similarities between items or users, which grow quadratically with input size.

[Zilei S and Nianlong L, 2010][106] presented their statistics and analysis on some recognized datasets. The analysis shows that the real rating features of the users cannot follow even distribution while most current algorithms were based on this premise. They proposed a new user-based collaborative filtering algorithm combining data-distribution.

[Zhaobin L, Wenyu Q, Haitao L and Changsheng X (2010][107] described one of the most important challenges may be due to the sparse attributes inherent to the rating data. Another important challenge is that existing CF methods consider mainly user-based or item-based ratings respectively. They proposed a P2P-based hybrid collaborative filtering mechanism for the support of combining user-based and item attribute-based ratings is considered. They took advantage of the inherent item attributes to construct a Boolean matrix to predict the blank elements for a sparse useritem matrix.

[Zhimin C, Yi J and Yao Z, 2010][108] proposed a improved Collaborative filtering algorithm. Firstly, the user's rating is given a weight by a gradual time decrease and credit assessment in the course of user similarity measurement and then several users highly similar with active user are selected as his neighbor. Finally, the active user's preference for an item can be represented by the average scores of his neighbor. Experimental results show that the algorithm can make the neighbor recognition more accurately and enhance the quality of recommendation system effectively.

[Xavier A, Alejandro J, Nuria O, and Josep M. P (2011][109] described and gave an overview of the main Data Mining techniques used in the context of Recommender Systems. authors first described common preprocessing methods such as sampling or dimensionality reduction. Next, they reviewed the most important classification techniques, including Bayesian Networks and Support Vector Machines. Then described the *k*-means clustering algorithm and discussed several alternatives and also presented association rules and related algorithms for an efficient training process.

Careful study of Literature indicated that this area became prominent for researchers experimenting with new methodologies for recommender systems and such a context generated motivation to the present research work.

The present work there by introduces certain modifications to some of the existing techniques so as to improve the quality and then investigate the suitability of them in terms of performance in applications.

2.2 Problem Identification

The problem statement, "**Some Studies on Personalized Recommendation Algorithms with Collaborative Filtering**", has been identified with an aim to evaluate several collaborative filtering algorithms and suggest the best based of the predictions and evaluation metrics. The proposed methods should have the properties like scalability, find good items, coverage should be maximum, and performance should not degrade with sparsity.

The amount of information in the world is increasing far more quickly than our ability to process it. All of us have known the feeling of being overwhelmed by the number of new books, journal articles, and conference proceedings coming out each year. Technology has dramatically reduced the barriers to publishing and distributing information. Now it is time to create the technologies that can help us sift through all the available information to find that which is most valuable to us. One of the most promising such technologies is collaborative filtering.

The main objective of this thesis is to minimize the drawbacks of various collaborative filtering recommendation systems and to improve the quality of various prediction algorithms, which were measured by comparing the predicted values for the withheld ratings to the actual ratings. Finally, based on the modifications and performance evaluation, some modified collaborative filtering recommendation systems were proposed which retain some of the best characteristics of the available methods and produce even better results.

CHAPTER III

MEMORY-BASED COLLABORATIVE FILTERING ALGORITHM BASED ON USER SIMILARITY USING PEARSON CORRELATION

The Huge amount of information and growing number of visitors to World Wide Web and the limitations of search engines create key challenges for recommendation systems. The question of finding eligible data seems to be increasingly harder without some information filtering or recommendation systems. Recommender system plays vital role in extracting additional value for a business from its user databases and help users find items they want to buy from a business. The significance in recommender Systems and its area is still remains high because it constitutes not only a problem-rich research area but also its abundance of practical applications that help users to deal with relevant information from the internet.

Memory based collaborative filtering technique is successful approach to build a recommender system uses the known preferences of a group of users to make predictions of the unknown preferences for other users. In order to make such predictions the Pearson correlation coefficient is considered for user similarity. User-based Collaborative Filtering is efficient when compared to k-Nearest Neighbor algorithm (k-NN) and Item-based collaborative filtering algorithms from the experiment results. In this Chapter a Memory based technique on user similarity using Pearson correlation coefficient is proposed and applied for Collaborative Filtering. The methodology using Pearson correlation coefficient used for predictions have been discussed. The Formulas that were used to implement these models including Pearson correlation coefficient, Weighted average rating, Simple weighted average and Prediction. The experimentation is done with MovieLens dataset which is available for research purpose provided by the GroupLens Research Project agency at the University of Minnesota. The measured Mean Absolute Error (MAE) of the proposed model are compared with available models from literature and finally the performance analysis is done based on parameter MAE.

3.1 Memory Based Collaborative Filtering Techniques

In this section the algorithms of Memory-based collaborative filtering (CF) systems are discussed and User based Collaborative filtering is proved efficient.

In Memory-based CF technique [1], User-based collaborative filtering systems depend on item rating predictions. The process of considering items to a user is based upon the opinions of people with similar likes or dislikes. Recommender systems helps to users to overcome information overload by providing personalized suggestions based on a history of a user's likes and dislikes. Memory-based CF techniques such as the User-based, Item-based CF techniques, Neighborhood-based CF computes similarity between users or items, and then uses the weighted sum of ratings or simple weighted average to make predictions based on the similarity values. Pearson correlation and vector cosine similarity are commonly used similarity calculations, which are usually conducted between co-rated items by a certain user or both users that have co-rated a certain item. To make top- N recommendations, neighborhood-based methods can be used according to the similarity values. Memory-based CF algorithms are easy to implement and have good performances for dense datasets. Memory-based CF algorithms drawbacks are dependence on user ratings, decreased performance when data are sparse, new users and items problems, and limited scalability for large datasets.

In this part of the work, the idea is to compute the active user's vote on a target item in terms of weighted average of the votes given to that item by other like-minded users. The existing user-based CF focused only on the user ratings, without considering the user accessing time for items. Since the user's demand can change over time and their ratings for different items would also change as their interests change. The existing recommendation system can hardly find the change, which will cause the system to deviate from the user needs for recommended resources. Thus the existing system cannot guarantee the reliability of user rating data. To solve this problems, in the present work it is proposed to improve the existing user-based algorithm by integrating the weight of user accessing time and the weight of reliability degree of user ratings, which would reflect the change of user interest in time and enhance the evaluation accuracy of user reliability. The Pearson correlation has been used as a similarity measure between users.

Memory-based collaborative filtering or neighborhood-based CF algorithm is one among the traditional collaborative filtering technique in recommender system which utilizes the entire or a sample of the user-item database to generate predictions. It evaluates the similarity between each user or item, generates nearest neighborhoods, and predicts preference scores with nearest neighborhoods. The evaluation of similarity is the most essential step, and the evaluated

similarity is used as a weight for predicting preference scores and as a measure for generating nearest neighborhoods. These systems utilize statistical techniques to find a set of users, known as neighbors that have a history of agreeing with the target user. Once a neighborhood of users is formed, these systems use different algorithms to combine the preferences of neighbors to produce a prediction for the active user. Memory based collaborative techniques are classified into three categories and they are:

1. Item-based collaborative filtering
2. k-NN collaborative filtering
3. User-based collaborative filtering.

3.1.1 Item-based Collaborative Filtering Algorithm

Item-based recommendation algorithms are meant for producing predictions to users with different approach looks into the set of items the target user has rated and computes how similar they are to the target item *i* and then selects *k* most similar items $\{i_1, i_2, \ldots, i_k\}$. At the same time their corresponding similarities $\{s_{i1}, s_{i2}, \ldots, s_{ik}\}$ are also computed. The prediction is then computed by taking a weighted average of the target user's ratings on these similar items when the most similar items are found. Similarity computation between two items *i* and *j* is to first isolate the users who have rated both of these items and then to apply a similarity computation technique to determine the similarity S_{ij}. Fig.4.1* illustrates this process, the matrix rows represent users and the columns represent items.

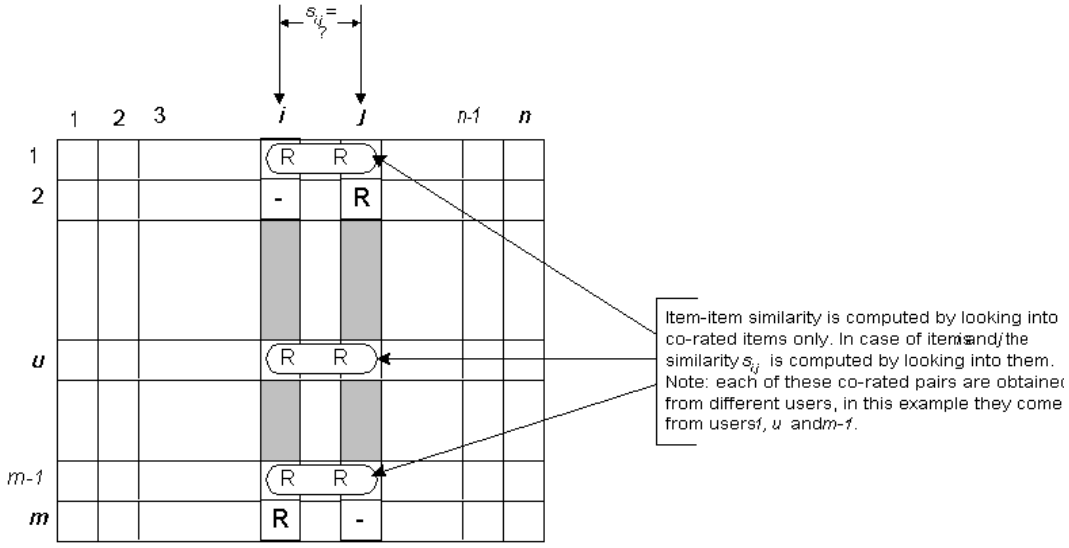

Figure 3.1: Isolation of the co-rated items and similarity computation

There are a number of different ways to compute the similarity between items. The three important methods are cosine-based similarity, correlation-based similarity and adjusted-cosine similarity. In which, correlation based similarity is the best and programmatically proved. Hence Pearson correlation similarity measure is used in this proposed model. The Prediction computation is the next step is to look into the target users' ratings and use a technique to obtain predictions. The weighted sum computes the prediction on an item i for a user u by computing the sum of the ratings given by the user on the items similar to i. Each ratings is weighted by the corresponding similarity $s_{i,j}$ between items i and j. N represents number of all similar items. The prediction $P_{u,i}$ can be noted as

$$P_{u,i} = \frac{\sum_N (S_{i,N} * R_{u,N})}{\sum_N (|S_{i,N}|)}$$

Basically, this approach tries to capture how the active user rates the similar items. The weighted sum is scaled by the sum of the similarity terms to make sure the prediction is within the predefined range.

3.1.2 k-NN based Collaborative Filtering

k-Nearest Neighbor (kNN) was commonly used in early CF-based systems. k is the most important parameter in a text categorization system based on k-Nearest Neighbor algorithm (k-NN). The predication can be made according to the category distribution among this k nearest neighbors after training set is determined. It consists of three major steps, namely user similarity weighting, neighbor selection, and prediction computation. The similarity weighting step requires all users in the database to be weighted according to their similarity with the active user. Similarities are reflected in the ratings that users have given items. In order for two particular users to be comparable, only items that both users have rated are counted. The neighbor selection step requires that a number of k-nearest neighbors of the active user be selected as item predictors. These selected users have the highest similarity weights and it is based on their interests and some partial information of the active user that an Item's prediction score is computed. k-NN is well known for its simplicity and prediction accuracy, which improves as the active user rates more items. Further, ratings based and content independent predictions allows k-NN to be used even in domains where textual descriptions of products are not available, not meaningful, or cannot be easily categorized by any attribute.

3.1.3 User Based Collaborative Filtering

User-based CF algorithm produces recommendation list for object user according to the view of other users. The assumptions are if the ratings of some items rated by some users are similar, the rating of other items rated by these users will also be similar. CF recommendation system uses statistical techniques to search the nearest neighbors of the object user and then basing on the item rating rated by the nearest neighbors to predict the item rating rated by the object user, and then produce corresponding recommendation list. Collaborative Filtering component that uses a neighborhood-based algorithm is a subset of users are chosen based on their similarity to the active user, and a weighted combination of their ratings is used to produce predictions for the active user.

In the user-based collaborative filtering recommendation system, the user ratings data are usually described as a user-item rating matrix R_{mxn} , in which m means the number of all users, n is the number of all items, and $R_{i,j}$ is the score of item j rated by user i , indicating the user's preference degree for the item. The most important step in the user-based CF is the searching of the target user's neighbor. Usually, the similarity is adopted as a means to measure the similar degree of user interests and hobbies through the common user ratings data. There are three main methods: cosine similarity, Pearson correlation coefficient similarity and the modified cosine similarity. Many experiments show that the Pearson correlation coefficient (PCC) can represent the similarity of users or items better than other similarity computation measures. Hence, it is adopted in our experiments.

3.1.4 User Based Collaborative Filtering- Efficient

The three main algorithms are implemented in JAVA under the classification of memory-based collaborative filtering for evaluating the prediction quality. The most commonly used metric for measuring the quality of recommendations is Mean Absolute Error (MAE). For Experimentation MovieLens dataset, which is a web-based research recommender system, has over 43000 users who have expressed opinions on 3500+ different movies and was developed by the GroupLens project at the University of Minnesota, is used to evaluate these algorithms.

3.1.4.1 Results of Item Based Collaborative Filtering

After computation of the predictions, the mean of the predictions of the active users and the actual ratings can be computed with mean absolute error (MAE) and the results are tabulated as follows.

Neighbor Set Size	4	8	12	16	20	25	28
MA E	2.95	2.93	2.89	2.86	2.85	2.85	2.76

Table 3.1: Nearest neighbor set and MEA on predictive validity

The influence of various nearest neighbors set on predictive validity is tested by gradually increasing the size of neighbors set. The evaluated results, MAE values and respective

their neighbor set sizes, are observed that the Nearest Neighbor Set value increases the corresponding MAE decreased.

3.1.4.2 Results of k-NN Based Collaborative Filtering

k-Nearest Neighbor Algorithm (kNN) is Recommender System which displays the results in terms of likeliest and unlikeliest of movie for the user ID's. The k parameter is the number of the closest neighbors in the space of interest is to be identified. Then, the computation of the distance between the query vector and all the objects in the training set is derived. Sorting of the distances for all the objects in the training set and determining the nearest neighbor based on the k^{th} minimum distance. Thereafter, all the categories of the training set for the sorted values falling under *k* are collected. The final step involves using the majority of the closest neighbours as prediction values.

The results for the percentage of likeliness using k-NN:

UID	Total movies	Like	Dislike
925	20	14	06
887	20	13	07
817	20	09	13
299	20	09	11
026	18	09	11
684	20	13	07
595	20	09	11
474	20	07	13
299	20	06	14
165	20	10	10
092	20	09	11
050	23	12	11
026	25	15	10
018	20	09	11

Table 3.2 Results of k-NN algorithm

3.1.4.3 Results of User Based Collaborative Filtering

After computation of the predictions, the mean of the predictions of the active users and the actual ratings can be computed with Mean Absolute Error (MAE) and the results are tabulated as follows.

Neighbour Set Size	4	8	12	16	20	25	28
MAE	2.29	2.48	2.56	2.56	2.57	2.57	2.57

Table 3.3: Nearest neighbour set and MAE on predictive validity

The influence of various nearest neighbours set on predictive validity is tested by gradually increasing the number of neighbours. It is observed that when Nearest Neighbour Set value increases the corresponding MAE also increased but the quality of prediction is increased when compared to the item-based collaborative filtering technique. Whereas, k-nearest neighbour algorithm (k-NN) is one of the best algorithm for Memory-based collaborative filtering which exhibits the results in terms of likeliest and unlikeliest of movie for the user ID's. The evaluation process of the k-NN algorithm is not derived the mean absolute error, which is the only evaluation metric used in this thesis for comparing the prediction quality. The performance of k-NN algorithm will be compared with the other content-based collaborative filtering technique to form a new model-based collaborative filtering which will be discussed in the next chapters. Mean while, the comparative analysis shows that user-based collaborative filtering performs well with respect to the item-based collaborative filtering. **Hence, the user-based collaborative filtering algorithm using is proposed for modification.** The Comparative of results analysis of implemented memory-based CF

Neighbour Set Size	4	8	12	16	20	25	28
MA E for Item-based CF	2.95	2.93	2.89	2.86	2.85	2.85	2.76
MA E for User-based CF	2.29	2.48	2.56	2.56	2.57	2.57	2.57

Table 3.4 Results of Item and User Based Collaborative Technique

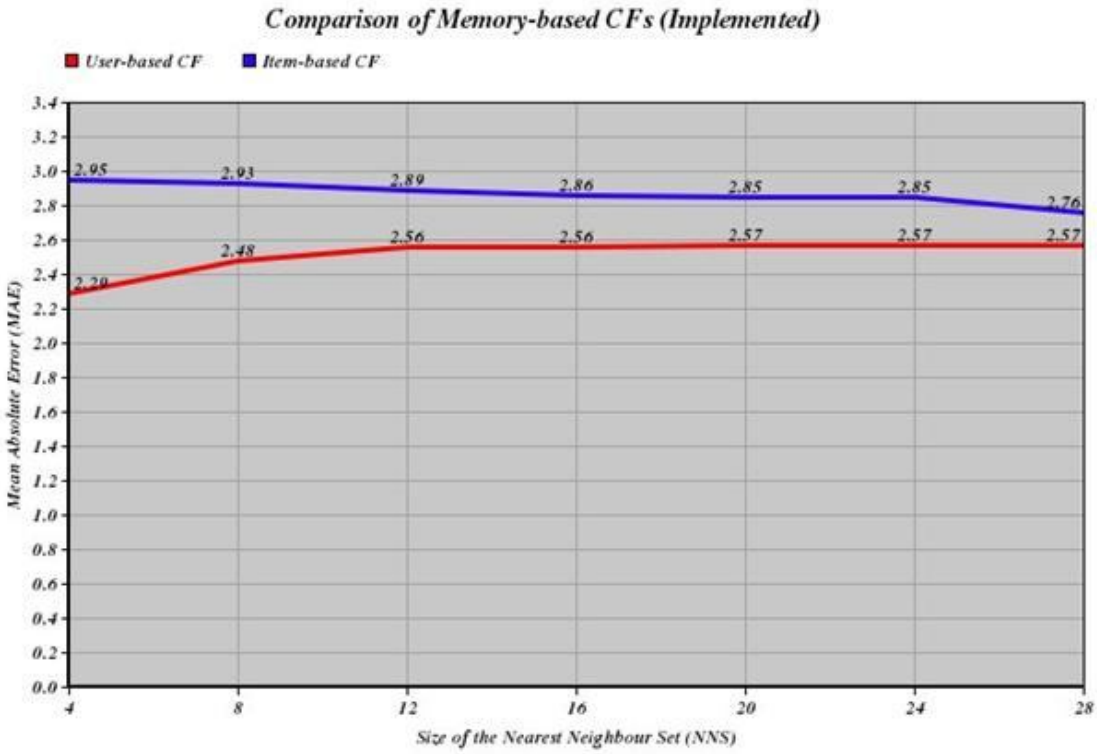

Fig. 3.2. Comparison of MAE for user-based collaborative filtering algorithm vs item-based collaborative filtering.

Findings:

However, the memory-based CF suffers from two basic problems: sparsity and scalability. Sparsity refers to the difficulty that most users rate only a small number of items. As a result, the accuracy of the method is often quite poor. As for scalability, memory-based approaches often cannot deal with the large numbers of users and items. Item-based filtering is significantly faster than user-based when getting a list of recommendations for a large dataset, but it does have the additional overhead of maintaining the item similarity table. Also accuracy difference depends on how it sparse the dataset.

The main advantage of the k-NN method is that it is effective when the datasets is large and robust where noisy data is used. This will help to address one of the main drawbacks of the naive technique with regard to reduced utility due to noise and redundancy. The naive method seeks to reduce the high error rate, especially with reference to increasing datasets. By calculating the distance of the objects from the nearest neighbours and using a threshold to get rid of outliers, it is hoped that only valid data will be fed into the naive k-NN algorithm consequently helping to weed out inconsistent data, and restricting the dataset to the minimum.

3.2 Proposed Model Description

3.2.1 Methodology of Memory-Based Collaborative Filtering Algorithm Based On User Similarity Using Pearson Correlation

In this model Pearson correlation coefficient (PCC) is used for the similarity of users or items.

In user-based collaborative filtering the predictions are computed as the weighted average of deviations from the neighbor's mean. In the modification process, a neighbourhood size is considered as a constant. It is common for the active user to have highly correlated neighbors that are based on very few co-rated items. These neighbors based on a small number of overlapping items tend to be bad predictors. The correlations based co-rated items are devalued by multiplying the correlation by a Significance Weighting factor then the resulting weighted sum will be decreased which is caused for improvement of the prediction quality.

It is proposed to introduce a coefficient is E which represents the number of neighborhood set in the intersection set that rated both by user i and j, the range of the coefficient is derived based the size of neighbourhood set. The condition that the users take part in majority rating and the rating items are almost the same can the user have the most possibility to become similar user. The users that take part in a few items rating, even though these rating are similar, in fact the users are not similar. In traditional similarity measurement method, large similarity could be acquired which is not accurate. After modify it and applied with a proportion coefficient E, the final value of weight factor becomes small; obviously the Mean Absolute Error (MAE) is decreased. The quality of the prediction is improved.

3.2.1.1 Similarity Computation

Similarity computation between items or users is a critical step in collaborative filtering algorithms. The basic idea of the similarity computation between two different items is first to work on the users who have rated both of these items and then to apply a similarity computation to determine the similarity between the two co-rated items of the users.

3.2.1.2. Correlation-Based Similarity

Similarity between two users and or between two items is measured by correlation-based similarities like the Pearson correlation or other one. Pearson correlation measures the extent to which two variables linearly relate with each other.

There are a number of different ways to compute the similarity between items. The three important methods are cosine-based similarity, correlation-based similarity and adjusted-cosine similarity. In which, correlation based similarity is providing better results than others and widely used in the CF research community. The Pearson correlation between users and is the summations are over the items that both the users and have rated and is the average rating of the co-rated items of the user and measures the extent to which two variables linearly relate with each other. Hence, it is used to measure the similarity in the thesis for various collaborative filtering algorithms.

The Pearson correlation between users is

$$P_{a,u} = \frac{\sum_{i=1}^{rn} (r_{a,i} - \bar{r}_a) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{rn} (r_{a,i} - \bar{r}_a)^3 \times \sum_{i=1}^{rn} (r_{u,i} - \bar{r}_u)^3}}$$

Where $\mathbf{r}_{a,i}$ is the rating given to item i by user a;

and r_a is the mean rating given by user a.

$P_{a,u}$ is the similarity between users a and u.

m is the number of users in the neighborhood.

3.2.1.3. Evaluation Metric

Evaluation is one of most important parts in any experiment. It is important because it allows us to see whether the results we obtain are good enough or not. Furthermore evaluation will provide us with objective metrics about the real performance of our system. In our specific case, an empirical evaluation will be determined to produce better recommendations to the users. However, evaluation in recommendation systems is a difficult task. One of the reasons that make this task difficult is unique and typical drawbacks. Furthermore, similarity is such a subjective phenomenon that can vary not only among users, but also across time or according to the mood or context or the context of the user. For this reason it is hard to establish a ground truth that could be used to evaluate systems. Since similarity is quite subjective first it is necessary to define some ways to measure it.

The evaluation of the results will be evaluated by Mean Absolute Error (MEA), which is a measure to calculate quality of predictions with similarity mean of user's ratings. Characteristics of MEA as mentioned in clearly in Introduction.

3.2.1.4. Data Sets:

One of the most important components of evaluation to perform the experiments is dataset which is a constant and easy to reproduce. The aim of this work is to provide recommendation using collaborative filtering techniques which needs data files and information about user preferences which are readily available with the datasets chosen. Throughout the present work two different datasets are chosen to provide recommendations. The first dataset is Jester; a web based online joke recommendation system, which has been developing at University of California, Berkeley. This data has 17,998 users collected with a rating from -10 to +10. The second dataset MovieLens is a web-based research recommender system. Each week hundreds of users visit MovieLens to rate and receive recommendations for movies. The site now has over 43000 users who have expressed opinions on 3500+ different movies. MovieLens was developed by the GroupLens project at the University of Minnesota.

Name	Users on a particular dataset	Items	Ratings	Ratings/User	Density	Rating Scale
MovieLens	943	1682	100,000	106.0	0.0630	1 to 5, integer
Jester	17,998	100	908,312	50.5	0.5048	-10.00 to 10.00,real

Table 3.5: Summary of datasets.

3.2.1.5. MovieLens Dataset:

MovieLens is a web-based research recommender system that debuted in fall 1997. Each week hundreds of users visit MovieLens to rate and receive recommendations for movies. The site now has over 43000 users who have expressed opinions on 3500+ different movies.

MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota. This data set consists of:

- 100,000 ratings (1-5) from 943 users on 1682 movies.
- Each user has rated at least 20 movies.
- Simple demographic info for the users (age, gender, occupation, zip)

Detailed description of the data files: these are the compressed tar files which are to rebuild the u.data files. ml-data.tar.gz , gunzip ml-data.tar.gz, tar xvf ml-data.tar and mku.sh

u.data: The full u data set which is of having100000 ratings by 943 users on 1682 items. Each user has rated at least 20 movies. Users and items are numbered consecutively from 1. The data is randomly ordered. This is a tab separated list of user id | item id | rating | timestamp. The time stamps are UNIX seconds since 1/1/1970 UTC.

u.info : The number of users, items, and ratings in the u data set.

u.item : Information about the items (movies); this is a tab separated list of movie id | movie title | release date | video release date | IMDb URL | unknown | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi |Thriller | War | Western |

The last 19 fields are the genres, a 1 indicates the movie is of that genre, a 0 indicates it is not; movies can be in several genres at once. The movie ids are the ones used in the u.data data set. The dataset provides optional unaudited demographic data such as age, gender, and the zip code supplied by each person. For each movie, information such as the name, genre, and release date are provided. Finally, the dataset provides the actual rating data provided by each user for various movies. User ratings range from zero-to-five stars. Zero stars indicate extreme dislike for a movie and five stars indicate high praise.

3.2.1.6 Jester Dataset:

Jester is a web based online joke recommendation system, which has been developing at University of California, Berkeley. This data has 73,421 users collected with a rating from -10 to +10. 500 users are selected programmatically with complete rating to generate the results. It is assumed that the rating value to its round value. The rating analysis of the Jester data set represents the count of movies the user has rated. It consists of ratings on 100 jokes by almost 18,000 users, over 900,000 ratings in all. Each joke was rated immediately after being read by the user, using an image map with one extreme representing strong liking and the other strong dislike.

Rating	Number of ratings		
	Dataset-1	Dataset-2	Dataset-3
-10	38884	41152	17024
-9	63616	61715	25707
-8	57479	57392	25731
-7	61089	63532	29237
-6-	60251	60840	26130
-5	63076	58706	22855
-4	70114	64608	24408
-3	71575	67041	24682
-2	70439	67579	23358
-1	97837	90862	32498
0	132710	126058	43376

1	117352	109594	36960
2	126957	121264	41122
3	131434	119720	41842
4	125769	112320	38204
5	115442	105300	34965
6	110447	102930	36660
7	106411	102652	36138
8	92249	84014	28205
9	97277	91706	27751
10	47	8	59

Table 3.6: Rating for different datasets

While importing the ratings into our database it is noticed that over 900 of the ratings were outside the allowed range of –10.00 to +10.00. On the advice of Goldberg, these ratings were removed and not considered in our experiments. In the dataset missed ratings are represent as 99. It is assumed that the rating value to its round value.

3.2.2. Proposed Algorithm for Memory-Based Collaborative Filtering Algorithm Based On User Similarity Using Pearson Correlation

Input : set of items and average ratings.
Output: Prediction and MAE

Step 1: All users are weighted with respect to similarity with the active user.
Step 2: Similarity between users is measured as the Pearson correlation between their ratings vectors.

$$P_{a,u} = \frac{\sum_{i=1}^{rn} (r_{a,i} - \bar{r}_a) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{rn} (r_{a,i} - \bar{r}_a)^3 \times \sum_{i=1}^{rn} (r_{u,i} - \bar{r}_u)^3}}$$

Where $\overline{r}_a, \overline{r}_u$ are the average ratings for the user a and u on all other rated items. The summations are over all the users who have rated item i.

Step: 3. Select n active users that have the highest similarity.

Step: 4. Predictions are computed as the weighted average of deviations from the neighbor's mean.

$$P_{a,i} = E + \frac{\sum_{u=1}^{n} (r_{u,i} - \overline{r}_u) \times P_{a,u}}{\sum_{u=1}^{n} P_{a,u}}$$

Step: 5. Compute a predictions from a weighted combination.

3.3. Implementation of Proposed Model

The implementation of the proposed model is done using JAVA. The description of implementation process is as follows:

NBSSimblanceRow objects each one containing its row number and its rating with the 1st row. *Probability.java* Probability class's constructor will calculate the probability of a class (1-5) of the given a document given a user. *XYSplineRendererDemoTest.java*: this class is responsible for generating the graph for exposed neighbour set size on X-axis and its corresponding MAE values on Y-axis. *CBA5.java:* this class is responsible for generating the MAE values.

CFA3.java is generating the MAE values for user-based collaborative filtering algorithm. It achieved through the steps Step 1, Step2, Step3 explained in the pseudo code algorithm which is mentioned earlier.

```
CFA3 we = new CFA3();

List original = new ArrayList();

String fileName2 = "D:\\Excelwork\\movielens-data-x.xls";

int requiredSize = 30;

we.populateExcelToList(original, fileName2, requiredSize);
```

In the above code snippet, the excel sheet data to an array list '*original'* is generated and the implementation of above function could be done in *populateExcelToList* method. Here '*fileName2'* is the path to the excel sheet we are testing. '*requiredSize'* is the number of users tested assumed as 30.

```
List sheetData = new ArrayList();
```

'sheetData' is an array list which will hold the randomly picked values from the 'original' data.

```
we.initialize(sheetData, requiredSize);
```

In the above line we are initializing 'sheetData' with all zeros

```
int[] randoms = we.getRandomUsers(original);
int[] lines = new int[randoms.length];
int[] columns = new int[randoms.length];
    Arrays.sort(randoms);
    int counter = 0;
    for(int d : randoms){
        lines[counter] = d/100;
        columns[counter] = d%100;
        counter++;
    }
we.getRandomList(lines, columns, original, sheetData);
```

getRandomList function generated 500 random numbers from the dataset numbers which rating is to be picked up by using the division with 30 (number of users). In this process, 'lines' represents the users and 'columns' represents the items. Lines[i] and columns[k] together would represent ith user's rating on k item. we.getRandomList(lines, columns, original, sheetData) is shuffled the randomly picked 500 ratings from 'original' list to 'sheetData'

```
NBSSimblanceRow[] nbsSimilarRows = null;
NBSSimblanceRow oNBSSimblanceRow = null;
ArrayList<NBSSimblanceRow[]>listSimblances=new
ArrayList<NBSSimblanceRow[]>();
for(int i=0; i<30; i++){
nbsSimilarRows = new NBSSimblanceRow[30];
for(int j=0; j<30; j++){
oNBSSimblanceRow = new NBSSimblanceRow();
nbsSimilarRows[j] = oNBSSimblanceRow;
}
listSimblances.add(nbsSimilarRows);
}
```

Then initialize arraylist which holds the prediction.

```
we.populateRows(listSimblances,sheetData);
for(int i=0; i<30; i++){
nbsSimilarRows = listSimblances.get(i);
Arrays.sort(nbsSimilarRows, we.new MyComparator());
}
```

In the above lines, we are sorting the obtained semblances.

```
List sheetData1 = ((List) ((ArrayList) sheetData).clone());
Hashtable table = new Hashtable();
for(int i=4; i<30; i=i+4){
List s3 = we.populatePredictUJ(sheetData1, listSimblances, i);
```

```
List  s4 =  new ArrayList();
double mae = we.getMAE(sheetData, original, s3);
BigDecimalz1=new
BigDecimal(mae).setScale(2,BigDecimal.ROUND_HALF_UP);
mae = z1.doubleValue();
System.out.println(" For neighbourset size -- " + i +" MAE is " + mae);
table.put(new Double(i), mae);
}
```

sheetData is cloned to sheetData1 which is used to obtain s3 which contains the predicted values for non rated ratings. The method *populatePredictUJ* is used for implementation for getting the predicted values. The parameters sent to this method are *sheetData1(*randomly picked ratings arraylist), *listSimblances(* its calculated and sorted semblances list) and i(size of neighbour set).

Once predicted values set is obtained, MAE values are calculated using the *getMAE* method for which the parameters are *original (*actual data from excel sheet), *sheetData* (ramdomly selected 500 ratings) and S3 (predicted ratings) are used.

3.4. Model Experimentation

To carry out the research and analysis for user-based collaborative filtering system, MovieLens dataset is used which is available for research purpose provided by the GroupLens Research Project agency at the University of Minnesota. The dataset consists of 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies. It provides demographic data such as age, gender, and the zip code supplied by each person. The content of the information of every movie is considered as a set of slots. Each slot is represented by number of words. Further, the data has been segregated and discarded for having less than 20 ratings or in complete demographic information. A subset of the ratings data from the MovieLens data set used for the purposes of comparison. 20% of the users were randomly selected to be the test users. The data

sets u1.base and u1.test through u5.base and u5.test are 80%/20% splits of the u data into training and test data. Each of u1, u2, u3, u4, and u5 has disjointed test sets for cross validation. These data sets can be generated from u.data by mku.sh.

The influence of various nearest neighbors set on predictive validity is tested by gradually increasing the number of neighbors. User-based collaborative filtering (UBCF) predicts item rating of the users are evaluated as per the opinions of the users chosen ratings. The results are shown in graphically representing MAE values and respective their neighbor set sizes.

3.5 Results and Discussions

The MAE values are computed using existing user-based collaborative filtering (UBCF) and modified UBCF for U1.test, U2.test, U3.test, U4.test and U5.test for test dataset and tabulated in table 4.5 to table 4.9.

The Comparative analysis of these computed values are presented.

a) MAE values for UBCF on U1.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for UBCF existing	2.61	2.62	2.62	2.62	2.62	2.62	2.62
MAE for UBCF Proposed Model	1.37	1.37	1.37	1.37	1.37	1.37	1.37

Table 3.7: MAE values for different neighbor sets for CF on **u1.test**

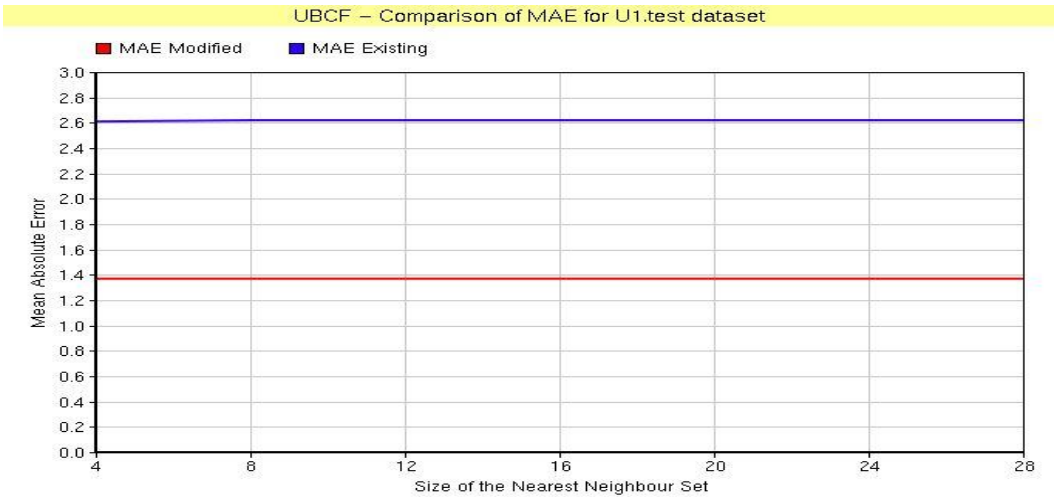

Fig. 3.3 Comparison of MAE for user-based collaborative filtering (UBCF) algorithm vs proposed algorithm on the U1.test.

MAE values derived based on prediction quality recommendations is generated, lower values of MAE indicate better performance. MAE is shown in as two graphical representations, the blue line, represents an existing user-based CF using Pearson correlation and the red line, and represents a modified algorithm, with lesser values than the existing.

b) MAE values for UBCF on U2.test dataset:

Neighbor Set Size	4	8	12	16	20	24	28
MAE Values of User-based CF existing	2.61	2.61	2.61	2.61	2.61	2.61	2.61
MAE Values of User-based CF Proposed Model	1.39	1.39	1.39	1.39	1.39	1.39	1.37

Table 3.8: MAE values for different neighbor sets for CF on **u2.test**

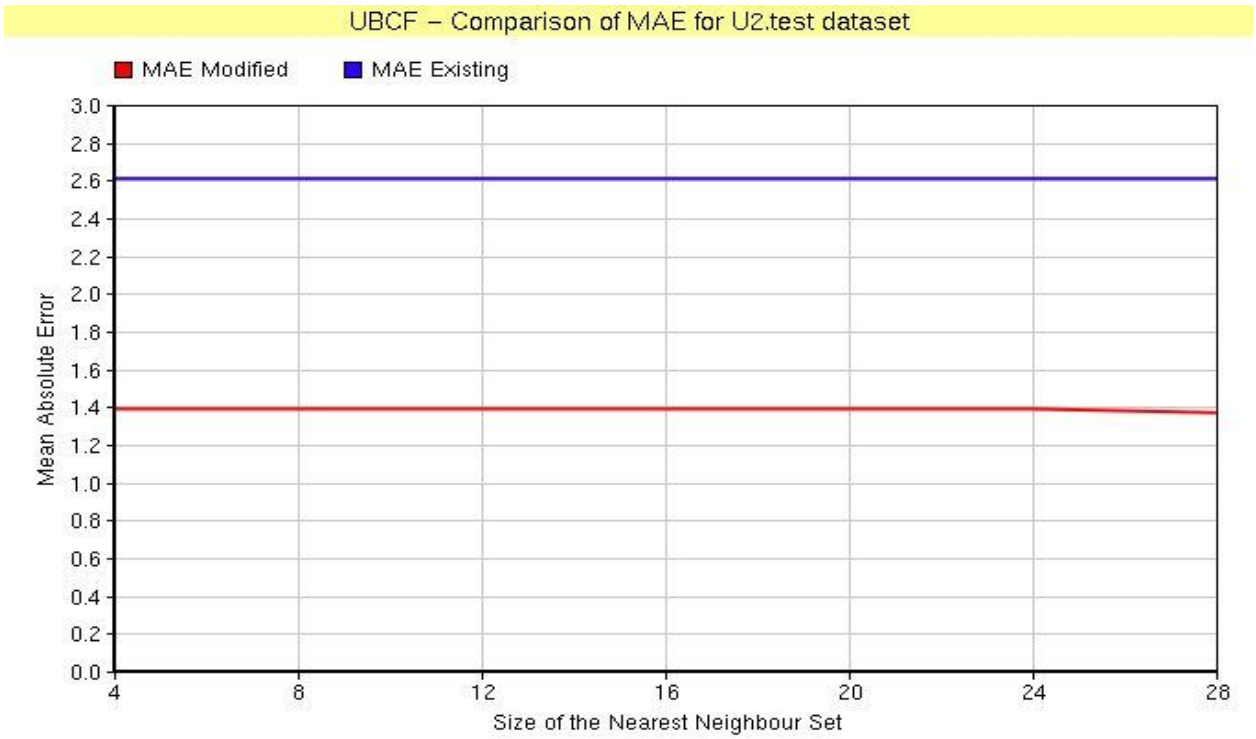

Fig. 3.4 Comparison of MAE for user-based collaborative filtering (UBCF) algorithm vs modified algorithm on the U2.test.

c) MAE values for UBCF on U3.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE Values of User-based CF existing	2.62	2.61	2.61	2.61	2.61	2.61	2.61
MAE Values of User-based CF Proposed Model	1.38	1.39	1.39	1.39	1.39	1.38	1.39

Table 3.9: MAE values for different neighbor sets for CF on **u3.test**

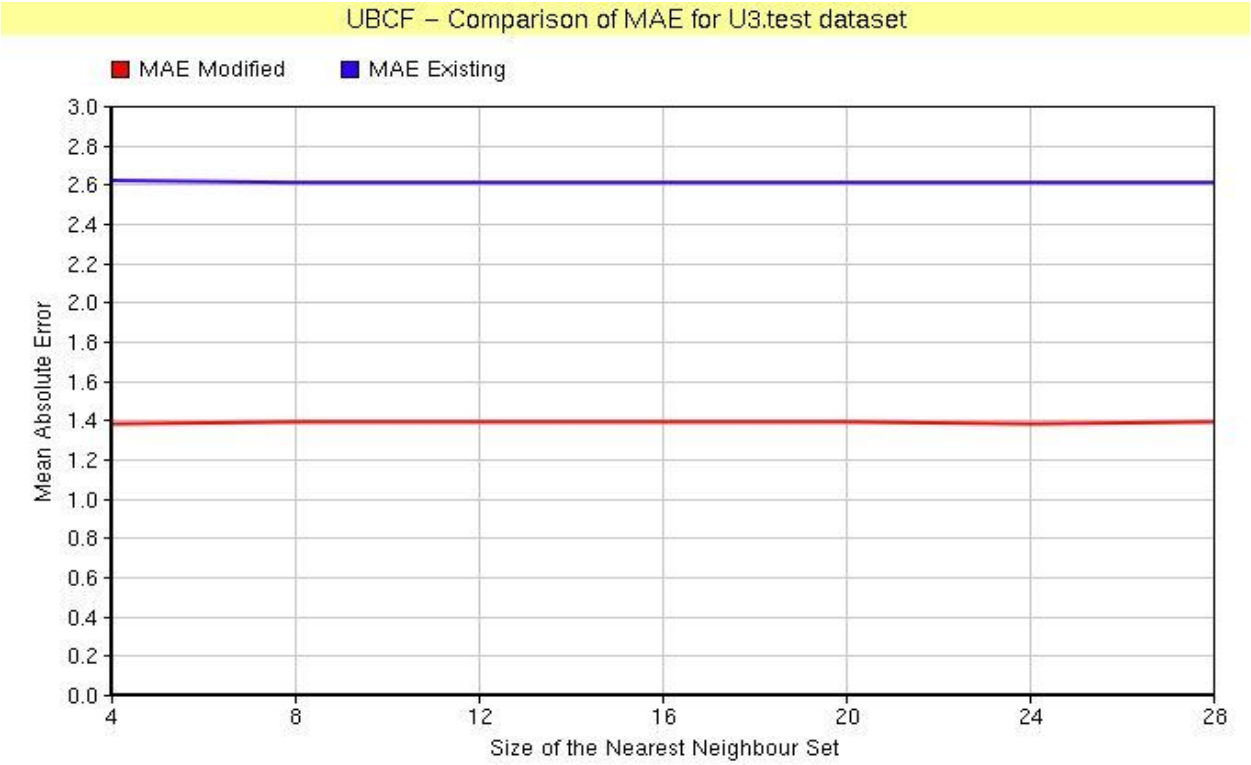

Fig. 3.5 Comparison of MAE for user-based collaborative filtering (UBCF) algorithm vs proposed algorithm on the U3.test.

d) MAE values for UBCF on U4.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE Values of User-based CF existing	2.21	2.54	2.56	2.56	2.54	2.54	2.54
MAE Values of User-based CF Proposed Model	1.39	1.39	1.39	1.39	1.38	1.37	1.37

Table 3.10. MAE values for different neighbor sets for CF on **u4.test**

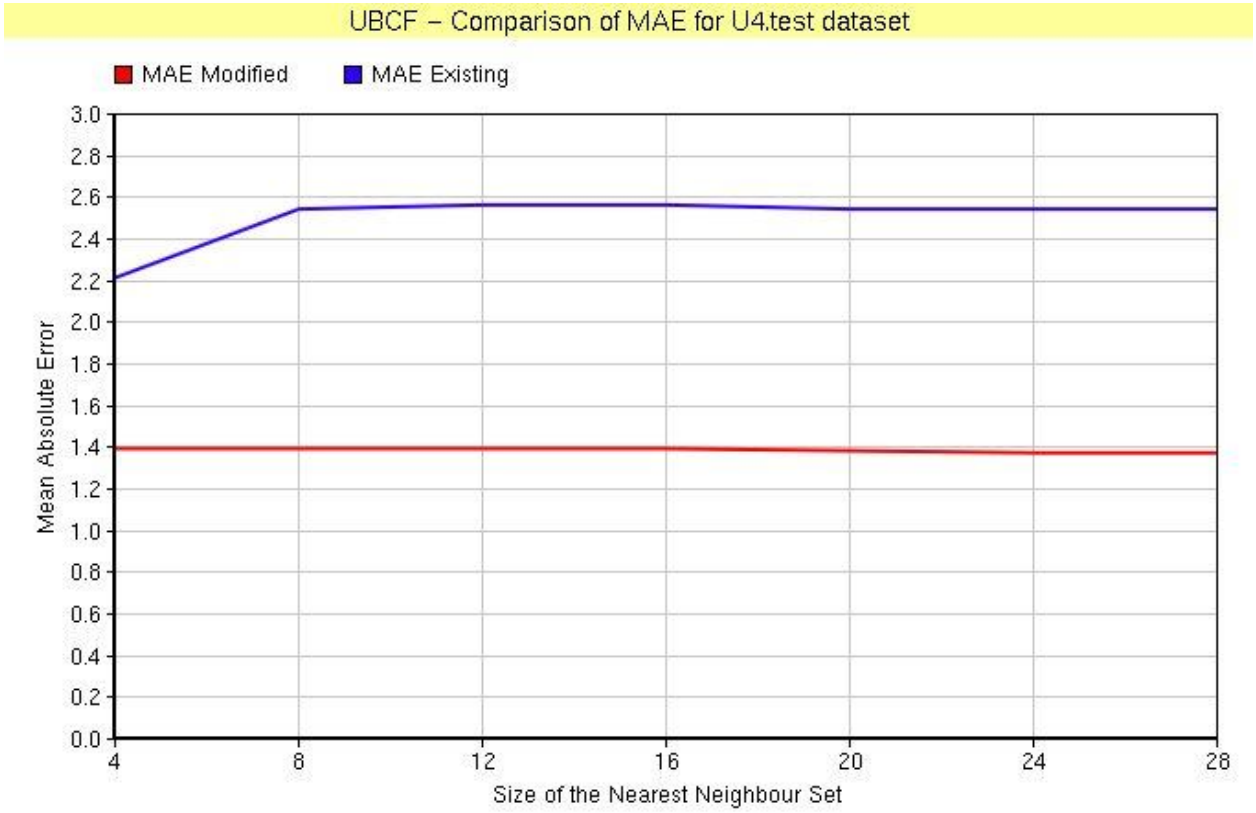

Fig 3.6: Comparison of MAE for user-based collaborative filtering (UBCF) algorithm versus proposed algorithm on the U4.test.

e) MAE values for UBCF on U5.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE Values of User-based CF existing	2.29	2.48	2.56	2.56	2.57	2.57	2.57
MAE Values of User-based CF Proposed Model	1.39	1.39	1.39	1.39	1.39	1.40	1.40

Table 3.11: MAE values for different neighbor sets for CF on **u5.test**

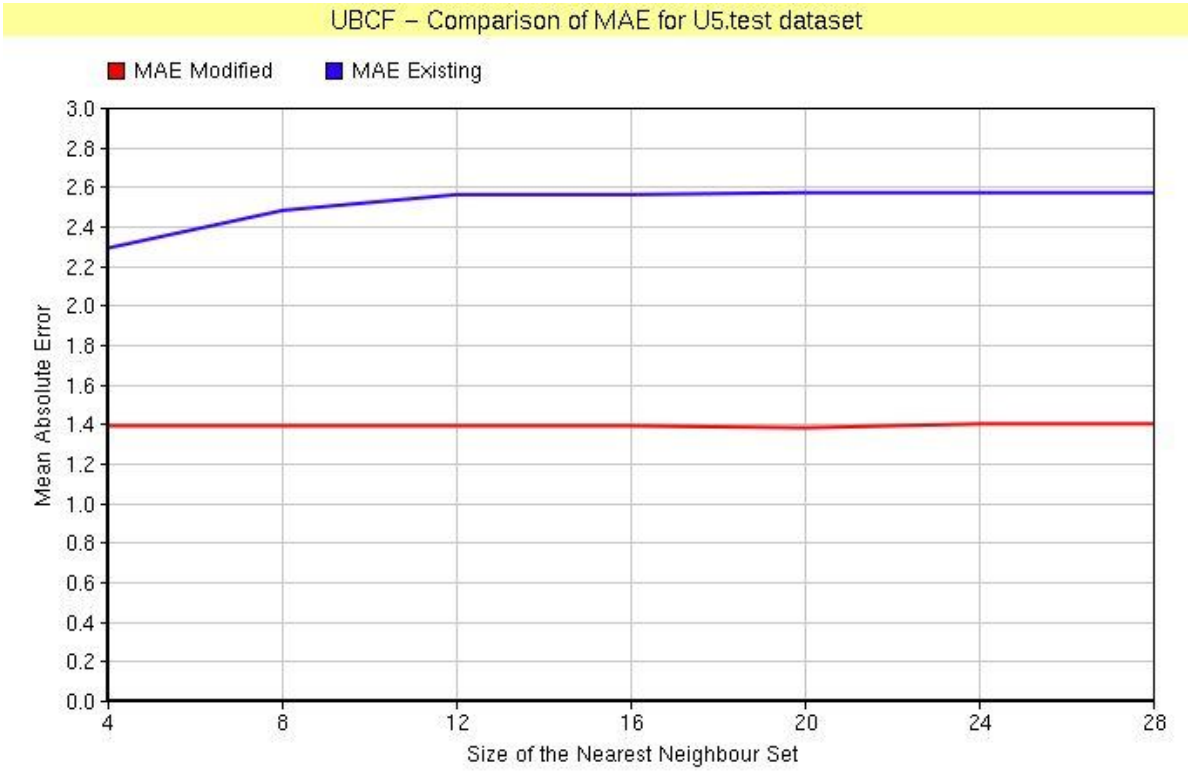

Fig. 3.7 Comparison of MAE for user-based collaborative filtering (UBCF) algorithm vs proposed algorithm on the U5.test.

The results presented in this chapter are given according to evaluation procedures with the experiments performed. The results for traditional user-based collaborative filtering and the proposed user-based collaborative filtering algorithm will be compared and presented. Derived

MAE values for different test datasets from U1.test to U5.test is related with recommendation accuracy which is computed and compared for the existing and modified methods to see which one performs better. MAE is obtained for every fold in our 5-fold cross validation experiment. Finally the total MAE was computed from the whole set of users and folds in the experiments.

The results presented in tables shows the MAEs for the different NNSs evaluation users using U1.test dataset performs 46.7% better improvement over existing UBCF. Whereas with U2.test dataset it is 47.5%, slightly increase is noticed. 47.1% improvement is noticed in case of the results performed with U3.test dataset and U4.test dataset. 39.3% improvement is noticed in U5.test dataset. It can be observe that both methods are the performed with unique performance in most of the cases, but the improvement of the prediction quality is decreased with increase of NNS. One thing that it is important to notice is the differences between the overall performances of the modified UBCF apparently perform much better than the existing UBCF.

Although UBCF presents a relatively good performance, it is expecting to obtain better results from it. The main reason for this relatively low performance is due to the correlations between users. Hence, it is proposed to use the modified UBCF for further experiments in this thesis in model-based collaborative filtering and hybrid collaborative filtering algorithm evaluations.

3.6. Conclusion

With an idea to compute the active user's choice on a target item as a weighted average of the ratings given to that item by other like-minded users, Pearson correlation was introduced to measure the similarity in the algorithm. Derived results show that the user-based collaborative filtering algorithm allows CF-based algorithms to scale to large data sets and at the same time produce high-quality recommendations. Here the user-based collaborative filtering algorithm (UBCF) is modified and tested its suitability.

CHAPTER IV

MODEL-BASED COLLABORATIVE FILTERING ALGORITHM BASED ON COMPOSITE PROTOTYPES

The amount of information and growing number of visitors to WWW and the ranges of search engines create vital role of challenges in recommendation systems. To find eligible data seems to be increasingly harder without some information filtering or recommendation systems. Recommender system plays vital role in extracting additional value for a business from its user databases and help users find items they want to buy from a business. The significance in recommender Systems and its area is still remains high because it constitutes not only a problem-rich research area but also its abundance of practical applications that help users to deal with relevant information from the internet.

A number of approaches which use Model-based Collaborative Filtering (MBCF) for scalability in building recommendation systems in web personalization have poor accuracy due to the fact that web usage data is often sparse and noisy. In this chapter the basic concepts of model-based collaborative filtering systems and the most popular algorithms-Apriori algorithm, Simple Bayesian CF Algorithm and Singular Value Decomposition algorithm techniques and there importance's are discussed. A new Model-Based Collaborative filtering algorithm based on Composite prototypes is proposed by introducing modifications in Singular Value Decomposition technique. The methodology using Composite prototypes used for predictions have been discussed. The Formulas that were used to implement these models including Rank determination, gradient, derivatives, Frobenius form and Prediction. The experimentation is done with MovieLens dataset which is available for research purpose provided by the GroupLens Research Project agency at the University of Minnesota. The measured Mean Absolute Error (MAE) of the proposed model is compared with available models from literature and finally the performance analysis is done based on parameter MAE.

4.1 Model-Based Collaborative Filtering Techniques

In this section the algorithms of Model-based Collaborative Filtering systems are discussed and Singular Value Decomposition (SVD) Collaborative filtering is proved efficient.

Model-based CF on imputed rating data and on dimensionality-reduced rating data will produce more accurate predictions than on the original sparse rating data. The development of models can allow the system to learn to recognize complex patterns based on the training data, and then make intelligent predictions for the collaborative filtering tasks for test data or real-world data, based on the learned models. Usually, classification algorithms can be used as CF models if the user ratings are categorical, and regression models and Singular Value Decomposition methods and be used for numerical ratings. Model-based CF techniques need to train algorithmic models, such as Bayesian belief nets, clustering techniques, Singular Value Decomposition or MDP-based ones to make predictions for CF tasks.

Model-based collaborative filtering algorithms based on composite prototypes allow the system to learn to recognize complex patterns based on the training data, and then make intelligent predictions for the collaborative filtering tasks for test data or real-world data, based on the learned models. Clustering, mining association rules, and sequence pattern discovery have been used to determine the access behavior model. Making use of some of the characteristics of the modeling process can provide significant improvements to recommendation effectiveness. Model-based CF algorithms have been investigated to solve the shortcomings of memory-based CF algorithms. Usually, classification algorithms can be used as CF models if the user ratings are categorical, and regression models and SVD methods and be used for numerical ratings. Model based collaborative techniques are classified into three categories and they are

1. Bayesian Belief Net CF Algorithm
2. Apriori Algorithm
3. Singular Value Decomposition Algorithm

4.1.1 Bayesian Belief Net CF Algorithms

A Bayesian network is a graphical representation of the joint probability distribution for a set of variables. The representation was originally designed to encode the uncertain knowledge of an expert. They also have become the representation of choice among researchers interested in uncertainty in Artificial Intelligence. Bayesian networks have formal probabilistic semantics which can serve as a natural mirror of knowledge structures in the human mind This facilitates the encoding and interpretation of knowledge in terms of a probability distribution, enabling inference and optimal decision making. A Bayesian network consists of two components. The first is a Bayesian belief net (BN) is a directed, acyclic graph (DAG) with a triplet N,A,O, where each node n ∈ N represents a random variable, each directed arc a ∈ A between nodes is a probabilistic association between variables, and O is a conditional probability table quantifying how much a node depends on its parents. This graph represents a set of conditional independence properties of the represented distribution: each variable is probabilistically independent of its non-descendants in the graph given the state of its parents. This graph captures the qualitative structure of the probability distribution, and is exploited for efficient inference and decision making. Thus, while Bayesian networks can represent arbitrary probability distributions, they provide computational advantage for those distributions that can be represented with a simple structure. The second component is a collection of local interaction models that describe the conditional probability of each variable X, given its parents. These two components represent a unique joint probability distribution over the complete set of variables.

The simple Bayesian CF algorithm uses a naive Bayes (NB) strategy to make predictions for CF tasks. Assuming the features are independent given the class, the probability of a certain class given all of the features can be computed, and then the class with the highest probability will be classified as the predicted class. For incomplete data, the probability calculation and classification production are computed over observed data. The Laplace Estimator is used to smooth the probability calculation and avoid a conditional probability of 0. Making the naive assumption that features are independent given the class label, the probability of an item belonging to class *j* given its *n* feature values, P ($class_j|f_1$, f2,…, fn) is proportional to:

$$P.\text{class} \prod P\ f \mid \text{class}$$

Where both P ($class_j$) and P ($f_1|class_j$) can be estimated from training data.
Here, two variants of the Simple Bayesian Classifier for collaborative filtering are transformed data model and sparse data model.

In Transform Data Model the features, even dual features ($U_i$like and $U_i$dislike), is completely independent. After selecting a certain number of features, absent or present information of the selected features is used for predictions, i.e. P($class_j$ | f_1=1, f_2=0,…, f_{n-1} = 1, f_n=0), Where f_1=1means that f_i is present on the target item and f_i=0 means that f_i is absent on the target item. When estimating conditional probabilities, e.g. P(f_i=1|$class_j$) , it is calculated overall ratings of the target user and some conditions must hold implement the model.

In Sparse Data Model, assumed that only known features are informative for classification. Therefore, only known features are used for predictions. Therefore the following formula is considered as P ($class_j$ | f_1=1, f_2=0,…, f_{n-1} = 1). Moreover, the only use of the data which both users in common rated when estimating conditional probabilities. In this representation, the following condition holds as P($U_i$like = 1| $class_j$)+P($U_i$like = 1 | $class_j$) = 1. Feature selection is a common preprocessing technique in many supervised learning algorithms. By restricting the number of features, it might be expected that it would increase the accuracy of the learner by ignoring irrelevant features or reduce the computation time.

4.1.2 Apriori algorithm

Association rules are "if-then rules" with two measures which quantify the support and confidence of the rule for a given data set. Having their origin in market basked analysis, association rules are now one of the most popular tools in data mining. This popularity is to a large part due to the availability of efficient algorithms. The first and possibly most influential algorithm for efficient association rule discovery is Apriori. Association rule mining and its

association rules can find out the predefined minimum support and confidence from a given database. The problem is usually decomposed into two subproblems. One is to find those itemsets whose occurrence exceeds a predefined threshold in the database; those itemsets are called frequent or large itemsets. The second problem is to generate association rules from those large itemsets with constrain of minimal confidence. Suppose one of the large item sets is L_k, $L_k=\{I_1,I_2,\ldots,I_{k-1}\}$, association rules with this itemsets are generated in the following way: the first rule is $\{I_1,I_2,\ldots,I_{k-1}\}=>\{I_k\}$,by checking the confidence this rule can be determined as interesting or not. The other rule are generated by deleting the last items in the antecedent and inserting it to the consequent, further the confidence of the new rules are checked to determine the interestingness of them.

4.1.3 Singular Value Decomposition (SVD)

The singular value decomposition (SVD) plays a vital role in numerical linear algebra and in many statistical techniques as well. Using two orthonormal matrices, SVD can diagonalizable any matrix A and the results of SVD can notify a lot about consequences of the matrix. A collaborative filtering deals with a large sized matrix which stands for customers and items. Basically, its mechanism is to compute which-is-near relationship of the column or row vectors. The features and the user preferences for these features are already embedded in the user-movie-rating triplets; combined together as a weighted sum. Singular Value Decomposition states that every matrix A_{mxn} can be decomposed as

$$A = USV^T,$$

Where U and V are orthogonal and S is diagonal with singular values of A on the diagonal.
U, S and V values are maximum in full singular value decomposition. Since S is a diagonal matrix, it can remove rows from S and columns from U to obtain a more compact representation where U is m x n, S is n x n, and V is n x n. The process is known as reduced singular value decomposition. Singular Value Decomposition has many applications, but of particular interest to us is the application of the SVD method to find a rank-r approximation to a given matrix.

4.1.4 Singular Value Decomposition (SVD)-Efficient

The three main algorithms are implemented in JAVA under the classification of memory-based collaborative filtering for evaluating the prediction quality. The most commonly used metric for measuring the quality of recommendations is Mean Absolute Error (MAE). For Experimentation MovieLens dataset, which is a web-based research recommender system, has over 43000 users who have expressed opinions on 3500+ different movies and was developed by the GroupLens project at the University of Minnesota, is used to evaluate these algorithms**.**

4.1.4.1 Results of Bayesian Belief Net CF Algorithms

Simple Bayesian Classifier Algorithm is implemented with two different models transform model and sparse model with two different datasets (Jester dataset is also used). The evaluation criteria for this are F-Measure which is described as follows.

The most accurate predictions of the exact rating which a user would have given to the target item are not involved. Rather we would like to have a system that can accurately distinguish items that are liked by the user and items disliked. To distinguish items, transformed numerical ratings into these two labels are considered. Not only does an assigning class label allow us to measure classification accuracy, we can also apply additional performance measures, precision and recall, commonly used for information retrieval tasks. However, it might be easy to optimize each of these measurements. To avoid this problem, we use F-Measure, which combines Precision and recall:

$$\text{F-Measure} = \frac{\cdot \qquad \cdot}{}$$

a. Transformed data model results

Number of Features	Classification of accuracy (%)
50	64.0
100	65.0
150	66.0
200	67.0
250	66.9
300	66.8
350	66.7
400	66.6
450	66.5
500	66.4
550	66.3

Table 4.1 classification of accuracy – Transformed data model

b. Sparse data model

Number of users	Classification of accuracy (%)
10	66.0
20	66.5
30	67.0
40	67.5
50	68.0
60	67.75
70	67.50
80	67.25

90	67.00
100	66.50
110	66.00
120	65.50
130	65.00
140	64.50
150	64.00
160	63.50

Table 4.2 Classification of accuracy – Sparse data model

By observing two representations for the Simple Bayesian Classifier, it can be found that the Sparse Data Model performs better than the Transformed Data Model, a typical correlation-based approach. The Transformed Data Model also outperforms the correlation-based approach although it shows similar accuracy to the correlation approach in some parts of the experiment with MovieLens dataset.

4.1.4.2 Results of Apriori algorithm

The accuracy of our association rule recommender is analyzed to compare with Apriori algorithm under memory-based collaborative filtering technique. However, the association rule algorithm produces a ranked list, such that the recommendation score is the confidence that a target user will like the recommended item. It is also not possible to make a prediction of the rating value from the association rule recommendation list. However, the association rule recommender does make a more general prediction; it predicts a binary “like” or “dislike” classification for a recommended item if the confidence value is positive or negative, respectively. Apriori selects recommendations from only among those item sets that have met the support threshold and it will by necessity have lower coverage than our baseline algorithms. The resulting association rules are

Rules	Support(XY)	Support(Y)	Confidence
{A}=>{C}	2.0	2.0	1.0
{C}=>{A}	2.0	3.0	0.6666666666666666
{B}=>{C}	2.0	3.0	0.6666666666666666
{C}=>{B}	2.0	3.0	0.6666666666666666
{B}=>{E}	3.0	3.0	1.0
{E}=>{B}	3.0	3.0	1.0
{C}=>{E}	2.0	3.0	0.6666666666666666
{E}=>{C}	2.0	3.0	0.6666666666666666
{B}=>{C E}	2.0	3.0	0.6666666666666666
{C E}=>{B}	2.0	2.0	1.0
{C}=>{B E}	2.0	3.0	0.6666666666666666
{B E}=>{C}	2.0	3.0	0.6666666666666666
{E}=>{B C}	2.0	3.0	0.6666666666666666

Table 4.3: Strong association rules from the frequent itemsets

4.1.4.3 Results of Singular Value Decomposition (SVD)

The MAE values are computed using existing Singular value decomposition (SVD) algorithm and modified SVD for test data sets are tabulated.

MAE values for SVD on given dataset

Neighbour Set Size	4	8	12	16	20	24	28
MAE for SVD	1.086	1.086	1.086	1.086	1.086	1.086	1.086

Table 4.4: MAE values for different neighbor sets datasets

From the above analysis and observations Singular Value Decomposition is then best method for prediction. That is the reason we try to make some modification in SVD approach using Composite Prototypes as the proposed model and results are compared with the existing SVD model in the next sections.

4.2 Proposed Model Description

4.2.1 Methodology of Model-Based Collaborative Filtering Algorithm Based On Composite Prototypes

Given a matrix R, Compute a rank-r R_{app} (approximation) to this matrix such that the Frobenius form of $R - R_{app}$ is minimized. Then, Frobenius form ($||R - R_{app}||F$) is defined as simply the sum of squares of elements in $R - R_{app}$. It can achieve such an approximation by only considering the first r most significant singular values in the singular value decomposition of R. Returning to our domain, it can formulate the problem as an **u x m** matrix R which contains the actual ratings by the users, where u is the number of users and m is the number of movies. Assume that consider f features, regarding the rest as insignificant. Compute an approximation R_{app} to this matrix R, such that $||R-R_{app}||F$ is minimized, and $R_{app} = P_{uxf}(F_{mxf})T$. Notice that the i^{th} row of P vector is the preference vector for user i, and the k^{th} row of F is the feature vector for movie k. Therefore we have extracted the an approximation to the desired data, which can then use to fill unknown entries of R by computing the dot product of user preference and movie feature vectors.

A set of features for each movie from MovieLens dataset is used, and a set of weights for each user indicating how likely a user is to enjoy a movie with such users. The task of predicting a rating then becomes simply multiplying the user preference vector with the movie feature vector. It is extremely difficult to collect such data. First of all, it is by itself a difficult task to compute & rate features for each movie due to the subjective nature of the task. Second, this would require retrieving information from external resources and combining it with the user-movie rating and this data would require tremendous cleaning-up effort.

4.2.2 Proposed Algorithm for Model-Based Collaborative Filtering Algorithm Based On Composite Prototypes

There are of course various modifications and adjustments to make this perform slightly better, including regularization, using different functions for rating prediction instead of just taking a dot product of the preference and feature vector, doing the rounding in a clever method, etc. Here, a constant 'K' is included to balance the P and F updates. The given 'K' constant can compute the required singular vectors because each singular vector is computed independently of each other. Similarly, SVD is capable of selecting the secular equation to be solved. This algorithm performs a little slow to converge as with all gradient descent algorithms, but the method performs well even in its raw form.

Algorithm:

Require: average ratings. the given ratings convert into a matrix of ratings R, Compute an approximate matrix R_{app} such that MAE is minimized

Step 1: Task: Find the best dictionary to represent the data samples as sparse compositions.

Step 2: Initialization: Set the dictionary matrix D. Set J = 1
Repeat until convergence .

Step 3: Sparse Coding Stage - Use any pursuit algorithm to compute the representation vectors.

Step 4: Update Stage- For each column k = 1; 2; : : : ;K in D

Step 5: Compute the overall representation error matrix.

Step 6: Restrict E by choosing only the columns corresponding to k**.**

Step 7: Apply SVD decomposition. Choose the updated dictionary column.

Step 8: Update the coefficient vector multiplied vectors

$MAE = ||R-R_{app}||F.$

Step 9: compute P as $US1^{/2}$ and F as $S^{1/2}V'^{T.}$

Step 10: minimize the error: $E = (R-R_{app})_{ij}^{2}$

Step 11: compute Pik (t+1) and $F_{jk}(t+1)$.

Take the derivative with respect to p_{ij} and f_{jk} and the updates become

$P_{ik}(t+1) = P_{ik}(t) + L*(R-Rapp)_{ij}* F_{jk}(t) – K* P_{ik}(t)$

$F_{jk}(t+1) = F_{jk}(t)+ L*(R-Rapp)_{ij} * P_{ik}– K* P_{ik}(t)$

4.3 Implementation of Proposed Model

The implementation of the proposed model is done using JAVA. The description of implementation process is as follows:

The main java classes designed and developed to evaluated the predictions for the SVD Filtering algorithms are *CBA5.java, NBSSimblanceRow.java, Probability.java XYSplineRendererDemoTest.java* . A segment of java code snippet and the structure of the java classes that implements the SVD Filtering algorithms proposed in the system are as follows.

```
List original = new ArrayList ();
String fileName2 = "D:\\Excelwork\\ml-data_0\\u.data";
int usersSize = 100;
```

```
int itemsSize = 1000;
we.initialize(original, usersSize, itemsSize);
we.populateFileToList(original, fileName2,  usersSize, itemsSize);
```

Here the List '*original*' is the list which contains the original ratings of the users which will be compared with the predicted ratings. It is designed to populate the list with the ratings read from the u.data file with the mentioned Path in the code.

```
List  test =  new ArrayList();
fileName2 = "D:\\Excelwork\\ml-data_0\\u5.test";
we.initialize(test, usersSize, itemsSize);
we.populateFileToList(test, fileName2,  usersSize, itemsSize);
```

The List '*test*' is the list which contains the test ratings of the users. Test data is the subset of original data. Using test data, it is designed to produce the user rating predictions and populating the 'test' list with values read from u5.test.

```
fileName2 = "D:\\Excelwork\\ml-data_0\\u.genre";
ArrayList  genre =  new ArrayList();
we.initializeGenre(genre, fileName2);
```

The List '*genre*' is the list of genre of the movies. Each movie genre is given a unique number which is used in item classification and populating the test list with values read from u5.test.

```
fileName2 = "D:\\Excelwork\\ml-data_0\\u.item";
List  items =  new ArrayList();
```

we.initializeItems(items, 1682, 30);

we.populateItemsToList(items, fileName2, 1682, 30, genre);

The List 'items' is the list of all the items that presented in u.item. 1682 is number of items given, and 30 is the number of properties mentioned in the u.item file. It is designed and developed to populate the test list with values read from u.item. All the properties are embedded in a child list and the child list is added to parent list.

Here a list s3 is generated, which contains user given ratings along with content boosted predicted values filled in the place of non given ratings.

```
for(int user=0; user<s3.size(); user++){
                currentUserData = (ArrayList)s3.get(user);

 for(int item=0; item<currentUserData.size(); item++){
        intRating = (Double) currentUserData.get(item);
 rating = intRating.doubleValue();
        if(rating == 0 ){
              rating = avgRating;
        }
                ratingsVector[user][item] = rating;
                 }
              }
```

Here *ratingsVector* has been generated, which is a double indexed array contains user ratings along with default values generated by SVD collaborative filtering algorithm.

```
Matrix A = new Matrix(ratingsVector);
SingularValueDecomposition svd = new SingularValueDecomposition(A);

int M = A.getRowDimension();
int N = A.getColumnDimension();
Matrix Ur1 = svd.getU();
Matrix Ur = Ur1.getMatrix(0, M-1, 0, M-1);
Matrix Vr1 = svd.getV();
Matrix Vr = Vr1.getMatrix(0, N-1, 0, M-1);
double[] s = svd.getSingularValues();
Matrix Sr = getS(s,M);
```

Here class *A* is matrix representation of *ratingsVector which is used to* calculate the left matrix, singular matrix and right matrix of the singular value decomposition of the Matrix A. Here *Ur* is the left matrix, *Sr* is singular matrix, *Vr* is right matrix of the singular value decomposition of the Matrix A.

```
Matrix aproxA = Ur.times(Sr).times(Vr.transpose());
int minRank =  calMinError(A, Ur, Sr, Vr, M);
```

Here *aproxA* is calculated using the formula . $A = USV^T$

minRank is calculated such that a rank-r approximation A' to A such that $A' = U'S'V'^T$ where U' is mxr, S' is rxr, and V' is mxr.

```
Matrix preferences = Ur.times(sqrtOfSingularMatrix(Sr));
Matrix features = sqrtOfSingularMatrix(Sr).times(Vr.transpose());
```

trainNext(test, preferences, features, minRank);

In the above, the preferences will be trained using the formulae.

$$P_{ik}*(t+1)=p_{tk}(t)+L*(R-R_a)_{ij}*f_{jk}(t)$$

$$F_{jk}(t+1)=f_{jk}(t)+L*(R-R_a)_{ij}*p_{ik}(t)$$

Matrix pridictedRatings2 = preferences.times(features);

ArrayList s6 = *getListFromMatrix*(pridictedRatings2);

double mae4 = we.getMAE1(test, original, s6);

The user ratings predictions are generated again to calculate MAE. MAE calculates the irrelevance between the recommendation value predicted by the system and the actual evaluation value rated by the user. The measurement method of evaluating the recommendation quality of recommendation system mainly includes statistical precision measurement method it includes to measure the recommendation quality. The generated prediction values are stored in an arraylist of Examples mentioned above and tabulated in the next sections. This arraylist Example is input for the class *populatePredictContentBoostedUJ* to generate the MAE values. The arraylist contains the values of the predicted user rating is geneated with the java class 'Examples' and actual user rating arraylist generated with the java class 'original'. These two arraylist are the inputs for *populatePredictContentBoostedUJ* java class to generate MAE values.

4.4 Model Experimentation

Dataset description: One of the largest datasets of explicit user preferences is MovieLens, a movie rating database collected over a period of 18 months by the Compaq Corporation. EachMovie contains the ratings of approximately 60,000 users for a set of 1,800 movies, 2.8 million ratings in all, or an average of 46 ratings per user. The rating scale ranges from 0 to 5. A subset of the ratings data from the MovieLens data set used for the purposes of comparison. 20% of the users were randomly selected to be the test users. In the dataset from grouplens website [114], it mentioned that the data sets u1.base and u1.test through u5.base and u5.test are 80%/20% splits of the u data into training and test data. Each of u1, u2, u3, u4, and u5 has

disjointed test sets for cross validation. These data sets can be generated from u.data by mku.sh. Source file u.data contained the u dataset by 943 users with 100000 ratings on 1682 items. Each user has rating at least 20 movies. This is a tab separated list of user id, item id, rating and timestamp.

4.5 Results and Discussions

The MAE values are computed using existing Singular value decomposition (SVD) algorithm and modified SVD for different test data sets u1.test, u2.test, u3.test, u4.test and u5.test and tabulated in table 1 to table 5.

The Comparative analysis of these computed values are presented.

a) MAE values for SVD on U1.test dataset

Neighbour Set Size	4	8	12	16	20	24	28
MAE for SVD existing	1.086	1.086	1.086	1.086	1.086	1.086	1.086
MAE for SVD modified	1.049	1.049	1.049	1.050	1.049	1.049	1.049

Table 4.5: MAE values for different neighbor sets for CF on u1.test

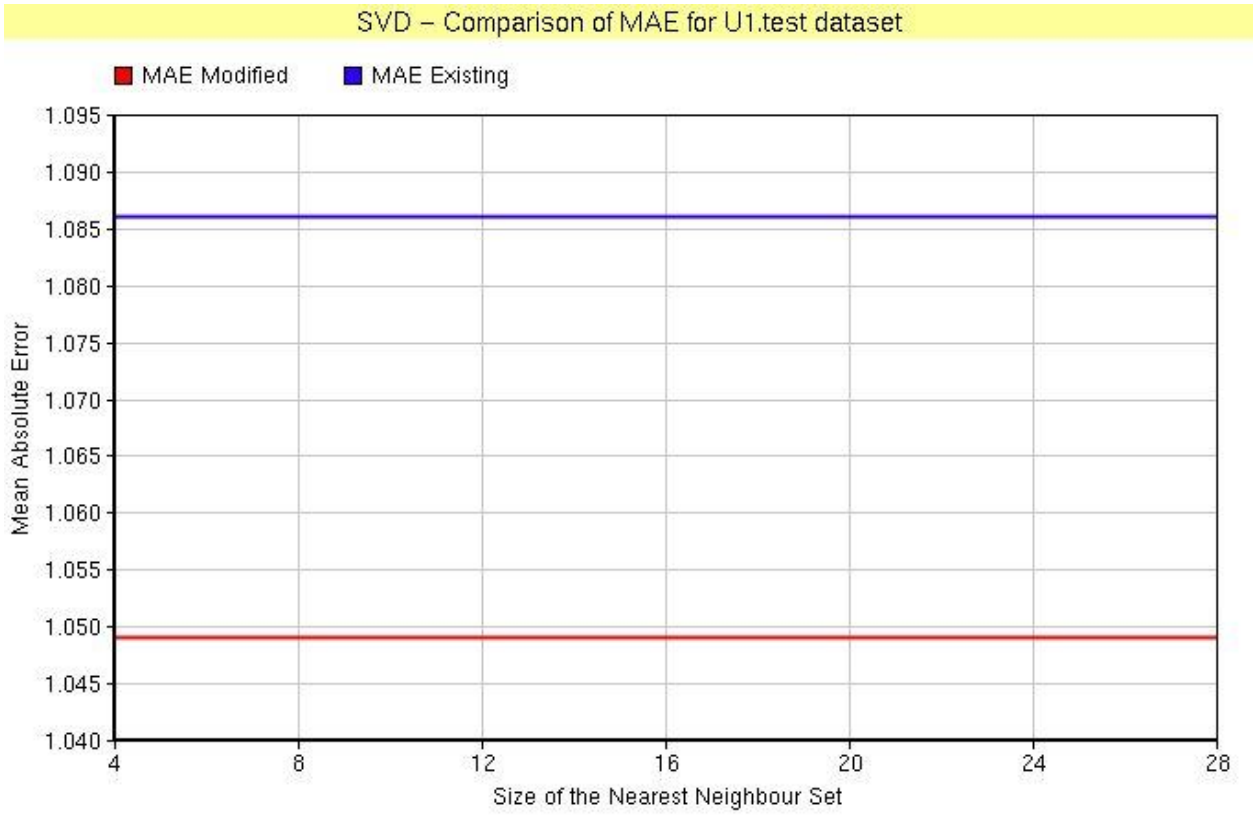

Fig.4.1. Comparison of MAE for Singular Value Decomposition (SVD) algorithm vs proposed algorithm on the U1.test dataset.

MAE values derived based on prediction quality recommendations is generated, lower values of MAE indicate better performance. MAE is shown in as two graphical representations, the blue line, represents an existing Singular Value Decomposition and the red line, and represents a modified algorithm, with lesser values than the existing.

b) MAE values for SVD on U2.test dataset

Neighbour Set Size	4	8	12	16	20	24	28
MAE for SVD existing	1.110	1.110	1.110	1.110	1.110	1.109	1.109
MAE for SVD modified	1.091	1.090	1.090	1.090	1.090	1.090	1.090

Table 4.6: MAE values for different neighbor sets for CF on u2.test

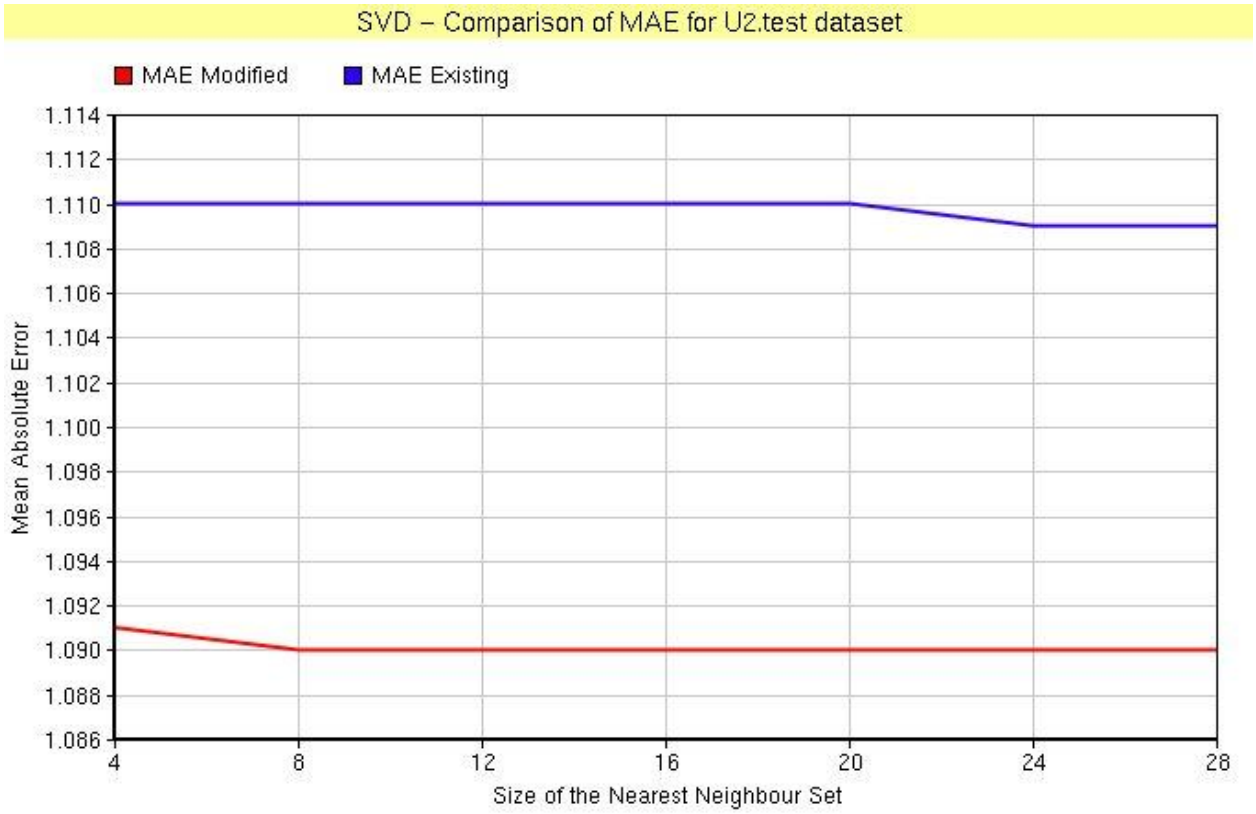

Fig. 4.2. Comparison of MAE for Singular Value Decomposition (SVD) algorithm vs proposed algorithm on the U2.test.

c) MAE values for SVD on U3.test dataset

Neighbour Set Size	4	8	12	16	20	24	28
MAE for SVD existing	1.110	1.110	1.110	1.110	1.110	1.110	1.110
MAE for SVD modified	1.091	1.091	1.091	1.091	1.091	1.091	1.091

Table 4.7: MAE values for different neighbor sets for CF on u3.test

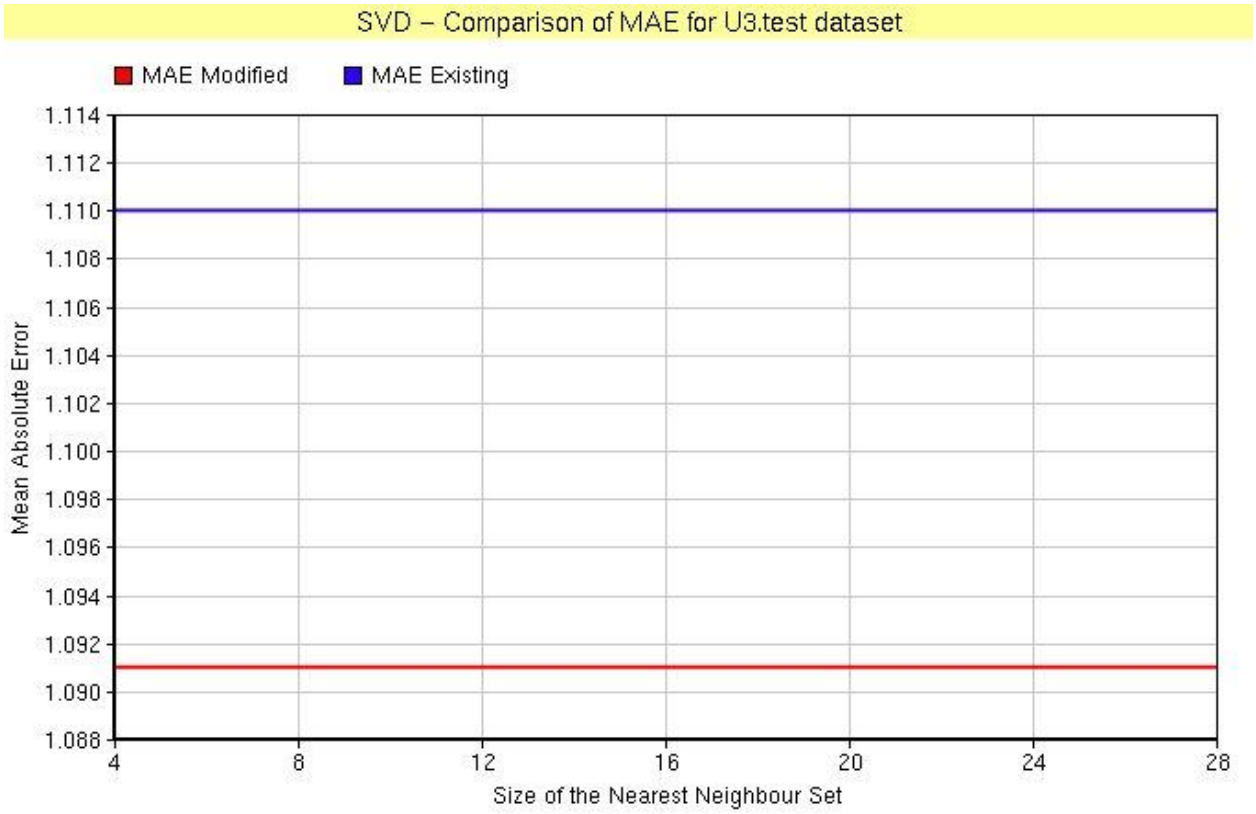

Fig.4.3. Comparison of MAE for Singular Value Decomposition (SVD) algorithm vs proposed algorithm on the U3.test.

d) MAE values for SVD on U4.test dataset

Neighbour Set Size	4	8	12	16	20	24	28
MAE for SVD existing	1.210	1.210	1.210	1.210	1.209	1.209	1.209
MAE for SVD modified	1.161	1.161	1.161	1.161	1.161	1.161	1.161

Table 4.8: MAE values for different neighbor sets for CF on u4.test

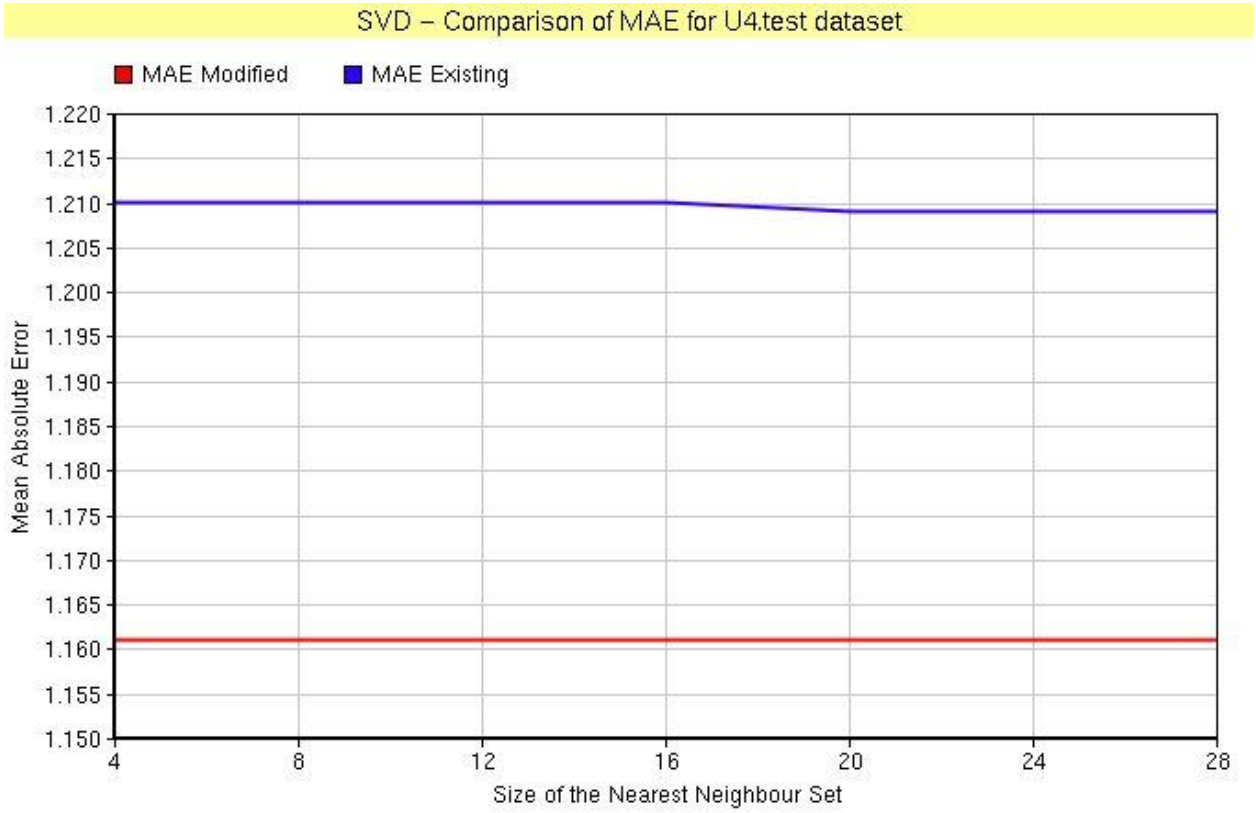

Fig. 4.4. Comparison of MAE for Singular Value Decomposition (SVD) algorithm vs proposed algorithm on the U4.test.

e) MAE values for SVD on U5.test dataset

Neighbour Set Size	4	8	12	16	20	24	28
MAE for SVD existing	1.187	1.187	1.187	1.187	1.187	1.188	1.188
MAE for SVD modified	1.168	1.168	1.167	1.167	1.167	1.167	1.167

Table 4.9: MAE values for different neighbor sets for CF on u5.test

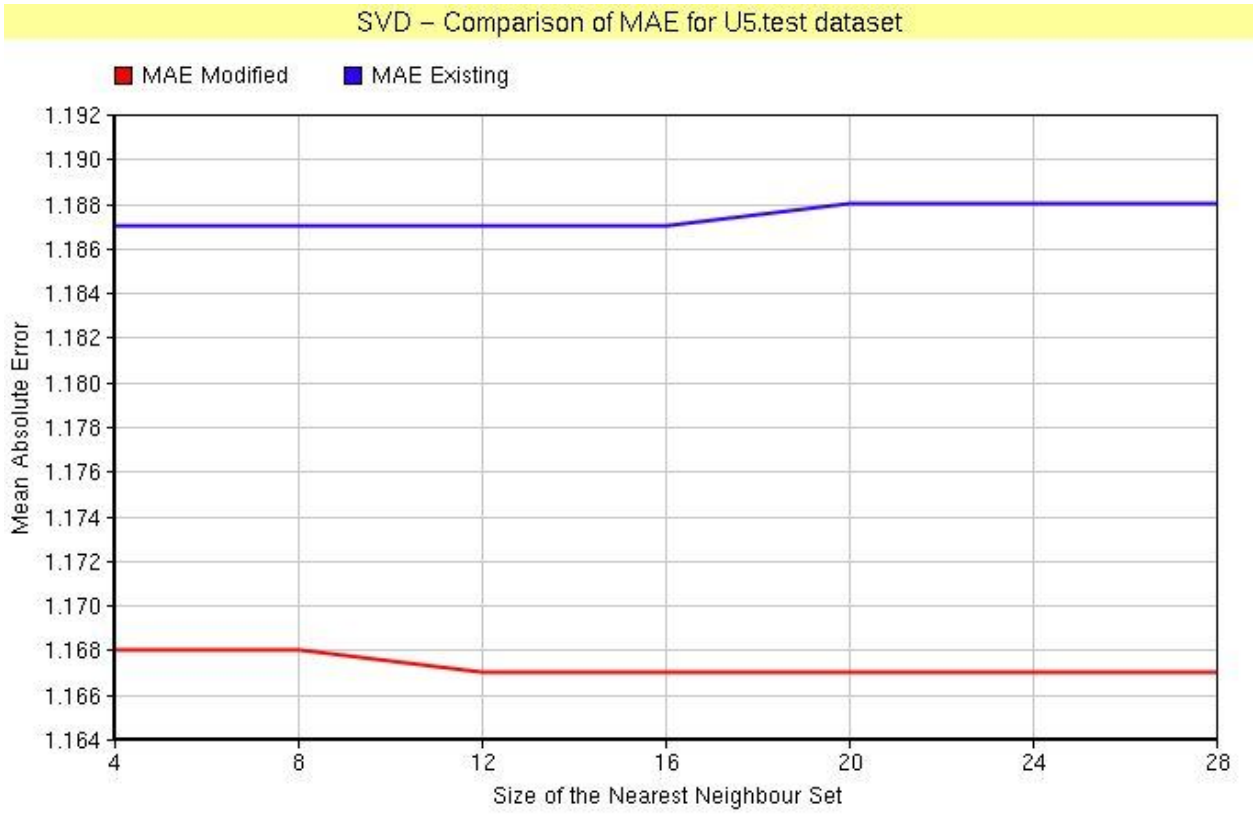

Fig. 4.5. Comparison of MAE for Singular Value Decomposition (SVD) algorithm vs modified algorithm on the U5.test.

Even though, there are an abundant number of free parameters, finding a good set of parameters which will work at optimum solution in the SVD algorithm is not possible. Although this is typical of machine learning problems, especially in this case, the experiments showed that even a single blind combination of results from very similar models gives performance almost close to the best model, which suggests that different models work especially good for different cases. Singular value decomposition is used in a wide variety of applications like latent semantic indexing (LSI), collaborative filtering and DNA expression analysis. The singular value decomposition problem evaluating for sparse matrices is adopted in this work. Based on this, the incremental SVD method proposed by Simon Funk [11] for predicting movie ratings based on previous user preferences using the dataset, MovieLens, is proposed for modification in this work.

The results presented in this chapter are given according to evaluation procedures with the experiments performed. The results for existing incremental SVD and the modified will be compared and presented. Derived MAE values for different test datasets from U1.test to U5.test is related with recommendation accuracy which is computed and compared for the existing and modified methods to see which one performs better. As mentioned earlier of this section, the dataset and the evaluation metric, mean absolute error (MAE) is evaluated for every fold in our 5-fold cross validation experiment. Finally the total MAE was computed from the whole set of users and folds in the experiments.

The MAEs for the different NNSs evaluation using U1.test dataset performs only 3.41% better improvement over existing SVD. Whereas with U2.test dataset and U3.test dataset it is only 1.71% increase is observed. 4.04% improvement is noticed in case of the results performed with U4.test dataset. 1.90% of improvement is noticed in U5.test dataset. It can be observe that both methods are the performed with unique performance in most of the cases the different in improvement is also very low. The prediction quality is decreased with increase of NNS in many cases. The overall performance of the modified SVD is slightly better than the existing. It performs much faster than existing methods when more singular values are required.

In this part the idea is to make intelligent predictions for the collaborative filtering data with the design of a model which would allow the system to learn to recognize complex patterns based on the training data. Singular value decomposition will be used to analyze the correlation of the data. The model should allow the user a simple and quick form of selecting the some of the most relevant collections and should be able to perform high level pattern comparisons. Modifications suggested into the algorithm computes singular values using the compact and then it compute the corresponding singular vectors by the twisted factorization with inverse iteration. The twisted factorization will selectively compute the required singular vectors because each singular vector is computed independent of other.

4.6 Conclusion

Simple Bayesian classifier, Apriori and Singular value decomposition algorithms are implemented in this chapter but only the singular value decomposition has been selected for modification as it is found to generate quick information needed apart from delivering high level pattern comparison. Overall the implemented algorithms i.e., Simple Bayesian CF Algorithm, Apriori algorithm and singular value decomposition (SVD), singular value decomposition algorithm performed well while deriving the prediction quality with Mean Absolute Error (MAE). The modified version results of incremental SVD are compared with the existing SVD algorithm. The modified incremental SVD method algorithm performs slightly well when compared to the standard algorithms when singular values are isolated in terms of prediction quality by performance evaluation of MAE. Experimenting with entirely different algorithms and combining results seems to be the best approach to improve particularly in model-based collaborative filtering.

CHAPTER V

HYBRID COLLABORATIVE FILTERING BASED ON PROBABILISTIC PROTOTYPE

Collaborative filtering recommender systems-recommend items by identifying other users with similar taste and use their opinions for recommendation. In this chapter the concepts of Content-Boosted Collaborative Filtering algorithm is discussed and proposed for modification in the existing approach using probabilistic measures to improve the performance. The methodology using Probabilistic Prototype used for predictions have been discussed. The Formulas that were used to implement these models including Bayes conditional probability, rating, Significance Weighting Factor, Harmonic Mean Weighting and Self-Weighting and Prediction. The experimentation is done with MovieLens dataset which is available for research purpose provided by the GroupLens Research Project agency at the University of Minnesota. The measured Mean Absolute Error (MAE) of the proposed model is compared with available models from literature and finally the performance analysis is done based on parameter MAE.

5.1 Hybrid Methods for Recommendation

In this section discussed about two hybrid algorithms. Collaborative filtering has been very successful in both research and practice, even though it has some disadvantages. For instance, it cannot recommend new items to the users and completely denies any information that could be extracted from contents of item. On the other hand, content-based methods fail in providing as good recommendations as collaborative filtering does. The reason for this is that it is hard to extract really high level meaningful features. Hybrid recommendations systems are developed in the recent years as an attempt of overcome the weakness of pure content-based collaborative methods. The main idea behind hybrid recommendation techniques is that a combination of algorithms can provide more accurate recommendations than a single algorithm and disadvantages of one algorithm can be overcome by other algorithms. A collaborative filtering system allows increasing the quality of a recommendation system by incorporating the content. Besides, when data is too sparse additional content information is a need in order to fit global

probabilistic models. The work presented in explains that a method that integrates both ratings and content data enables more accurate recommendations with a richer variety than pure content-based filtering techniques. Hybrid collaborative filtering systems can exploit the content and the different similarities or dissimilarities among user preferences in explicit cases. This specific combination can be important factor for recommending truly relevant items to the user.

The idea here is to develop some hybrid recommendation systems in order to overcome the weakness of traditional collaborative systems. Such a system should increase the quality especially when data is too sparse and additional content information is a need in order to fit global probability. The key factor of such a proposed algorithm is to convert a sparse user-rating matrix into a full ratings matrix using content data. Although the content-based information in this case is extracted from metadata, it could be used for the specific purpose of recommendation. This user interest model directly influences the recommendation quality of the recommendation system. During the construction of such a user interest model agents are used, which include user tracking agent, feedback agent and recommendation filtering agent. Since the agent could learn the interest and hobby information of the user based on the behavior and feedback of the user, the user interest library gets updated immediately, so that the recommendation results would more suit the requirement of the user.

This proposed work presents some ideas on how to create a content-boosted collaborative filtering system for movies. The main idea of this work is to convert a sparse user-rating matrix into a full ratings matrix using content data. Although the content-based information in this case is extracted from metadata, the general idea still can be used for the specific purpose of recommendation.

a) Collaborative Filtering with Content-based for Recommendation

Recommender Systems apply machine learning and data mining techniques for filtering hidden information and can predict whether a user would like a given resource. Collaborative filtering recommender systems recommend items by identifying other users with similar taste and use

their opinions for recommendation; whereas content-based recommender systems recommend items based on the content information of the items. These systems suffer from scalability, data sparsity, over specialization, and cold-start problems resulting in poor quality recommendations and reduced coverage. It explains that a method that integrates both ratings and content data enables more accurate recommendations with a richer variety than pure content-based filtering techniques. Recommender System is to generate significant recommendations to a collection of users for items or products that might interest them. Content-based filtering and collaborative filtering (CF) are two technologies used in recommender systems. Content-based filtering systems analyze the contents of a set of items together with the ratings provided by individual users to infer which non-rated items might be of interest for a specific user. Collaborative filtering methods accumulate a database of item ratings cast by a large set of users and then use those ratings to predict user's preferences for items. Collaborative filtering does not depends on the content descriptions of items, but purely depends on preferences expressed by a set of users. One major difficulty in designing content-based filtering systems lies in the problem of formalizing human perception and preferences. Practically it is not possible to formalize one user likes or dislikes a Movie or prefers one item over another. Like-wise, it is difficult to derive features which represent the difference between two items of extreme ends. Collaborative filtering provides a powerful way to overcome these difficulties. The information on personal preferences, tastes, and quality are all carried in either explicit or implicit user ratings. In the specific case of hybrid systems can exploit the content and the different similarities or dissimilarities among user preferences. This specific combination can be important factor for recommending truly relevant items to the user. Several hybrid collaborative filtering combinations have been examined and tested to combine two different algorithms to improve the quality of prediction as it is mentioned in the primary concept of the hybrid collaborative filtering. A hybrid recommendation approach by combining a content-based collaborative filtering and user-based collaborative filtering as it is stands for the better performance.

b) Content-Boosted Collaborative Filtering For Recommendation

Content-boosted Collaborative Filtering algorithm is proposed to address the cold start problem, in which items are classified into groups and predictions are made for users considering proper distribution of user ratings and external content information can be used to produce predictions for new users or new items. Likewise, Clustering Collaborative Filtering algorithms and other approaches such as an incremental-singular value decomposition collaborative filtering algorithm is found promising in dealing with the scalability problem. Latent semantic indexing (LSI) is helpful to handle the synonymy problem.

Most recommender systems use Collaborative Filtering or Content-based methods to predict new items of interest for a user. While both methods have their own advantages, individually they fail to provide good recommendations in many situations. Incorporating components from both methods, a hybrid recommender system can overcome these shortcomings. An elegant and effective framework for combining content and collaboration is presented. A content-based predictor to enhance existing user data, and then provides personalized suggestions through collaborative filtering is used in this approach. The dataset, MovieLens, provides the user-ratings matrix, which is a matrix of users versus items; the user-ratings matrix is very sparse, since most items have not been rated by most users. The content-based predictor is trained on each user-ratings vector and a pseudo user-ratings vector is created. A pseudo user-ratings vector contains the user's actual ratings and content-based predictions for the unrated items. All pseudo user-ratings vectors put together form the pseudo ratings matrix, which is a full matrix. Now given an active user's ratings, predictions are made for a new item using CF on the full pseudo ratings matrix.

Content based Predictor Algorithm

The implementation process starts with a bag-of-words naive Bayesian text classifier to learn a user profile from a set of rated movies. Prediction task can be assumed as a text-categorization problem and Movie content information can be assumed as text documents, and user ratings (1-5) can be as one of six class labels. Multinomial text model is adopted, in which a document is

modeled as an ordered sequence of word events drawn from the same vocabulary, V. The naive Bayes assumption states that the probability of each word event is dependent on the document class. For each class C_i , and word, $w_k \in V$ V, the probabilities, $P(C_i)$ and $P(w_k \mid C_i)$ can be evaluated from the training data. The subsequent probability of each class given a document D is computed using Bayes rule.

$$P(C_i \mid D) = \frac{P(C_i)}{P(D)} \prod_{i=1}^{|D|} P(a_i \mid C_i)$$

where a_i is the ith word in the document, and | D |is the number of words in the document.

In the implementation process, movies are represented as a vector of documents", d_m , one for each slot, the probability of each word given the category and the slot, $P(w_k \mid C_{i,} S_m)$must be estimated. The subsequent category probabilities for a film, F, computed using:

$$P(C_i \mid F) = \frac{P(C_i)}{P(F)} \prod_{m=1}^{S} \prod_{i=1}^{|d_m||} P(a_{mi} \mid C_{i,} S_m)$$

where S is the number of slots, and a_{mi} is the i^{th} word in the m^{th} slot.

Content-Boosted collaborative filtering process:

In content-boosted collaborative filtering to generate a pseudo user-ratings vector for every user 'u' in the database. The pseudo user-ratings vector, v_u, consists of the item ratings provided by the user 'u' and those predicted by the content-based predictor otherwise.

$V_{u,i}$ = { $r_{u,i}$: if user u rated item i
$C_{u,i}$: otherwise }

In the above equation $r_{u,i}$ denotes the actual rating provided by user u for item i, while $C_{u,i}$ is the rating predicted by the pure content-based system. The pseudo user-ratings vectors of all users

put together give the dense pseudo ratings matrix V. Collaborative filtering using this dense matrix is executed at this stage. The similarity between the active user a and another user u is computed using the Pearson correlation coefficient. Significance weighting factor is used to devalue the correlations generated based on few co-rated items to prevent bad predictors. If the number of co-rated items (n) is less than 50 then SWF is the product of their correlations. When $n \geq 50$ then the factor $Sg_{a,u} = 1$.

5.2 Proposed Model Description

5.2.1 Methodology of Hybrid Collaborative Filtering Based On Probabilistic Prototype

As the hybrid collaborative filtering is a combination two different techniques either one of the collaborative filtering algorithm performances of the individual components would almost certainly improve the performance of the whole system. The improvement in performance of content- based predictor or the CF algorithm, obviously, it would be able to improve total hybrid system's predictions. A better content-based predictor would mean that the pseudo ratings matrix generated would more accurately approximate the actual full user- ratings matrix. The final predictions in our system are based on a collaborative filtering (CF) algorithm. Hence, user-based collaborative filtering algorithm is proposed as CF algorithm, which was already proved as the best algorithm in memory-based collaborative filtering. The performance evaluation of the modified user-based collaborative filtering is tabulated in the chapter 4. The content-based predictor, a naive Bayesian text-classifier is used to learn a six-way classification task. It is proposed to improve content-based predictions using this approach.

The results presented in this chapter are given according to our evaluation measures and the experiments performed. The constant E_{ns} (equilibrium neighbor set) is introduced, which is the size of neighbour sets used where MAE is stable when run the modified collaborative algorithms. The test data is splitted into two different data sets to measure the E_{ns}. Part one will be used as test data on which predictions will be measured. Part two will be used to generate the MAE with predictions made from part one.

5.2.2 Proposed Algorithm

Input Require: set of items and average ratings.

Step 1: A pseudo user-rating vector for all users in the database is created by Using Harmonic Mean Weighting Factor (HMW). The process is, an accuracy of a pseudo user-ratings vector is computed for a user depends on the number of movies rated. If the user rated many items, the content-based predictions are good and hence his pseudo user-ratings vector is fairly accurate. Otherwise, if the user rated only a few items, the pseudo user-ratings vector will not be as accurate. It is clear that inaccuracies in pseudo user-ratings vector often yielded misleadingly high correlations between the active user and other users. Harmonic Mean weighting factor (HMW) is used to incorporate these low user-rated correlations.

$$HMW= 2m_i m_j / m_i + m_j$$

In the above equation, n_i refers to the number of items that user i has rated. The harmonic mean tends to bias the weight towards the lower of the two values namely m_i and m_j .Thus correlations between pseudo user-ratings with at least 50 user-rated items each, will receive the highest weight, regardless of the actual number of movies each user rated. Otherwise, even if one of the pseudo user-rating vectors is based on less than 50 user-rated items, the correlation will be devalued appropriately. The threshold 50 is based on the learning curve of the content predictor. It can be noted that initially as the predictor is given more and more training examples the prediction performance improves, but at around 50 it begins to level off. Beyond this is the point of diminishing returns; as no matter how large the training set is, prediction accuracy improves only marginally. The HMW includes the significance weighting to obtain the hybrid correlation weight.

$$hw_{a,u} = hm_{a,u} + Sg_{a,u}$$

Step 2 : Compute pseudo rating matrix V by combine the pseudo user-ratings vectors of all users.

Step 3:. Compute the similarity between active user a and another user u using the Pearson Correlation coefficient.

Step 4 : Compute mean-centered ratings of the best-n neighbors of that user as weighted sum of the active user.

Step 5: Constant E_{ns} equilibrium neighbor set is included to calculate modified self weighting factor in the final predictions. The other neighbors are given more importance than pseudo active user. A Self Weighting(SW_a) factor has been incorporated in the final prediction,

$$SW_a = \begin{cases} (n_a/50) * \max : & if\ n_a < 50 \\ \max : & Otherwise \end{cases}$$

where n_a is the number of items rated by the active user.

Step 6: Combine the above two weighting schemes to evaluate the CBCF predictions. Combining the above two weighting schemes, the final CBCF prediction for the active user a and item i is produced as follows:

$$p_{a,i} = \bar{v}_a + \frac{sw_a (c_{a,i} - \bar{v}_a) + \sum_{\substack{u=1 \\ u \neq a}}^{n} hw_{a,u}\, p_{a,u} (v_{u,i} - \bar{v}_u}{sw_a + \sum_{\substack{u=1 \\ u \neq a}}^{n} hw_{a,u} P_{a,u}}$$

Where

$C_{a,i}$ Corresponds to the Content predictions for the active user and item i.

$v_{y,i}$ is the pseudo user-rating for a user u and item i

$\bar{v}_u$ is the mean over all items for that user.

Sw_a, $hw_{a,u}$, and $P_{a,u}$ are evaluated.

n is the size of neighborhood.

5.3 Implementation of Proposed Algorithm

The implementation of the proposed model is done using JAVA. The description of implementation process is as follows:

A prediction for the active user is computed as a weighted sum of the mean centered votes of the best-n neighbors of that user is self weighting. In our approach, the pseudo active user to the neighborhood is added to give more importance than the other neighbors and to increase the confidence in the pure-content predictions for the active user. This has been done by incorporating a Self Weighting factor.

The main java classes designed and developed to evaluate the predictions for the content-based algorithm and content-boosted algorithm are *CBA5.java, NBSSimblanceRow.java, Probability.java and XYSplineRendererDemoTest.java.* A segment of java code snippet and the structure of the java classes that implements the content-based collaborative filtering, collaborative filtering predictor and Content-Boosted Collaborative Filtering algorithms proposed in the system is as follows.

```
List original = new ArrayList ();
String fileName2 = "D:\\Excelwork\\ml-data_0\\u.data";
int usersSize = 100;
int itemsSize = 1000;
we.initialize(original, usersSize, itemsSize);
we.populateFileToList(original, fileName2,  usersSize, itemsSize);
```

Here the List '*original*' is the list which contains the original ratings of the users which will be compared with the predicted ratings. It is designed to populate the list with the ratings read from the u.data file with the mentioned Path in the code.

```
List  test =  new ArrayList();
fileName2 = "D:\\Excelwork\\ml-data_0\\u5.test";
we.initialize(test, usersSize, itemsSize);
we.populateFileToList(test, fileName2,  usersSize, itemsSize);
```

The List '*test*' is the list which contains the test ratings of the users. Test data is the subset of original data. Using test data, it is designed to produce the user rating predictions and populating the 'test' list with values read from u5.test. Content-based predictions are generated by treating the task as a text-categorization problem. The movie data which contains the content information is considered as text documents, and user ratings given as 1-5.

```
fileName2 = "D:\\Excelwork\\ml-data_0\\u.genre";
ArrayList  genre =  new ArrayList();
we.initializeGenre(genre, fileName2);
```

The List '*genre*' is the list of genre of the movies. Each movie genre is given a unique number which is used in item classification.

```
fileName2 = "D:\\Excelwork\\ml-data_0\\u.item";
List  items =  new ArrayList();
we.initializeItems(items, 1682, 30);
we.populateItemsToList(items, fileName2,  1682, 30, genre);
```

The List 'items' is the list of all the items that presented in u.item. 1682 is number of items given, and 30 is the number of properties mentioned in the u.item file. It is designed and developed to populate the test list with values read from u.item. All the properties are embedded in a child list and the child list is added to parent list.

```
List docsIJ = new ArrayList();
we.initialize(docsIJ, usersSize, 5);
we.populateNoOfRatingsVsClazz(docsIJ, test, usersSize);
```

The List docsIJ is the list that contains the data of users which rated for first grade (rating given as 1). Similarly second grade and so on ratings given by particular user. Hence it contains number particular ratings (1-5) which was graded by each and every user. The method populateNoOfRatingsVsClazz is designed to develop the list of the ratings of all users.

```
List Examples = new ArrayList();
we.populateExamples(Examples, docsIJ);
```

'Examples' is the list of number of total ratings given by every user. It is designed and developed to generate MAE values for content boosted collaborative filleting.

MAE calculates the irrelevance between the recommendation value predicted by the system and the actual evaluation value rated by the user. The measurement method of evaluating the recommendation quality of recommendation system mainly includes statistical precision measurement method it includes to measure the recommendation quality [6].

The generated prediction values are stored in an arraylist of Examples mentioned above and tabulated in the next sections. This arraylist Example is input for the the class *populatePredictContentBoostedUJ* to generate the MAE values. The arraylist contains the values of the predicted user rating is generated with the java class 'Examples' and actual user rating arraylist generated with the java class 'original'. These two arraylist are the inputs for *populatePredictContentBoostedUJ java class to generate MAE values*

```
sheetData1 = ((List) ((ArrayList) test).clone());
    s3 =new ArrayList();
   Hashtable table = new Hashtable();
   for(int i=4; i<30; i=i+4){
// s3 =  we.populatePredictUJ(sheetData1, listSimblances,  i);
        s3  =  we.populatePredictContentBoostedUJ(test1,  test2,  Examples,
listSimblances,  i);
   double mae = we.getMAE1(test, original, s3);
      BigDecimalz1=new
BigDecimal(mae).setScale(2,BigDecimal.ROUND_HALF_UP);
 mae = z1.doubleValue();
System.out.println(" For neighbourset size -- " + i +" MAE is " + mae);
          table.put(new Double(i), mae);
          }
```

When the user search information, the system use the search key words that the user fills out as the representation of user interest key words, store them in the user interest table, and then assign them weight value. When the user collect some information, it can conclude that the user

is interested in such information, some words are taken out as the user interest key words and store them in the user interest table, assign them weight value, use the number of times to represent the weight value of the key words. The key words in the user interest table reflect the interest and requirement of the user, weight value reflects the degree of the preference, if the weight value is very large, it shows that the user is more interested in this information; on the other hand, the preference degree is small. When the number of the key words in the user interest table reach a certain value, delete the key words which had low weight value, thus keep the capacity of the user key words at a fixed level, so the key words in the user interest table can approach the preference of the user more accurately.

5.4 Model Experimentation

Collaborative filtering techniques need data files and information about user preferences from different people. It is important to use a constant and easy to reproduce dataset during experiments. Throughout the present work three different datasets are chosen to provide recommendations. The first dataset is Jester [114], a web based online joke recommendation system, which has been developing at University of California, Berkeley. This data has 73,421 users collected with a rating from -10 to +10. The second dataset MovieLens[115] is a web-based research recommender system. Each week hundreds of users visit MovieLens to rate and receive recommendations for movies. The site now has over 43000 users who have expressed opinions on 3500+ different movies. Movie Lens was developed by the GroupLens project at the University of Minnesota.

The research and analysis for user-based collaborative filtering system is carried out using MovieLens dataset which is available for research purpose provided by the GroupLens Research Project agency at the University of Minnesota. As it is mentioned the chapter 3, the dataset consists of 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies. It provides demographic data such as age, gender, and the zip code supplied by each person. The content of the information of every movie is considered as a set of slots. Each slot is represented by number of words. Further, the data has been segregated and discarded for having

less than 20 ratings or in complete demographic information. A subset of the ratings data from the MovieLens data set used for the purposes of comparison. 20% of the users were randomly selected to be the test users. The data sets u1.base and u1.test through u5.base and u5.test are 80%/20% splits of the u data into training and test data. Each of u1, u2, u3, u4, and u5 has disjointed test sets for cross validation. These data sets can be generated from u.data by mku.sh.

5.5 Results and Discussions

The MAE values are computed using existing content-boosted collaborative filtering (CBCF) and modified CBCF for U1.test, U2.test, U3.test, U4.test and U5.test for test dataset and tabulated in table 1 to table 5.

The Comparative analysis of these computed values are presented.

a) MAE for CBCF on U1.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for CBCF existing	0.98	0.91	0.89	0.86	0.85	0.85	0.85
MAE for CB CF Proposed Model	0.82	0.81	0.81	0.81	0.81	0.81	0.81

Table 5.1: MAE for different neighbor sets for CF on **u1.test**

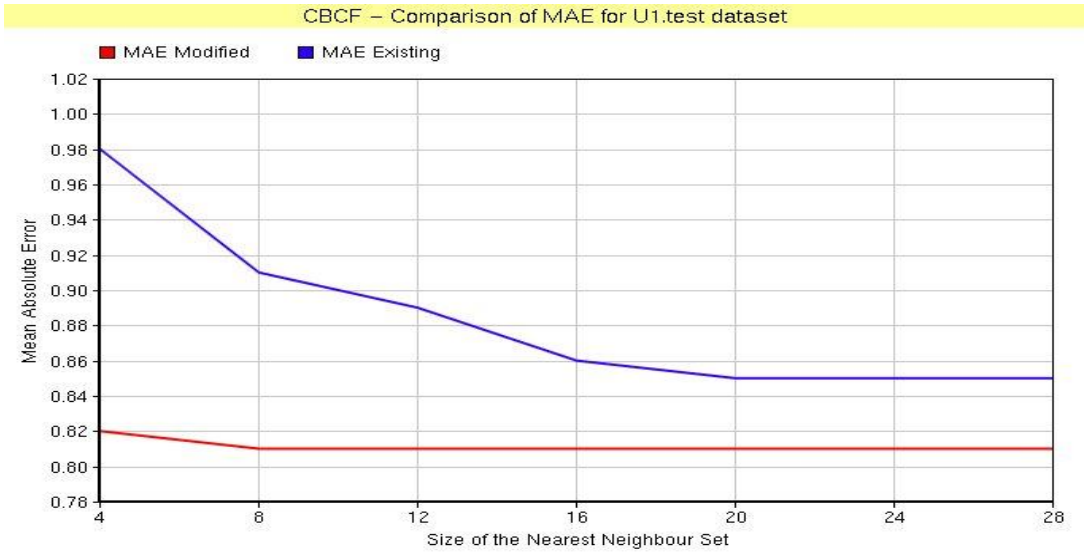

Fig. 5.1. Comparison of MAE for content-boosted collaborative filtering (CBCF) algorithm vs modified algorithm on the U1.test dataset.

MAE values derived based on prediction quality recommendations is generated, lower values of MAE indicate better performance. MAE is shown in as two graphical representations, the blue line, represents an existing content-boosted collaborative filtering and the red line, and represents a modified algorithm, with lesser values than the existing.

b) MAE values CBCF on U2.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for CBCF existing	1.07	0.98	0.94	0.92	0.91	0.90	0.90
MAE for CBCF Proposed Model	0.86	0.86	0.86	0.86	0.86	0.86	0.86

Table 5.2: MAE values for different neighbor sets on **u2.test**

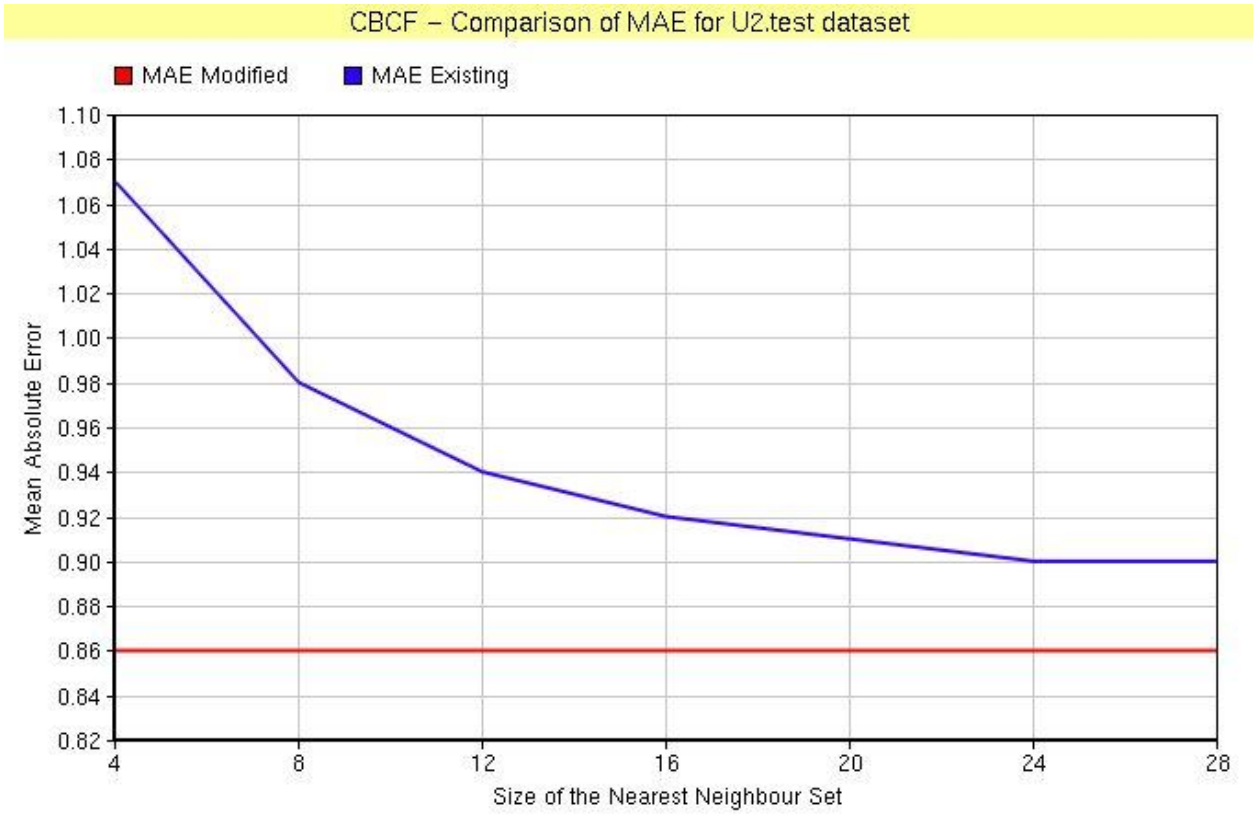

Fig. 5.2. Comparison of MAE for content-boosted collaborative filtering (CBCF) algorithm vs modified algorithm on the U2.test dataset.

c) MAE values for CBCF on U3.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for CBCF existing	1.07	0.98	0.95	0.94	0.93	0.92	0.91
MAE for CBCF Proposed Model	0.89	0.88	0.88	0.88	0.88	0.88	0.88

Table 5.3: MAE values for different neighbor sets on **u3.test**

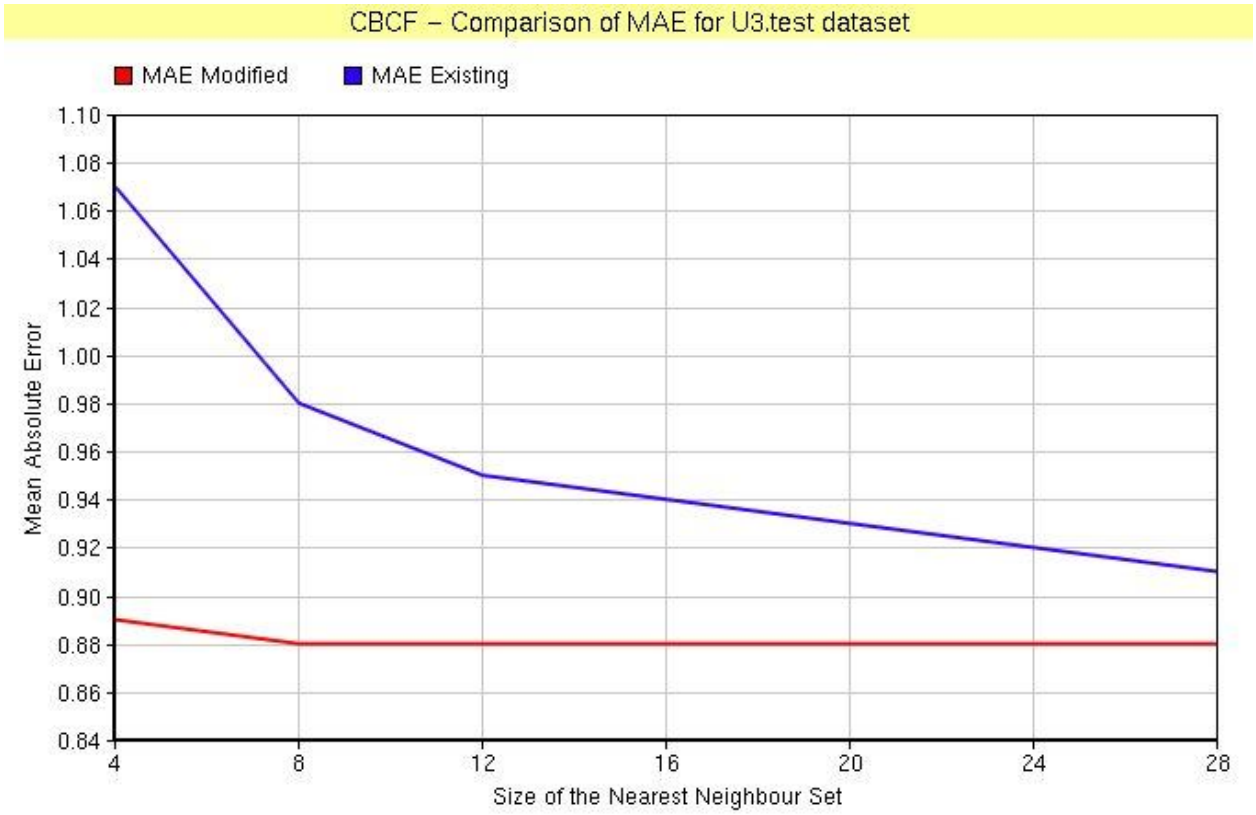

Fig. 5.3. Comparison of MAE for content-boosted collaborative filtering (CBCF) algorithm vs modified algorithm on the U3.test dataset.

d) MAE values for CBCF on U4.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for CBCF existing	1.11	1.05	1.01	0.99	0.99	0.98	0.97
MAE for CB CF Proposed Model	0.92	0.93	0.93	0.93	0.93	0.93	0.93

Table 5.4: MAE values for different neighbor sets on **u4.test**

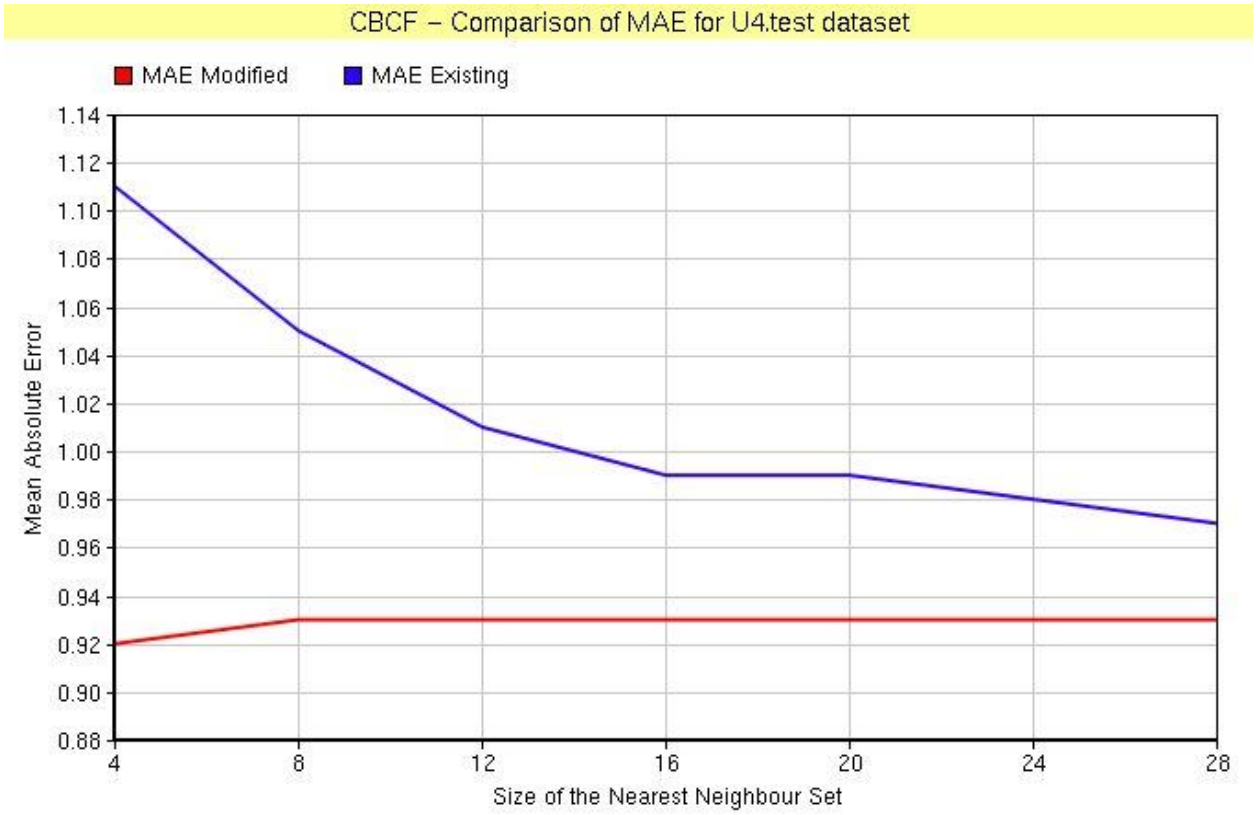

Fig. 5.4. Comparison of MAE for content-boosted collaborative filtering (CBCF) algorithm vs modified algorithm on the U4.test dataset.

e) MAE values for CBCF on U5.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for CBCF existing	1.13	1.04	1.01	1.00	0.99	0.98	0.97
MAE for CBCF Proposed Model	0.94	0.94	0.94	0.94	0.94	0.94	0.94

Table 5.5: MAE values for different neighbor sets on **u5.test**

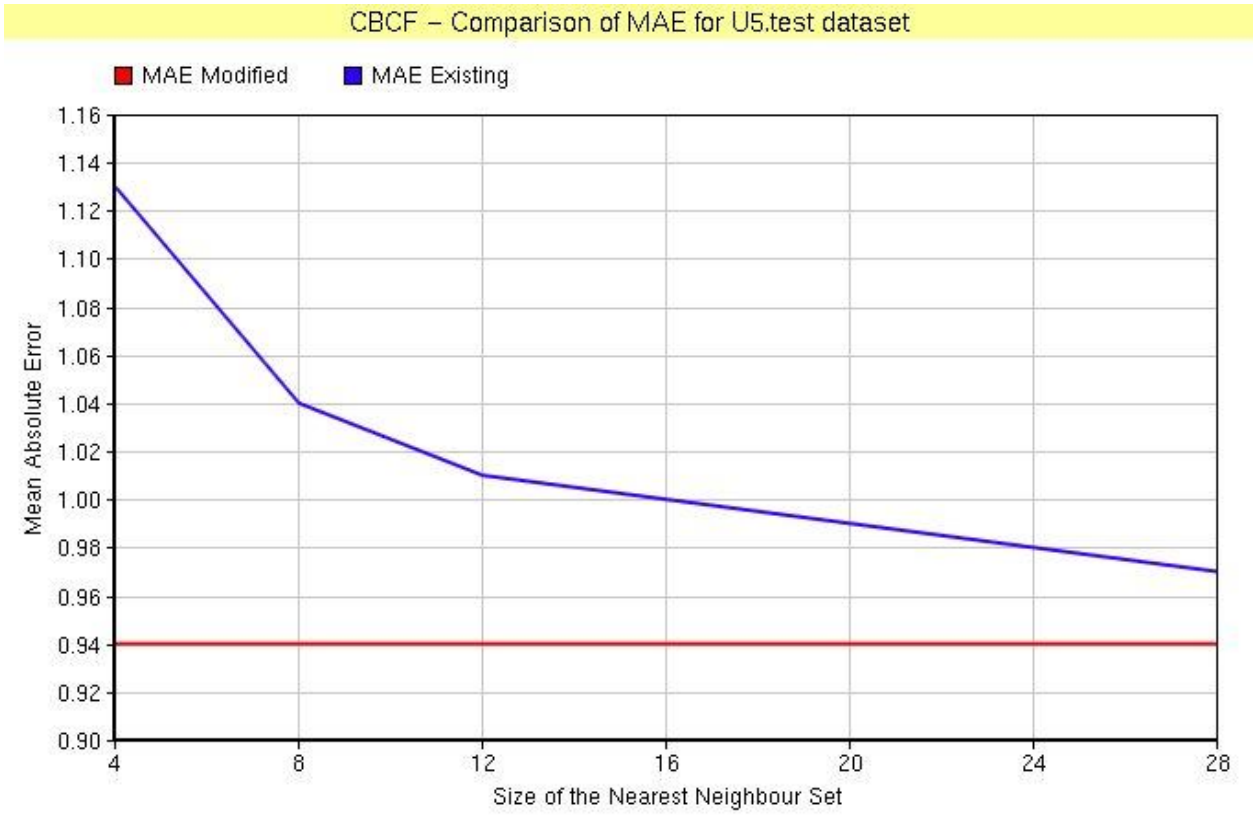

Fig. 5.5. Comparison of MAE for content-boosted collaborative filtering (CBCF) algorithm vs modified algorithm on the U5.test dataset.

The results presented in this chapter are given according to evaluation procedures with the experiments performed. Proposed user-based collaborative filtering algorithm and content-based predictor algorithm will be combined in content-boosted collaborative filtering (CBCF), the evaluated results are tabulated. Derived MEA values for different test datasets from U1.test to U5.test is related with recommendation accuracy which is computed and compared for both existing and modified methods to see which one performs better. MAE is obtained for every fold in our 5-fold cross validation experiment. Finally the total MAE was computed from the whole set of users and folds in the experiments.

The results presented in table 6.1 shows the MAEs for the different NNSs evaluation users using U1.test dataset performs 12.2% better improvement over existing CBCF. Whereas with U2.test dataset it is 19.6%, slightly increase is noticed. 17.1% improvement is noticed in case of the results performed with U4.test dataset. 16.8% improvement is noticed in U3.test dataset and U5.test dataset. It can be observe that modified CBCF is performed better than that of the existing traditional CBCF.

Most recommender systems use Collaborative Filtering methods to predict new items of interest for a user. While both methods have their own advantages, individually they fail to provide good recommendations in many situations. Incorporating components from both methods, a hybrid recommender system can overcome these shortcomings by Combine the information.

5.6 Conclusion

Hybrid recommendations systems are developed to overcome the weakness of traditional content-based collaborative systems and collaborative filtering algorithms. The design and development of models allowed the system to increasing the quality of a recommendation system when data is too sparse additional content information is a need in order to fit global probabilistic models. This chapter presents a modified content-boosted collaborative filtering system used the MovieLens dataset which contains user ratings on movies. One of the key factors of content-boosted collaborative filtering algorithm is to convert a sparse user-rating matrix into a full ratings matrix using content data. Although the content-based information in this case is extracted from metadata, the general idea still can be used for the specific purpose of recommendation. It is thus suggested to include the content of items into a collaborative-filtering system in order to improve the quality of its predictions and to solve the cold start problem.

CHAPTER VI

RESULTS AND DISCUSSIONS

In this chapter the performance evaluation and comparative analysis of Proposed Collaborative filtering algorithms, (i) Memory-Based Collaborative Filtering Algorithm Based on User Similarity Using Pearson Correlation, (ii) Model-Based Collaborative Filtering Algorithm Based on Composite Prototypes and (iii) Hybrid Collaborative Filtering Based on Probabilistic Prototype, are presented. The proposed algorithms and their efficiency have been already evaluated and presented in the respective previous chapters 4, 5 and 6. The results are utilized to evaluate the performance of the modified algorithms.

Collaborative filtering recommender systems recommend items by identifying other users with similar taste and use their opinions for recommendation; whereas content-based recommender systems recommend items based on the content information of the items. Content-based filtering and collaborative filtering (CF) are two technologies used in recommender systems. Content-based filtering systems analyze the contents of a set of items together with the ratings provided by individual users to infer which non-rated items might be of interest for a specific user. Collaborative filtering methods accumulate a database of item ratings cast by a large set of users and then use those ratings to predict user's preferences for items. Collaborative filtering does not depends on the content descriptions of items, but purely depends on preferences expressed by a set of users. These preferences can either be expressed explicitly by numeric ratings or can be indicated implicitly by user behaviors, such as clicking on a hyperlink, purchasing a book, or reading a particular news article. One major difficulty in designing content-based filtering systems lies in the problem of formalizing human perception and preferences. Practically it is not possible to formalize one user likes or dislikes a Movie or prefers one item over another. Like-wise, it is difficult to derive features which represent the difference between two items of extreme ends. Collaborative filtering provides a powerful way to overcome these difficulties. The information on personal preferences, tastes, and quality are all carried in either explicit or implicit user ratings. In the specific case of hybrid systems can

exploit the content and the different similarities or dissimilarities among user preferences. This specific combination can be important factor for recommending truly relevant items to the user.

Finally, after careful observations, it is proposed a hybrid recommendation approach by combining a content-based predictor and user-based collaborative filtering as it is stands for the better performance among all other filtering techniques. The proposed hybrid filtering transparently creates and maintains user preferences. It assists users by providing both collaborative filtering and content-based filtering includes two lists of recommendations based on two different filtering paradigms: collaborative filtering and content-based filtering. Content-based filtering is based on the correlation between the content of the pages and the user preferences. The collaborative filtering is based on a comparison between the user path of navigation and the access patterns of past users. Hybrid filtering may eliminate the shortcomings in each approach. Collaborative filtering can deal with any kind of content and explore new domains to find something interesting to the user. Content-based filtering can deal hidden with pages and the content by others.

6.1 Comparative Analysis between existing Collaborative Filtering Algorithms and Proposed Algorithms

Collaborative filtering is one of the most frequently used techniques in personalized recommendation systems. But currently used user-based collaborative filtering recommendation algorithm is based on item rating prediction has disadvantage in similarity computation method. In this work (i) user-based collaborative filtering, (ii) k-NN algorithm, (iii) item-based collaborative filtering (iv) Apriori algorithm, (v) singular value decomposition (SVD) (iv) Simple Bayesian CF Algorithm and (iiv) content-boosted algorithm are implemented and are further examined for several combinations of suitability for making an effective combination to form a hybrid collaborative filtering algorithm by reducing the quality of predictions. Based on the performance, the three algorithms (i) user-based collaborative filtering, (ii) singular value decomposition (SVD) and (iiv) content-boosted algorithm are modified for further improving the quality of prediction. Based on the performance of these proposed algorithms, the content-boosted collaborative filtering is performed well and hence it is proposed for modification.

6.2 Performance Evaluation of Proposed Collaborative Filtering Algorithms

The influence of various nearest neighbors set on predictive validity is tested by gradually increasing the number of neighbors. The dataset predicts item rating of the users are evaluated as per the opinions of the users chosen ratings. The results are shown in graphical representing MAE values vs. respective their neighbor set sizes.

MAE values are computed using content-based predictor, content-boosted CF algorithms and Singular value decomposition (SVD) for different test data sets u1.test, u2.test, u3.test, u4.test and u5.test and tabulated in table 1 to table 5. The Comparative analysis of these computed values are presented.

a) MAE values for CF, CBCF and SVD on U1.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for Proposed UBCF	1.370	1.370	1.370	1.370	1.370	1.370	1.370
MAE for Proposed SVD	1.049	1.049	1.049	1.050	1.049	1.049	1.049
MAE for Proposed CBCF	0.820	0.810	0.810	0.810	0.810	0.810	0.810

Table 6.1: MAE values for different neighbor sets for CF on **u1.test**

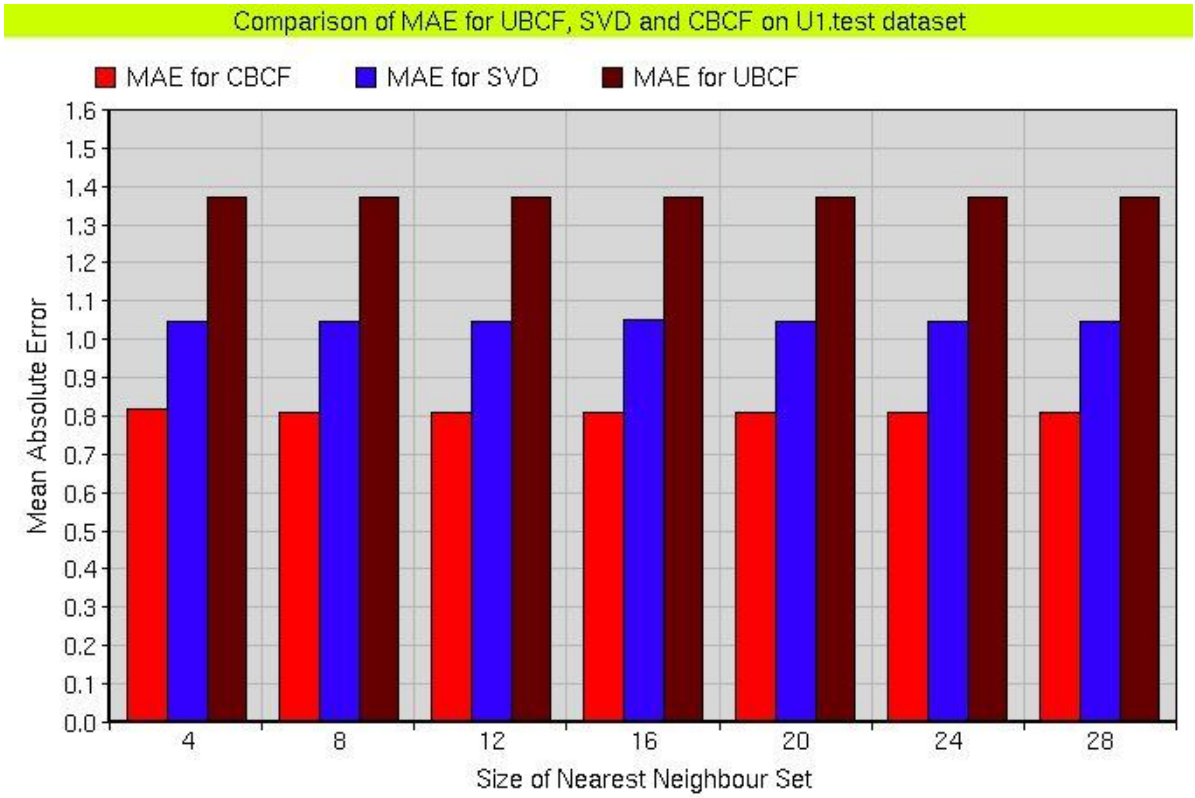

Fig.6.1. Comparing the performance of three modified collaborative filtering algorithms i.e., UBCF, SVD and CBCF recommendations on the U1.test dataset of MovieLens dataset.

MAE values derived based on prediction quality recommendations is generated, lower values of MAE indicate better performance. MAE is shown in as two graphical representations, the blue line, represents modified singular value decomposition (SVD) collaborative filtering , the maroon line, and represents a modified user-based collaborative filtering algorithm and the red line, represents a modified content-boosted collaborative filtering algorithm, with lesser values than the other modified algorithms.

b) MAE values for CF, CBCF and SVD on U2.test dataset:

Neighbor Set Size	4	8	12	16	20	24	28
MAE for Proposed UBCF	1.390	1.390	1.390	1.390	1.390	1.390	1.370
MAE for Proposed SVD	1.091	1.090	1.090	1.090	1.090	1.090	1.090
MAE for Proposed CBCF	0.860	0.860	0.860	0.860	0.860	0.860	0.860

Table 6.2: MAE values for different neighbor sets for CF on **u2.test**

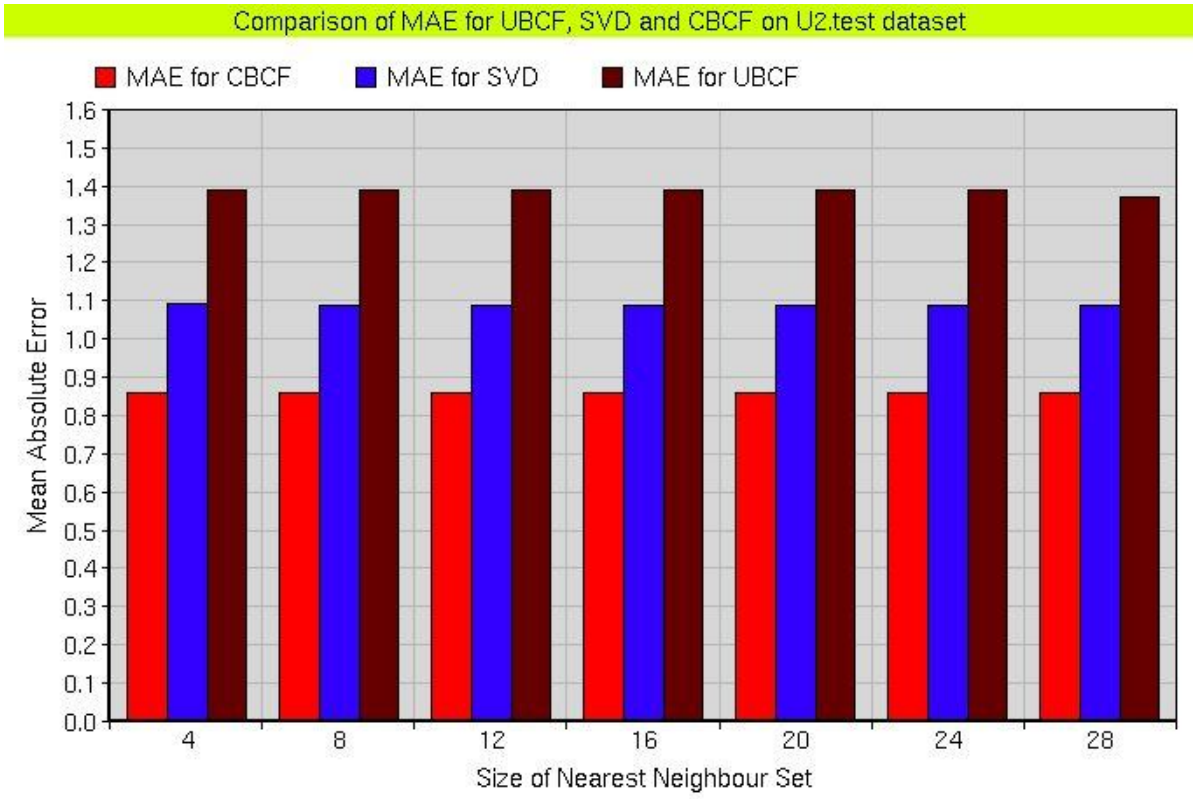

Fig.6.2. Comparing the performance of three modified collaborative filtering algorithms i.e., UBCF, SVD and CBCF recommendations on the U2.test dataset of MovieLens dataset.

c) MAE values for CF, CBCF and SVD on U3.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE Proposed UBCF	1.380	1.390	1.390	1.390	1.390	1.380	1.390
MAE for Proposed SVD	1.091	1.091	1.091	1.091	1.091	1.091	1.091
MAE for Proposed CBCF	0.890	0.880	0.880	0.880	0.880	0.880	0.880

Table 6.3: MAE values for different neighbor sets for CF on **u3.test**

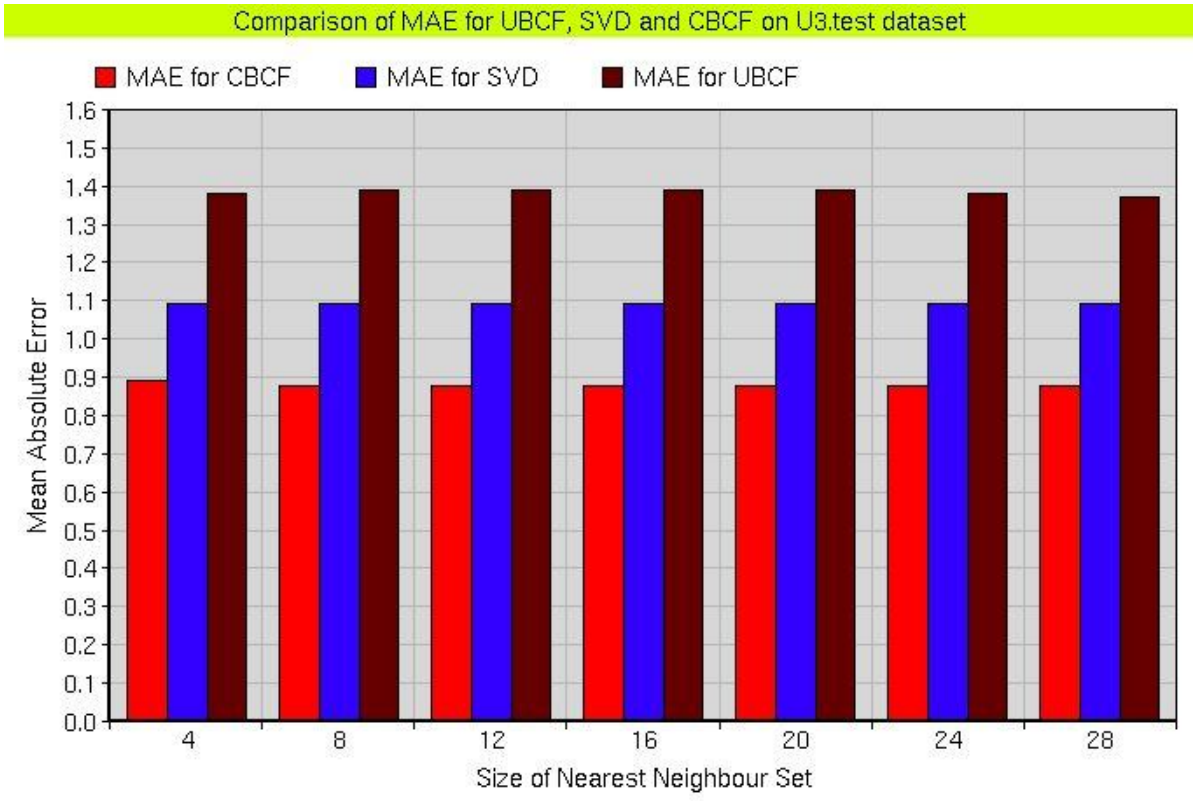

Fig.6.3. Comparing the performance of three modified collaborative filtering algorithms i.e., UBCF, SVD and CBCF recommendations on the U3.test dataset of MovieLens dataset.

d) MAE values for CF, CBCF and SVD on U4.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for Proposed UBCF	1.390	1.390	1.390	1.390	1.380	1.370	1.370
MAE for Proposed SVD	1.161	1.161	1.161	1.161	1.161	1.161	1.161
MAE for Proposed CBCF	0.920	0.930	0.930	0.930	0.930	0.930	0.930

.

Table 6.4: MAE values for different neighbor sets for CF on **u4.test**

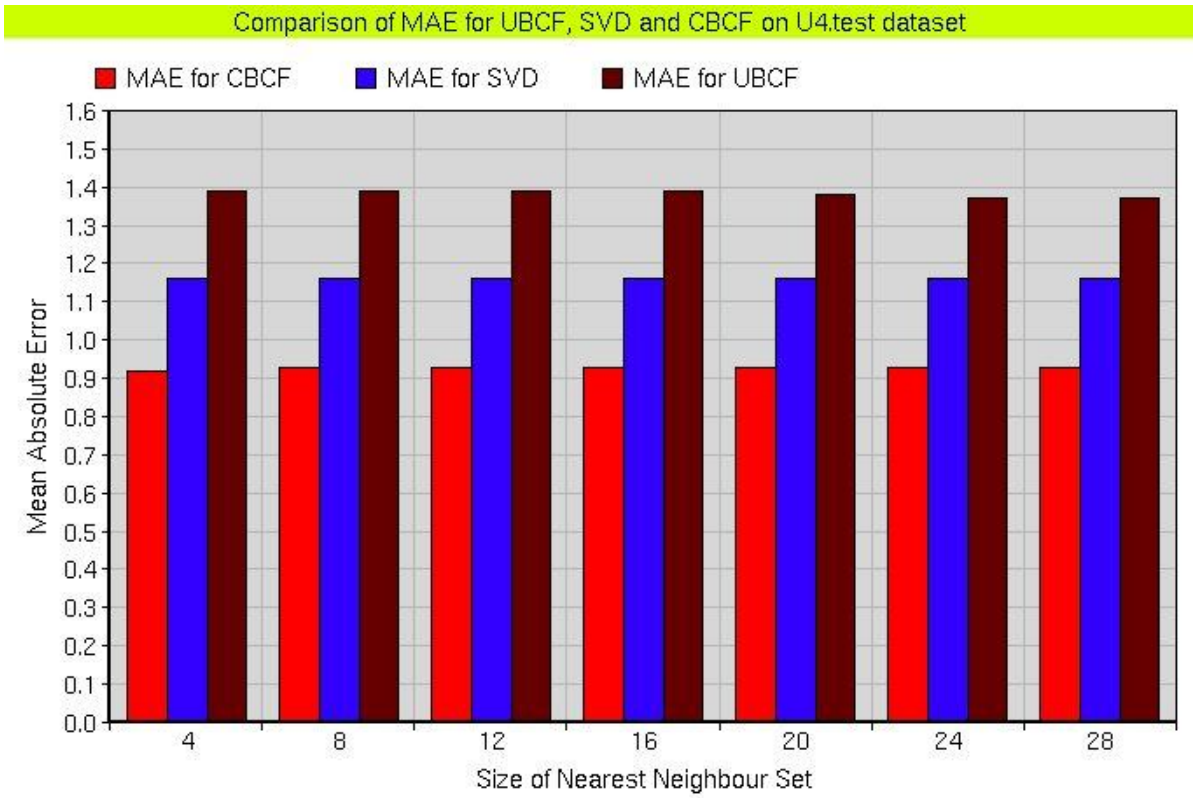

Fig.6.4. Comparing the performance of three modified collaborative filtering algorithms i.e., UBCF, SVD and CBCF recommendations on the U4.test dataset of MovieLens dataset.

e) MAE values for CF, CBCF and SVD on U5.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for Proposed UBCF	1.390	1.390	1.390	1.390	1.390	1.400	1.400
MAE for Proposed SVD	1.168	1.168	1.167	1.167	1.167	1.167	1.167
MAE for Proposed CBCF	0.940	0.940	0.940	0.940	0.940	0.940	0.940

Table 6.5: MAE values for different neighbor sets for CF on **u5.test**

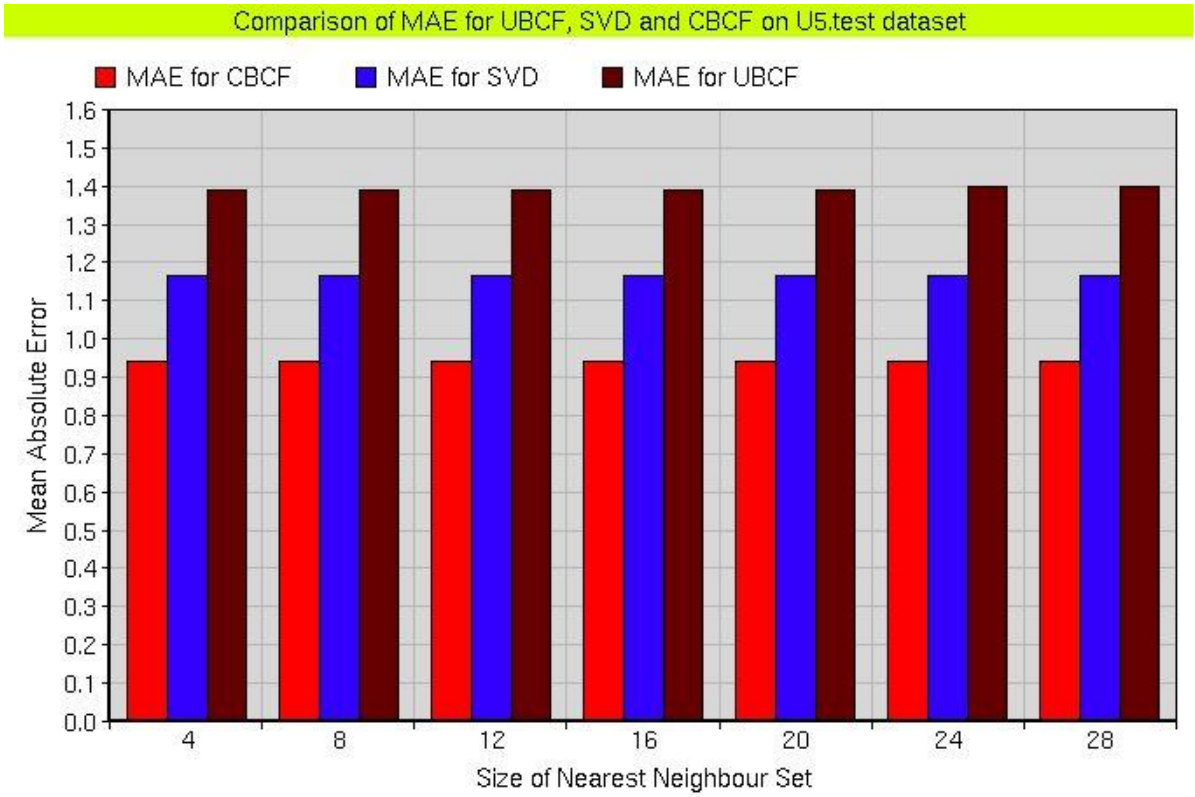

Fig.6.5. Comparing the performance of three modified collaborative filtering algorithms i.e., UBCF, SVD and CBCF recommendations on the U5.test dataset of MovieLens dataset.

6.3 Comparative Analysis

The proposed hybrid collaborative filtering is a combination of content-based filtering (content-based predictor) and a collaborative filtering (user-based collaborative filtering), the recommendation method using only the content-based filtering (Content), and a naive combined approach. The naive combined approach takes the average of the ratings generated by the collaborative filtering and the content-based filtering. The various methods were used to compare performance by changing the number of clustering users. Also, the proposed method was compared with the previously modified algorithms in chapter 3, 4 and 5 that use both collaborative filtering and content-based filtering method by changing the number of user evaluations on items. Since a pseudo rating matrix used, which is a full matrix, the root of the sparse rating problem and the first-rater problem is partially eliminated. Pseudo user-ratings vectors contain ratings for all items; and hence all users have been considered as potential neighbors. This increases the chances of finding similar users. The original user-ratings matrix may contain items that have not been rated by any user.

The MAE values tabulated in the above section are represented in bar diagrams for Comparative study. The different bar diagrams for different datasets U1.test to u5.test are presented in 6.1 to 6.5 tables respectively. The results presented in this chapter are given according to evaluation procedures with the experiments performed in the previous chapter 3, 4 and 5. The results for modified collaborative filtering algorithms i.e. user-based collaborative filtering, singular value decomposition and content-boosted collaborative filtering algorithm are compared along with their analysis. Derived MEA values for different test datasets from U1.test to U5.test is related with recommendation accuracy which is computed and compared for the existing and modified methods to see which one performs better. MAE is obtained for every fold in our 5-fold cross validation experiment. Finally the total MAE was computed from the whole set of users and folds in the experiments.

The results presented in table 6.1 shows the MAEs for the different NNSs evaluation users using U1.test dataset CBCF performs 40.7% and 21.9% better improvement over existing UBCF and SVD respectively. Whereas with U2.test dataset it is 38.1% and 21.1% of increase is noticed. 35.5% and 18.4% improvement is noticed in case of the results performed with U3.test dataset and with U4.test dataset it is 33.8% and 20.7%. In U5.test dataset it is 32.3% and 19.5% respectively. In most of the cases, it is observed that three modified algorithms have performed well and showed increase in the quality perdition performance. One thing that it is prominent is the differences between the overall performances of the modified UBCF, SVD and CBCF; CBCF performs better than the other compared algorithms.

6.4 Summary

The conclusions derived are based on the results produced by the experiments conducted with the different algorithms along with their performance evaluation done during the course of the present thesis. In this thesis, three classification methods namely; memory-based collaborative filtering, model-based collaborative filtering and hybrid collaborative filtering have been tested with incremental modification to better their performance. Among these modified techniques, the content-boosted collaborative filtering performed better in terms of results. The measured MAE values of modified collaborative filtering methods are tabulated to demonstrate the comparative analysis. The performance of content-boosted collaborative filtering algorithm is proven better during experimentation.

6.5 Scope for Future Research

Recommender systems are very useful and a powerful new technology for extracting additional value for a business from its user databases. The main aim of this work is to improve the quality recommendations and make it easier for the user to find relevant information from the internet. Recommender systems are becoming an essential tool in E-commerce being stressed by the huge volume of user data in existing databases which will be stressed even more by the increasing volume of user data available on the Web. These systems benefit users by enabling them to find items they like and helps to find items which they want to buy. In this thesis, modified algorithms have been proposed to improve the prediction quality for various

collaborative filtering recommender systems. The quality of the predictions evaluated is compared with similar traditional algorithms. Derived results show that the behavior of the modified algorithms with respect to the traditional algorithms better than that of the traditional algorithms.

The comparative analysis and performance evaluation based on modified collaborative filtering algorithms results are tabulated. Item-based collaborative filtering algorithm holds the assurance of allowing CF-based algorithms to scale to large data sets and at the same time produce high-quality recommendations. In the model-based approach, it is possible to retain only a small subset of items and produce reasonably good prediction quality. The item-item scheme is capable in addressing the two most important challenges of recommender systems are quality of prediction and high performance. Hybrid collaborative filtering, Content boosted collaborative filtering algorithm is designed by probabilistic prototype for better performance among the other comparative algorithms. Content-boosted collaborative filtering algorithm is ahead for better performance among the other modified comparative algorithms. Apriori algorithm is an efficient algorithm for finding all frequent item sets which implements level-wise search using frequent item property.

.

Furthermore, it is proved that content-based methods provide more accurate recommendations than other methods and they normally succeed on items that collaborative filtering cannot predict. Memory-based methods do not take into account other users opinions. In that sense, several relations between items, which normally are detected by collaborative filtering methods, are lost. Some final conclusions drawn for pure recommendation methods are related with the fact that most of the computations required to make predictions can be done offline. For instance, estimate similarities among items for content based methods and estimate correlations between users of items for collaborative filtering methods can be done while the system is being developed and when don't have to produce a fast answer immediately. Therefore, these methods can make predictions easily online while the user is waiting for a fast response, in web applications for instance. In the case of content based methods, since they don't need information about other users' preferences, the system might be used even in personal computers were information of a single user is stored. It is easier to improve the performance of collaborative

filtering prototypes since they only need more data about the users' preferences and strong correlations between items or users. It can conclude that with the correct amount of data collaborative filtering methods can outperform content based ones. However, this amount of data it is not easy to obtain and collaborative methods tend to fail a lot.

Although content-based and collaborative filtering methods are capable of producing good results, hybrid recommendation systems help to solve some of the disadvantages of both of them. This conclusion from the results obtained in our different metrics can be drawn by proved that if the data about users' preferences is not enough for the collaborative filtering methods to make a prediction then a hybrid system might use information about the content of the items to make a relevant prediction. Furthermore, when content based method fails on predicting quality hybrid methods produce results that are even better than the ones obtained with collaborative filtering. Another interesting point to discuss is that hybrid recommendation systems, especially if they depend on the predictions made by pure methods, can be used easily for making predictions online. As in the case of content-based collaborative filtering methods, hybrid, content-boosted collaborative filtering algorithm need some time to make some strong computations; once that this offline computing is done the time required for making predictions is minimal.

In this work seven Collaborative Filtering recommender algorithms are selected for implementation, in which one from different classification collaborative filtering is selected for modification and finally one of best of the three modified algorithms content-boosted collaborative filtering performs well. Firstly the user-based collaborative filtering with Pearson Correlation Coefficient as similarity measure, which is based on the relation between, pairs of users. Secondly the Item-based Collaborative Filtering algorithm that creates its predictions on the relation between pairs of items. And finally content-boosted collaborative filtering, which tries to be a fast but sufficiently accurate CF recommender algorithm, among all compared ones. These three algorithms using MAE as accuracy metric and recommendations per second as performance metric are compared.

The results showed that, although Pearson Correlation Coefficient was the most accurate, it was also several times slower than the content-boosted CF algorithm. Despite the fact that Item-based Collaborative Filtering was faster than Pearson Correlation Coefficient it was also the most inaccurate of the three compared algorithms. Moreover experiments with different neighbour sizes and an implementation of regression based predictions for Item-based Collaborative Filtering could be implemented. Furthermore the comparison could be extended by implementing default voting for the Pearson Correlation Coefficient.

Hybrid collaborative filtering and content-based filtering can significantly improve predictions of a recommender system. In this thesis, it is proved that how hybrid collaborative filtering significantly better than content-based filtering collaborative filtering by combined filtering approach. The proposed hybrid filtering exploits content-based filtering within a collaborative framework. It overcomes the disadvantages of both collaborative filtering and content-based filtering, by strengthening collaborative filtering with content-based collaborative filtering and vice versa. Further, due to the nature of the approach, any improvements in collaborative filtering or content-based filtering can be easily exploited to build a powerful improved recommender system i.e. content-boosted collaborative filtering (CBCF) which is a hybrid collaborative filtering approach.

6.6 Further Enhancements

Nowadays a lot of research is going on how to extract truly relevant items for users. One possible way to improve the results is to changing the modeling process. It is recommended that including features such as quality predictions in the model will increase the performance of the content-based recommendations. As explained in chapter 3, in this thesis a simply combination method of memory-based collaborative filtering algorithm are used to implement the hybrid prototype. Although this method has proven to outperform each traditional collaborative filtering method in different aspects, more complex way of mixing the traditional collaborative filtering methods will produce even better results is expected. Dataset collection, which is mainly based on rating of users on movies and produce bigger collections with more genres in it, might produce different results. Further study may adopt this, and perform experiments on bigger and

more varied collection by selecting MovieLens dataset. Besides, it is important to see whether efficiency and scalability are not really a big issue in recommendation systems. New ways of evaluating the system is another scope of extension. Hence, future work can be concentrate on research into new methods for mixing traditional collaborative filtering methods.

REFERENCES

[1]. Markov Balabanovic and Yoav S (1997) "Fab: Content Based, Collaborative recommendation" Vol. 40, No. 3, Communications of the ACM, pp.66-72, March 1997.

[2]. John S.B, David H and Carl K (1998) "Empirical Analysis of Predictive Algorithms for Collaborative Filtering", Proceedings of Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, July, 1998.

[3]. David M. P, Eric H and S L and C. Lee G (2000) "Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach", In Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-2000), pp.473-480, Morgan Kaufmann, San Francisco, 2000.

[4]. Thomas T and Robin C (2000) "Hybrid Recommender Systems for Electronic Commerce", AAAI Technical Report WS-00-04. AAAI (www.aaai.org), pp.78-84, 2000.

[5]. Badrul S, George K, Joseph K, and John R (2001) "Item Based Collaborative Filtering Recommendation Algorithms", WWW10, May 15, 2001, Hong Kong. ACM 1581133480/01/0005, pp.285-295, 2001.

[6]. Jonathan L.H, Joseph A, Konstan, Loren G. T and John T.R (2001) "Evaluating Collaborative Filtering Recommender Systems", 2001 ACM 1073-0516/01/0300-0034, pp.5-53, 2001.

[7]. Ken G, Theresa R, Dhruv G and Chris P (2001) "Eigentaste: A Constant Time Collaborative Filtering Algorithm", Kluwer Academic Publishers, Information Retrieval, 4, 133–151, pp.133-151, 2001.

[8]. Prem M, Raymond J. M and Ramadass N (2001) "Content-boosted Collaborative Filtering", Appears in *Proceedings of the SIGIR-2001 Workshop on Recommender Systems,* New Orleans, LA, , pp.1-9, September, 2001.

[9]. Mimi M. R, Andrew W and Kimberly L (2001) "Show me the way: A recommender system for educational web resources", International Journal of Artificial Intelligence and Education, pp.1-26, May 2001.

[10]. Steve C and Uwe A (2002) "A Recommender System based on the Immune Network", Proceedings CEC 2002, , Honolulu, USA, pp.807-813, 2002.

[11]. Zan H, Wingyan C, Thian-Huat O, Hsinchun C (2002) "A Graph-based Recommender System for Digital Library", *JCDL'02,* Portland, Oregon, USA. ACM 1-58113-513-0/02/0007, pp.65-73, 2002.

[12]. Edward F. H (2003) "Online Ranking/Collaborative filtering using the Perception Algorithm", Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

[13]. Michael C, Daniel Z, Hsinchun C, Michael H and David H (2003) "Design and evaluation of a multi-agent collaborative Web mining system" Decision Support Systems 35 (2003) 167– 183, 0167-9236/02/ 2002 Elsevier Science PII: S0167-9236(02)00103-3, 2003.

[14]. Michelle A, Daniel L, Marcel B, Harold B, Stephen G, Nancy H and Sean M (2003) "RACOFI: A Rule-Applying Collaborative Filtering System", Proc. IEEE/WIC COLA'03, Halifax, Canada, October 20F03. NRC 46507, 2003.

[15]. Andrew W, Mimi M. R, Kimberly L and David W (2004) "Collaborative Information Filtering: a review and an educational application", International Journal of Artificial Intelligence in Education 14 (2004) 1-26, 1560-4292/03 IOS Press, pp.3-28, 2004.

[16]. Byeong M. K and Qing L (2004) ” Probabilistic Model Estimation for Collaborative Filtering Based on Items Attributes” Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI’04) 0-7695-2100-2/04 IEEE, 2004

[17]. Kyung-Y. J, Dong-H. P and Jung-H. L (2004) “Hybrid Collaborative Filtering and Content-Based Filtering for Improved Recommender System”, ICCS 2004, LNCS 3036, pp.295-302, 2004. Springer-Verlag Berlin Heidelberg 2004.

[18]. Ludovic D and Patrick G (2004) ” Bayesian Network Model for Semi-Structured Document Classification” *Article published in IP&M: Bayesian Network and Information Retrieval (2004),* pp.1-25, 2004.

[19]. Saverio P, Marcos A G,Alves Edward A. F (2004) “Recommender Systems Research: A Connection-Centric Survey”, Journal of Intelligent Information Systems, 23:2, pp.107-143, 2004.

[20]. Yu L, Liu L and Li X (2004) “A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce” Expert Systems with Applications 28 (2005) 67–77, 0957-4174/2004 published by Elsevier Ltd. doi: 10.1016/ j.eswa.2004.08.013, 2004.

[21]. CaiNicolas Z, Sean M. McNee, Joseph A. K and Georg L (2005) “Improving Recommendation Lists Through Topic Diversification”, World Wide Web Conference Committee (IW3C2). WWW 2005, May 1014, Chiba, Japan. ACM 1595930469/ 05/0005, pp.22-32, 2005.

[22]. Gediminas A and Alexander T (2005) “Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions”, IEEE Transactions on Knowledge and Data Engineering, Vol.17, No.6, pp.734-749, June 2005.

[23]. Gui-Rong X, Chenxi L, Qiang Y, WenSi X, Hua-Jun Z, Yong Y and Zheng C (2005) "Scalable Collaborative Filtering Using Cluster-based Smoothing", *SIGIR'05*, August 15–19, 2005, Salvador, ACM 1-59593-034-5/05/0008, pp.114-121, 2005.

[24]. Guy S, David H and Ronen I. B (2005) "An MDP-Based Recommender System", Journal of Machine Learning Research 6 (2005) 1265–1295 Submitted 10/03; Revised 4/04; Published 9/05, pp.1265-1295, 2005.

[25]. Manos P and Dimitris P (2005) "Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents", Engineering Applications of Artificial Intelligence 18 (2005) 0952-1976/ 2005 Elsevier Ltd., doi: 10.1016/ j.engappai.2005.06.010, pp.781-789, 2005.

[26]. Miha G, Blaz F and Dunja M (2005) "kNN Versus SVM in the Collaborative Filtering Framework" *WebKDD* '05, August 21, Chicago, Illinois, USA 2005 ACM 1-59593-214-3, pp.24-31, 2005.

[27]. Monica C and J.D. Tygar (2005) "Collaborative Filtering CAPTCHAs", H.S. Baird and D.P. Lopresti (Eds.): HIP 2005, LNCS 3517, 2005. Springer-Verlag Berlin Heidelberg, pp.66-81, 2005.

[28]. Yan Z W, Luc M, and Nicholas R. J (2005) "A Market-Based Approach to Recommender Systems", ACM Transactions on Information Systems, Vol. 23, No. 3, July 2005, pp.227-266, 2005.

[29]. Ya-Y. S and Duen-R L (2005) "Hybrid recommendation approaches: collaborative filtering via valuable content information", Proceedings of the 38th Hawaii International Conference on System Sciences – 2005, 0-7695-2268-8/2005, 2005.

[30]. Byron L.D.B, Francisco D.A.T.C and Valmir M.F (2006) "C^2:: A Collaborative Recommendation System Based on Modal Symbolic User Profile", Proceedings of the

2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)0-7695-2747-7/06, 2006.

[31]. Jun W, Arjen P. D. V and Marcel J.T. R (2006) "Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion", SIGIR'06, August 6–11, Seattle, Washington, USA. Copyright 2006 ACM 1595933697/ 06/0008, 2006.

[32]. Michal A, Michael E and Alfred B (2006) "K-SVD: An Algorithm for Designing Over complete Dictionaries for Sparse Representation", IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 54, NO. 11, NOVEMBER 2006. 1053-587X, pp.4311-4322, 2006.

[33]. Xiaoyuan S, Taghi M. K (2006) "Collaborative Filtering for Multi-class Data Using Belief Nets Algorithms" Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06) 0-7695-2728-0/2006, 0-7695-2728-0/06, IEEE, pp.497-504, 2006.

[34]. Alexandros N, Apostolos N. P, Yannis M and Tatjana W-D (2007) " Robust Classifications Based on Correlation Between Attributes" International Journal of Data Warehousing & Mining, vol.3, issue.3, pp.14-27, July-September-2007.

[35]. Arkadiusz P (2007) "Improving regularized singular value decomposition for collaborative filtering", KDDCup.07 August 12, 2007, San Jose, California, USA, 2007 ACM 978-1-59593-834-3/07/0008, pp.39-42, 2007.

[36]. Hyung J.A (2007) "A Hybrid Collaborative Filtering Recommender System Using a New Similarity Measure", Proceedings of the 6th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 15-17, pp.494-498, 2007.

[37]. J. Ben S, Dan F, Jon H and Shilad S (2007) "Collaborative Filtering Recommender Systems**",** The Adaptive Web, LNCS 4321, 2007.Springer-Verlag Berlin Heidelberg, pp. 291- 324, 2007.

[38]. Laurent C, Frank M, and Marc B (2007) "Comparing State-of-the-Art Collaborative Filtering Systems", MLDM 2007, LNAI 4571, 2007. Springer-Verlag Berlin Heidelberg, pp.548-562, 2007.

[39]. Leo I, Anna L.G, Pasquale L, Marco D.G and Giovanni S (2007) "A Hybrid Content-Collaborative Recommender System Integrated into an Electronic Performance Support System", Seventh International Conference on Hybrid Intelligent Systems, 0-7695-2946-1/2007 IEEE DOI 10.1109/HIS.2007.30,2007.

[40]. Lijuan Z ,Yaling W, Jiangang Q and Dan L (2007) "Research and Improvement of Personalized Recommendation Algorithm Based on Collaborative Filtering", IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.7, July 2007.

[41]. Mandic .D, Vayanos. P, Boukis C, Jelfs. B, Goh S.L, Gautama T and Rutkowski T (2007) "Collaborative Adaptive Learning Using Hybrid Filters" ICASSP 2007, 1-4244-0728-1/07/C2007 IEEE, pp.III-921- III-924, 2007.

[42]. Pingfeng L, Guihua N and Donglin C (2007) "Exploiting Semantic Descriptions of Products and User Profiles for Recommender Systems", Proceedings of the 2007 IEEE Computational Intelligence and Data Mining (CIDM 2007), 1-4244-0705-2/2007, pp.179-185, 2007.

[43]. Robert M. B and Yehuda K (2007) "Improved Neighborhood-based Collaborative Filtering", KDDCup'07*,* August 12, 2007, San Jose, California, USA, ACM 978-1-59593-834-3/07/0008, pp.7-14, 2007.

[44]. Xiaoyuan S, Russell G, Taghi M. K and Xingquan Z (2007) "Hybrid Collaborative Filtering Algorithms Using a Mixture of Experts", 2007 IEEE/WIC/ACM International Conference on Web Intelligence, IEEE 0-7695-3026-5/07 DOI 10.1109/WI.2007.10.

[45]. Akhmed U and Alexander T (2008) "Improving Collaborative Filtering Recommendations Using External Data", 2008 Eighth IEEE International Conference on Data Mining, 1550-4786/08 IEEE, DOI 10.1109/ICDM.2008.44, pp.618-627, 2008.

[46]. Minchul Jung, Jehwan O and Eunseok L (2008) "Genetic Recommend Generating Method with Real-time Fitness Function Adaption" International Journal of u- and e- Service, Science and Technology. R&D Program in Korea and a result of subproject UCN 08B3-B1-10M, ITRC IITA-2008-(C1090-080-0046), Grant No. R01-2006-000-10954-0, pp.9-16, 2008.

[47]. Saara H, Pauli M and Evimaria T (2008) " Interpretable Nonnegative Matrix Decompositions", KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA, ACM 978-1-60558-193-4/08/08,2008.

[48]. Shiqian M, Donald G and Lifeng C (2008) "Fixed point and Bregman iterative methods for matrix rank Minimization", NSF Grant DMS 06-06712, ONR Grants N00014-03-0514 and N00014-08-1-1118, and DOE Grants DE-FG01-92ER-25126 and DE-FG02-08ER-58562, pp.321-353, 2008.

[49]. Simon F, Yvonne H and Yang H (2008) "Using Genetic Algorithm for Hybrid Modes of Collaborative Filtering in Online Recommenders", Eighth International Conference on Hybrid Intelligent Systems, 978-0-7695-3326-1/2008 IEEE, DOI 10.1109/HIS.2008.59,2008.

[50]. Somnath B and Krishnan R (2008) "Collaborative Filtering on Skewed Datasets", WWW 2008, April 21–25, 2008, Beijing, China, ACM 978-1-60558-085-2/08/04, pp.1135-1136 2008.

[51]. Xiaoyuan S, Taghi M. K and Russell G (2008) "Imputation-Boosted Collaborative Filtering Using Machine Learning Classifiers", *SAC'08*, March 16-20, 2008, Fortaleza, Ceará, Brazil. 2008 ACM 978-1-59593-753-7/08/0003, 2008.

[52]. Yehuda K (2008) "Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model", *KDD'08,* August 24–27, 2008, Las Vegas, Nevada, USA. Copyright 2008 ACM 978-1-60558-193-4/08/08, pp.426-436, 2008.

[53]. Zhang L, Xiao B and Guo J (2008) "A Hybrid Approach to Collaborative Filtering For Overcoming Data Sparsity", pp.1595-1599, ICSP2008 Proceedings, 978-1-4244-2179-4/2008 IEEE, 2008.

[54]. Antonina D, Felice F and Carlo T (2009) "Neighbor Selection and Recommendations in Social Bookmarking Tools", 2009 Ninth International Conference on Intelligent Systems Design and Applications, 978-0-7695-3872-3/09 IEEE, DOI 10.1109/ISDA.2009.245, pp.267-272, 2009.

[55]. Arno B, Elth O and Maarten V.S (2009) "Collaborative Filtering Using Random Neighbours in Peer-to-Peer Networks", CNIKM'09, November 6, 2009, Hong Kong, China. Copyright 2009 ACM 978-1-60558-807-0/09/11, 2009.

[56]. DanEr C (2009) "The Collaborative Filtering Recommendation Algorithm Based on BP Neural Networks", 2009 International Symposium on Intelligent Ubiquitous Computing and Education, 978-0-7695-3619-4/2009 IEEE, DOI 10.1109/IUCE.2009.121, pp.234-236, 2009.

[57]. DeJia Z (2009) "An Item-based Collaborative Filtering Recommendation Algorithm Using Slope One Scheme Smoothing" 2009 Second International Symposium on Electronic Commerce and Security, 978-0-7695-3643-9/2009 IEEE, DOI 10.1109/ISECS.2009.173, pp.215-217, 2009.

[58]. Fuguo Z (2009) “Reverse Bandwagon Profile Inject Attack against Recommender Systems”, 2009, Second International Symposium on Computational Intelligence and Design, 978-0-7695-3865-5/09IEEE, DOI 10.1109/ISCID. pp.15-18, 2009.

[59]. HengSong T and HongWu Y (2009) “A Collaborative Filtering Recommendation Algorithm Based On Item Classification”, 2009 Pacific-Asia Conference on Circuits, Communications and System, 978-0-7695-3614-9/2009 IEEE DOI 10.1109/PACCS.2009.68, pp.694-697, 2009.

[60]. Hyeong-J K and Kwang-S H (2009) “Moment Similarity of Random Variables to Solve Cold-start Problems in Collaborative Filtering” 2009 Third International Symposium on Intelligent Information Technology Application, 978-0-7695-3859-4/2009 IEEE, DOI 10.1109/IITA.2009.452, pp.584-587, 2009.

[61]. Hyeong-J K, Tae-H L and Kwang-S H (2009) “Improved Memory-based Collaborative Filtering Using Entropy-based Similarity Measures”, Proceedings of the 2009 International Symposium on Web Information Systems and Applications (WISA’09), pp.29-34, 2009.

[62]. Hyoung D.K (2009) “Applying Consistency Based Trust Definition to Collaborative Filtering”, published in KSII transaction of internet and information systems vol.3, no.4, pp.366-375, August, 2009.

[63]. Jian-guo L and Bing-Hong W (2009) “Improved Collaborative filtering algorithm via information transformation”, Published in "International Journal of Modern Physics C 20(2): pp.285-293, 2009.

[64]. Jinbo Z, Zhiqing L, Bo X and Chuang Z (2009) “An Optimized Item-Based Collaborative Filtering Recommendation Algorithm” Proceedings of IC-NIDC 2009 /978-1-4244-4900-2/2009 IEEE, 2009.

[65]. Mohammed N, Jenu S and Geun-Sik J (2009) "Enhanced Content-based Filtering using Diverse Collaborative Prediction for Movie Recommendation", 2009 First Asian Conference on Intelligent Information and Database Systems, 978-0-7695-3580-7/2009 IEEE,DOI 10.1109/ACIIDS.2009.77, pp.132-137, 2009.

[66]. Paul T.B, Noraswaliza A and Yue X (2009) "Improving the performance of collaborative filtering recommender systems through user profile clustering", 2009 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology – Workshops, 978-0-7695-3801-3/09 IEEE, DOI 10.1109/WI-IAT.2009.422, pp.147-150, 2009.

[67]. Ping S and HongWu Y (2009) "An Item Based Collaborative Filtering Recommendation Algorithm Using Rough Set Prediction", 2009 International Joint Conference on Artificial Intelligence, 978-0-7695-3615-6/2009 IEEE, DOI 10.1109/JCAI.2009, pp.308-311, 2009.

[68]. Prakash R, Juan L and Kendall N (2010) "A Multiagent System using Associate Rule Mining (ARM), a collaborative filtering approach", 978-1-4244-6349-7/2010 IEEE, pp.V7-574-V7-578, 2010.

[69]. Pu W and HongWu Y (2009) "A Personalized Recommendation Algorithm Combining Slope One Scheme and User Based Collaborative Filtering", 2009 International Conference on Industrial and Information Systems, 978-0-7695-3618-7/2009 IEEE, DOI 10.1109/IIS.2009.71, pp.152-154, 2009.

[70]. Reza S, Pedram P, George T, and Jean-P. H(2009) "Preserving Privacy in Collaborative Filtering through Distributed Aggregation of Offline Profiles", RecSys'09, October 23–25, 2009, New York, USA. ACM 978-1-60558-435-5/09/10, pp.157-164, 2009.

[71]. RuLong Z and SongJie G (2009) "Analyzing of Collaborative Filtering Using Clustering Technology", 2009 ISECS International Colloquium on Computing, Communication, Control, and Management," 978-1-4244-4246-1/2009, pp.57-59, 2009.

[72]. Sanjog R and Ambuj M (2009) "Filler Item Strategies for Shilling Attacks against Recommender Systems", Proceedings of the 42nd Hawaii International Conference on System Sciences – 2009, 978-0-7695-3450-3/2009 IEEE, pp.1530-1605, 2009.

[73]. Saptarshi G, Paul K, Ryan H and William S. C (2009) "Visualization Databases of the Analysis of Large Complex Datasets", Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5, pp.193-200, 2009.

[74]. SongJie G and HongWu Y (2009) "Joining User Clustering and Item Based Collaborative Filtering in Personalized Recommendation Services", 2009 International Conference on Industrial and Information Systems, 978-0-7695-3618-7/2009 IEEE, DOI 10.1109/IIS.2009.70, pp.149-151, 2009.

[75]. SongJie G and HongWu Y (2009) "An Item Based Collaborative Filtering Using BP Neural Networks Prediction", 2009 International Conference on Industrial and Information Systems, 978-0-7695-3618-7/2009 IEEE, DOI 10.1109/IIS.2009.69, pp.146-148, 2009.

[76]. SongJie Gong (2009)" Employing User Attribute and Item Attribute to Enhance the Collaborative Filtering Recommendation", Journal of Software, Vol.4, No.8, October 2009, doi:10.4304/jsw.4.8.883-890, pp.883-890, 2009.

[77]. SongJie G (2009) "Joining Case-based Reasoning and Item-based Collaborative Filtering in Recommender Systems" 2009 Second International Symposium on Electronic Commerce and Security, 978-0-7695-3643-9/2009 IEEE, DOI 10.1109/ISECS.2009.172, pp.40-42, 2009.

[78]. SongJie G, HongWu Y and HengSong T (2009) "Combining Memory-Based and Model-Based Collaborative Filtering in Recommender System", 2009 Pacific-Asia Conference

on Circuits, Communications and System, 978-0-7695-3614-9/09 IEEE, DOI 10.1109/PACCS.2009.66, pp.690-693, 2009.

[79]. Wolfgang W, Johannes H and Vivian P (2009) "Experiences from Implementing Collaborative Filtering in a Web 2.0 Application", Workshop on Adaptation and Personalization for Web 2.0, UMAP'09, June 22-26, pp.120-129, 2009.

[80]. X. Su and T. M. Khoshgoftaar (2009) "A Survey of Collaborative Filtering Techniques", Hindawi Publishing Corporation, Advances in Artificial Intelligence, Volume 2009, Article ID 421425, 19 pages, doi:10.1155/2009/421425, pp.1-20, 2009.

[81]. YiBo R and SongJie G (2009) "A Collaborative Filtering Recommendation Algorithm Based on SVD Smoothing", 2009 Third International Symposium on Intelligent Information Technology Application, 978-0-7695-3859-4/2009 IEEE, DOI 10.1109/IITA.2009.491, pp.530-532, 2009.

[82]. Yongjian F, Jianying M and Xiaofei R (2009) "A Rough Set-based Clustering Collaborative Filtering Algorithm in E-commerce Recommendation System" 2009 International Conference on Information Management, Innovation Management and Industrial Engineering, 978-0-7695-3876-1/09 IEEE, DOI 10.1109/ICIII.2009.556, pp.401-404 ,2009.

[83]. Zhang L, Xiao B and Guo J (2009) "An Approach of Finding Localized Preferences based-on Clustering for Collaborative Filtering", 2009 International Conference on Web Information Systems and Mining, 978-0-7695-3817-4/ IEEE,DOI 10.1109/WISM.2009.12, pp.19-22, 2009.

[84]. Zibin Z, Hao M, Michael R. L and Irwin K (2009) "WSRec: A Collaborative Filtering Based Web Service Recommender System", 2009 IEEE International Conference on Web Services, 978-0-7695-3709-2/09/ DOI 10.1109/ICWS.2009.30, pp.437-444, 2009.

[85]. A. Kumar, P. Thambidurai (2010) "Collaborative Web Recommendation Systems -A Survey Approach", Global Journal of Computer Science and Technology Vol. 9 Issue 5 (Ver 2.0), January 2010.

[86]. David S, Montserrat B, Aida V and Karina G (2010) "Ontology-driven web-based semantic similarity", J Intelligent Information Systems (2010) 35:383–413, DOI 10.1007/s10844-009-0103, 2010.

[87]. Fuguo Zhang (2010)"The Robustness of Trust-based Recommender Algorithm under Random Attack", 2010 2nd International Conference on Future Computer and Communication, 978-1-4244-5824-0/ 2010 IEEE, pp.V2-585-V2-588, 2010.

[88]. Fuzhi Z, Sushi F, Dongyan J and Qing T (2010) "DCFKQ : A DHT-based Distributed Collaborative Filtering Algorithm", Journal of Computational Information Systems 6:1(2010) 97-104 1553-9105/ 2010 Binary Information Press January, pp.97-104, 2010.

[89]. Hema B and Shikha M (2010) "Memetic Collaborative filtering based Recommender System", 2010 Second Vaagdevi International Conference on Information Technology for Real World Problems, 978-0-7695-4326-0/2010 IEEE,DOI 10.1109/VCON.2010.28, pp.102-107, 2010.

[90]. Jia R, Jin M, and Liu C (2010) "A New Clustering Method For Collaborative Filtering", 2010 International Conference on Networking and Information Technology, 978-1-4244-7578-0/2010/IEEE, pp.488-492, 2010.

[91]. Kimikazu K and Tikara H (2010)"Singular Value Decomposition for Collaborative Filtering on a GPU", WCCM/APCOM 2010, IOP Conf. Series: Materials Science and Engineering 10 (2010) 012017 doi:10.1088/1757-899X/10/1/012017, pp.1-5, 2010.

[92]. Martín L-N, Yolanda B-F, Jose J. P-A and Rebeca P. D-R (2010) "Property-Based Collaborative Filtering: A New Paradigm for Semantics-Based, Health-Aware

Recommender Systems" 2010 Fifth International Workshop on Semantics Media Adaptation and Personalization, 978-1-4244-8602-1/10/ 2010 IEEE, pp.98-103, 2010.

[93]. Mustansar A.G, and Adam P-B (2010) "An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering", Proceedings of the International Multi conference of Engineers and Computer scientists 2010 Vol I.IMECS 2010, March 17-19, 2010, Hong Kong, 2010.

[94]. Mustansar A.G and Adam P-B (2010) "A Scalable, Accurate Hybrid Recommender System", 2010 Third International Conference on Knowledge Discovery and Data Mining, 978-0-7695-3923-2/2010 IEEE DOI 10.1109/WKDD.2010.117, pp.94-98, 2010.

[95]. Prem Melville, vikas sindhwani (2010) "Recommender Systems" Encyclopedia of Machine Learning Chapter No: 00338 Page 1 22-4-2010. Encyclopedia of Machine Learning, DOI. Springer-Verlag Berlin Heidelberg, pp.1-9, 2010.

[96]. Prodan A (2010) "Implementation of a recommender system using collaborative filtering", studia univ. babes-bolyai, informatica, volume LV, number 4, pp.70-84, 2010.

[97]. Qian W, Xianhu Y and Min S (2010) "Collaborative Filtering Recommendation Algorithm based on Hybrid User Model", 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010). 978-1-4244-5934-6/10/2010 IEEE, pp.1985-1990, 2010.

[98]. Ryosuke F and Toshihiko W (2010) "Improvement of Collaborative Filtering Based on Fuzzy Reasoning Model", Biomedical Soft Computing and Human Sciences, Vol.16, No.2, IJBSCHS (2010-16-02-06), pp.49-57, 2010.

[99]. Sang H C, Young-S J, and Myong K. J (2010)" A Hybrid Recommendation Method with Reduced Data for Large-Scale Application", IEEE Transactions on Systems and

Cybernetics Part c: Applications and Reviews, Vol. 40, No.5, pp.557-566, September, 2010.

[100]. Teng-Kai F and Chia-Hui C (2010) "Learning to Predict Ad Clicks Based on Boosted Collaborative Filtering", IEEE International Conference on Social Computing / IEEE International Conference on Privacy, Security, Risk and Trust, 978-0-7695-4211-9/2010 IEEE,DOI 10.1109/SocialCom.2010.37, 2010.

[101]. WU Y and Tan X (2010) "A Real-time Recommender System Based on hybrid collaborative filtering", The 5th International Conference on Computer Science & Education Hefei, China. August 24–27, 2010. 978-1-4244-6005-2/10/2010 IEEE, pp.1909 -1912, 2010.

[102]. Xi C, Xudong L, Zicheng H, and Hailong S (2010) "RegionKNN: A Scalable Hybrid Collaborative Filtering Algorithm for Personalized Web Service Recommendation" 2010 IEEE International Conference on Web Services, 978-0-7695-4128-0/2010, DOI 10.1109/ICWS.2010.27, IEEE, pp.9-16, 2010.

[103]. Xiao C.C, Run J.L and Hui Y C (2010) "Research of Collaborative Filtering Recommendation Algorithm Based on Trust Propagation Model", 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), 978-1-4244-7237-611 0/ IEEE, pp.V4-177-V4-183, 2010.

[104]. Yanhong G, Xuefen C, Dahai D, Chunyu L and Rishuang W (2010) "An Improved Collaborative Filtering Algorithm Based on Trust in E-Commerce Recommendation Systems" 978-1-4244-5326-9/10/2010 IEEE,2010.

[105]. Yehuda K (2010) "Factor in the Neighbors: Scalable and Accurate Collaborative Filtering", ACM Transactions on Knowledge Discovery from Data, Vol. 4, No. 1, Article 1, pp.1-4, January 2010.

[106]. Zilei S and Nianlong L (2010) “A new user-based collaborative filtering algorithm combining data-distribution”, 2010 International Conference of Information Science and Management Engineering, 978-0-7695-4132-7/ 2010 IEEE, DOI 10.1109/ISME.2010.48, pp.19-23, 2010.

[107]. Zhaobin L, Wenyu Q, Haitao L and Changsheng X (2010) “A hybrid collaborative filtering recommendation mechanism for P2P networks”, Future Generation Computer Systems 26 (2010), 2010 Elsevier doi:10.1016/2010.04.002, pp.1409-1417, 2010.

[108]. Zhimin C, Yi J and Yao Z (2010) “A Collaborative Filtering Recommendation Algorithm Based on User Interest Change and Trust Evaluation”, International Journal of Digital Content Technology and its Applications Volume 4, Number 9, pp.107-113, December 2010.

[109]. Xavier A, Alejandro J, Nuria O, and Josep M. P (2011) “Data Mining Methods for Recommender Systems”, *Recommender Systems Handbook*, DOI 10.1007/978-0-387-85820-3_2, 2011. Springer Science + Business Media, LLC 2011, pp.39-71.

[110]. Reddit, An online site, http://www.reddit.com/

[111]. Amazon, an online portal, http://www.amazon.com/

[112]. eBay, an online portal, http://www.ebay.in/

[113]. Netflix, an online video portal, https://www.netflix.com/

[114]. Jester data, http://www.grouplens.org/.

[115]. MovieLens data, http://movielens.umn.edu

PUBLICATIONS FROM THE THESIS

[**1**]. Thomurthy Murali Mohan, Koichi Harada, Balakrishna.Annepu "**Recommended System for Neighborhood-Based Collaborative Filtering Algorithm using Pearson Correlation"** in **IJECCE** (International Journal of Electronics Communication and Computer Engineering) Volume 4, Issue 6, ISSN (Online): 2249-071X, ISSN (Print): 2278-4209, pp.1627-1632, November - 2013.

[**2**]. Thomurthy Murali Mohan, Koichi Harada, Balakrishna.Annepu "**Random Data Perturbation Technique on Model Based Collaborative Technique Using Composite Prototype Method" in IJCRD** (International Journal of Combined Research & Development) Volume 2, Issue 2, eISSN(Online): 2321-225X, pISSN (Print): 2321-2241, pp:1-8, February-2014.

[**3**]. Thomurthy Murali Mohan, Koichi Harada, Balakrishna.Annepu "**Hyrbid Collaborative Filtering Based on Probabilistic Prototype"** in **IJETT** (International Journal of Engineering Trends and Technology) Volume 5, Number 5, eISSN (Online): 2231-5381, pISSN (Print):2349-0918, pp.272-277, November - 2013.

[**4**]. Thomurthy Murali Mohan, Koichi Harada, Balakrishna.Annepu "**A Combination of Efficient Algorithms in Collaborative Filtering Techniques Using Pseudo Matrix"** in **IJETI(**International Journal of Engineering & Technology Innovations) Volume1, Issue 2, ISSN (Online): 2348-0866, pp.6-11, May - 2014.

INTERNATIONAL CONFERENCES

[1]. Thomurthy Murali Mohan, Koichi Harada, Balakrishna.Annepu **"Classification and Feature Selection Techniques in Memory based Collaborative Filtering"** in 13th International Conference **IC-GMBTI** (International Conference on Emerging Trends, Challenges & Opportunities in Global Business, Management, Tourism & Information Technology) on **September 28th & 29th 2013 Goa, India** Conducted by RDA (Research Development Association) Jaipur, India. Selected as BEST Paper Award in this Conference. ISBN No. 978-81-920965-2-0

[2]. Thomurthy Murali Mohan, Koichi Harada, Sasipalli. VS Rao and Naga Srinivas Rao. Kandala **"IT Implementation for Poverty Eradication – e-Commerce Perspective"** in **ECIC – 2008,** International Conference on Electronic Commerce in the 21st Century, **2-4 June, 2008,** Organized by Department of Computer Science & Technology and Information Technology, Tribhuvan University, Khatmandu, Nepal.